

# **МИКРОСХЕМА ИНТЕГРАЛЬНАЯ К1892ВГ1Я**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**РАЯЖ.431289.001Д17**

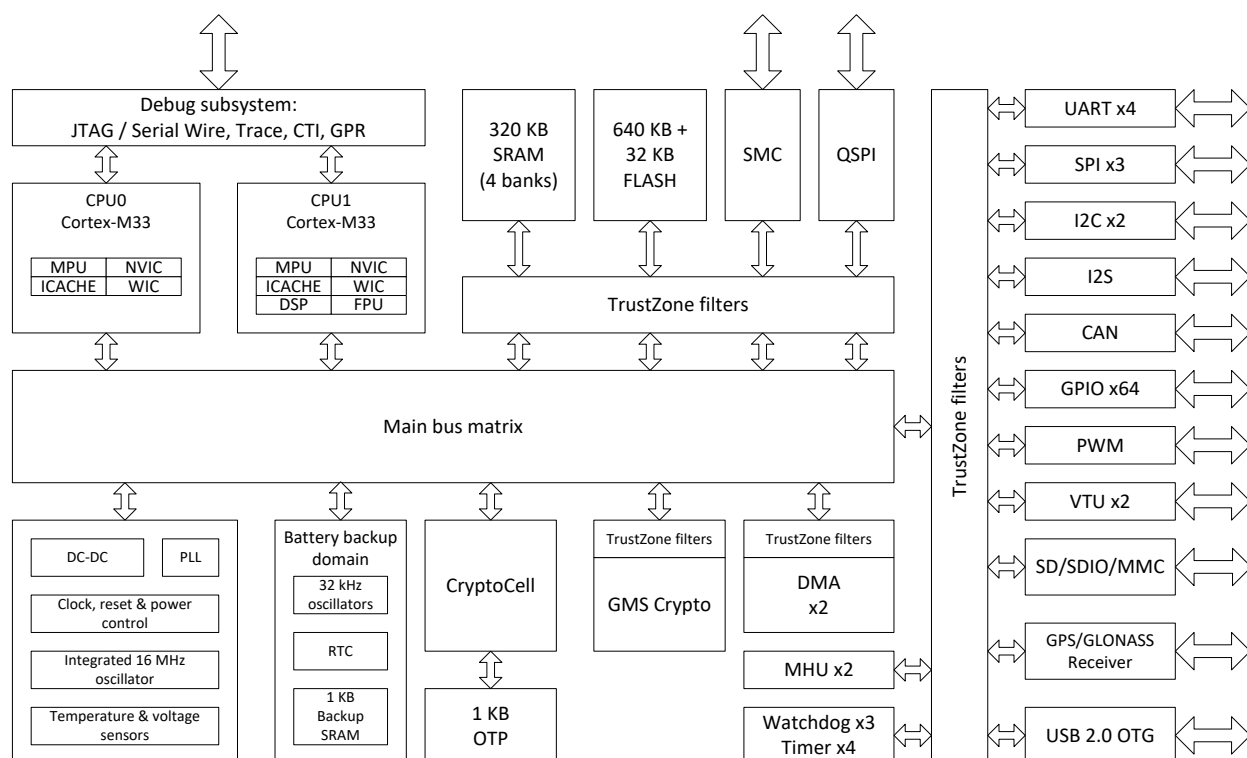
## ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ.....	4
2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ .....	8
3. SYSCTR: РЕГИСТРЫ УПРАВЛЕНИЯ СИСТЕМОЙ .....	74
4. SPCTR/NSPCTR: БЛОКИ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ .....	83
5. ICACHE: КЭШ ИНСТРУКЦИЙ .....	105
6. FCACHE: КЭШ FLASH-ПАМЯТИ.....	112
7. FLASH: ПОДСИСТЕМА FLASH ПАМЯТИ.....	119
8. JTM: СЕНСОРЫ ТЕМПЕРАТУРЫ И НАПРЯЖЕНИЯ.....	136
9. IOCTR: МНОГОФУНКЦИОНАЛЬНЫЕ ПОРТЫ ВВОДА/ВЫВОДА .....	141
10. GPIO: БЛОК ВВОДА/ВЫВОДА ОБЩЕГО НАЗНАЧЕНИЯ .....	148
11. UART: КОНТРОЛЛЕР UART .....	157
12. SPI: КОНТРОЛЛЕР SPI.....	208
13. I2C: КОНТРОЛЛЕР I2C .....	229
14. I2S: КОНТРОЛЛЕР ИНТЕРФЕЙСА I2S.....	256
15. CAN: КОНТРОЛЛЕРНАЯ СЕТЬ.....	272
16. VTU: МНОГОФУНКЦИОНАЛЬНЫЕ БЛОКИ ТАЙМЕРОВ .....	363
17. PWM: ШИРОТНО-ИМПУЛЬСНЫЙ МОДУЛЯТОР .....	364
18. QSPI: ИНТЕРФЕЙС ПАМЯТИ QUAD SPI.....	402
19. SMC: КОНТРОЛЛЕР ВНЕШНЕЙ СТАТИЧЕСКОЙ ПАМЯТИ .....	423
20. GNSS: КОНТРОЛЛЕР ЦИФРОВОЙ ЧАСТИ ПРИЕМНИКА НАВИГАЦИОННОГО СИГНАЛА .....	442
21. SDMMC: ИНТЕРФЕЙС КАРТ SECURE DIGITAL INPUT/OUTPUT И MULTI MEDIA .....	478
22. USB: UNIVERSAL SERIAL BUS 2.0 OTG.....	483
23. DMA: КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ.....	495
24. RWC: СЧЕТЧИК РЕАЛЬНОГО ВРЕМЕНИ, WAKE-КОНТРОЛЛЕР .....	547
25. MPC: КОНТРОЛЛЕР БЕЗОПАСНОСТИ ПАМЯТИ .....	556
26. OTP: ОДНОКРАТНО ПРОГРАММИРУЕМАЯ ПАМЯТЬ .....	563
27. CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCCELL.....	580
28. GMS: GMS CRYPTO CORE.....	582
29. MHU: БЛОК ОБМЕНА СООБЩЕНИЯМИ .....	669
30. PPU: КОНТРОЛЛЕР УПРАВЛЕНИЯ ПИТАНИЕМ .....	672
31. SWDT, NSWDT, LPWDT: СТОРОЖЕВЫЕ ТАЙМЕРЫ.....	684

<b>32. TIM0/TIM1/LPTIM: ТАЙМЕРЫ ОБЩЕГО НАЗНАЧЕНИЯ .....</b>	<b>685</b>
<b>33. DTIM: СДВОЕННЫЙ ТАЙМЕР ОБЩЕГО НАЗНАЧЕНИЯ .....</b>	<b>689</b>
<b>34. СИСТЕМА ОТЛАДКИ МИКРОСХЕМЫ .....</b>	<b>690</b>
<b>35. ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ .....</b>	<b>696</b>
<b>36. СПРАВОЧНАЯ ИНФОРМАЦИЯ .....</b>	<b>704</b>
<b>37. ИСТОРИЯ ИЗМЕНЕНИЙ.....</b>	<b>712</b>

# 1. ВВЕДЕНИЕ

Общая схема микросхемы K1892ВГ1Я представлена на Рисунок 1.1.



**Рисунок 1.1. Блок-схема K1892ВГ1Я**

Состав микросхемы и основные особенности:

- Два ядра ARM Cortex-M33
  - ядро CPU0 (энергоэффективное ядро);
  - ядро CPU1 с расширениями DSP, FPU (быстродействующее ядро).
- Кэш инструкций размером 16 Кбайт для каждого ядра CPU.
- Встроенная FLASH память
  - размер: 640 Кбайт основной раздел, 32 Кбайт системный раздел;
  - размер страницы: 8 Кбайт;
  - дополнительный Flash кэш размером 8 Кбайт.
- Встроенная SRAM память
  - четыре независимых банка основной памяти общим объемом 320 Кбайт;
  - банк памяти SRAM3 сопряжен по частоте с ядром CPU1;
  - поддержка режимов retention и shutdown независимо для каждого банка;
  - дополнительный блок памяти в батарейном домене размером 1 Кбайт
- OTP память



- однократно программируемая память для хранения ключей и пользовательских данных;
  - общий размер: 1 Кбайт; 640 байт доступно для пользовательских данных;
- Блок крипто-акселераторов CryptoCell (CRYPTO)
  - поддержка алгоритмов шифрования AES, Stream Ciphers, RSA, DH, ECC;
  - поддержка HASH и HMAC;
  - True Random Number Generator (TRNG);
  - обеспечение доверенной загрузки и отладки;
  - управление жизненным циклом изделия;
- Сервисный блок GMS для контроля начальной загрузки и целостности информации
- Приемник сигналов систем спутниковой навигации (GNSS)
  - прием навигационного сигнала ГЛОНАСС в полосах L1 и L2;
  - прием навигационного сигнала GPS в полосах L1 и L2;
- Интерфейс внешней статической памяти (SMC)
  - поддержка до двух микросхем памяти SRAM, PSRAM, NOR Flash общим объемом до 64 Мбайт;
  - разрядность данных для каждой микросхемы – 16 бит;
  - мультиплексирование адреса и данных;
  - максимальная частота работы интерфейса – 50 МГц;
  - поддержка LCD дисплеев с интерфейсами типа Intel 8080 и Motorola 6800
- Интерфейс внешней Flash памяти Quad SPI (QSPI)
- Интерфейс USB-2.0 OTG (USB)
  - работа в режимах Host и Device;
  - скорость передачи данных;
    - High-Speed (HS, 480-Mbps);
    - Full-Speed (FS, 12-Mbps);
    - Low-Speed (LS, 1.5-Mbps);
  - встроенный PHY, дополнительно поддержка внешнего PHY через интерфейс ULPI;
  - использование встроенного DMA для передачи данных внутри микросхемы;
  - возможность работы как с интегрированным PHY по UTMI+ интерфейсу (не поддерживает LS в режиме девайса), так и с внешним по ULPI интерфейсу
- Интерфейс SD/SDIO/MMC (SDMMC)
  - поддерживаемые стандарты:
    - SD Host Controller Standard Specification Version 3.00;
    - SDIO card specification Version 3.0;
    - SD Memory Card Specification Version 3.01;
    - SD Memory Card Security Specification version 1.01;
    - MMC Specification version 4.51.
  - интерфейс SD/SDIO поддерживает режимы Default speed (DS) и High speed (HS) со скоростью до 25 Мбайт/с;

- скорость интерфейса MMC – до 50 Мбайт/с в режиме SDR, до 100 Мбайт/с в режиме DDR;
  - встроенный контроллер DMA.
- Controller area network (CAN)
  - поддерживаемые стандарты:
    - CAN 2.0B;
    - CAN FD
  - программируемые фильтры сообщений;
  - поддержка Time-stamping:
    - ISO 11898-4 Time-Triggered CAN with partial hardware support;
    - CiA 603 time-stamping
  - совместимость с AUTOSAR
- Два системных контроллера прямого доступа в память (DMA)
- Таймер реального времени (RTC)
- Сторожевые таймеры (WDT, LPWDT)
- Таймеры общего назначения (TIM0, TIM1 DTIM, LPTIM)
- Блок генерации ШИМ сигналов (PWM)
- Многофункциональные таймеры (VTU)
  - два 8-ми разрядных генератора ШИМ сигналов;
  - 16-ти разрядный генератор ШИМ сигнала;
  - 16-ти разрядный счетчик с двумя каналами захвата.
- Универсальные асинхронные трансиверы (UART)
- Аудио интерфейс (I2S)
- Синхронные последовательные интерфейсы (SPI)
- Интерфейсы Inter-Integrated Circuit (I2C)
- Порты ввода/вывода общего назначения (GPIO)
  - четыре порта ввода/вывода общего назначения по 16 линий каждый;
  - индивидуальная настройка направления каждого вывода;
  - поддержка внешних прерываний с программируемым условием срабатывания.
- Отладочный интерфейс JTAG/Serial Wire
- Сенсоры температуры и напряжения
- Тактирование
  - встроенный осциллятор для подключения внешнего кварцевого резонатора с тактовой частотой до 48 МГц;
  - внутренний генератор тактовой частоты 16 МГц;
  - внутренний генератор тактовой частоты 32.768 кГц для таймера реального времени;
  - встроенный осциллятор для подключения внешнего кварцевого резонатора на частоту 32.768 кГц;
  - блок PLL для формирования системной тактовой частоты.

- Питание
  - напряжение основного источника питания: 3.3 В. Напряжение батарейного источника питания: 1.6 – 3.6 В.
  - встроенный DC-DC преобразователь напряжения для питания цифрового ядра;
  - выделенное питание для буферов ввода/вывода интерфейса SD;
- Технологические особенности
  - корпус BGA132, размер 7x7 мм, шаг выводов 0.5 мм;
  - рабочая температура: -40 – +85 °C.

## 2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ

### 2.1 Структурная схема

На рисунке **Рисунок 2.1** показана структурная схема верхнего уровня К1892ВГ1Я. На схеме изображены:

- Основная коммутационная матрица (Main АНВ Bus Matrix) и дополнительная коммутационная матрица банка памяти SRAM3 (SRAM3 АНВ Bus Matrix);
- Мосты интерфейсов АМБА АНВ/АРВ (серые и красные стрелки соответствуют интерфейсам АНВ, голубые – интерфейсам АРВ);
- компоненты TrustZone: Peripheral Protection Controller (PPC), Memory Protection Controller (MPC), Master Security Controller (MSC).
- блоки управления доступом (Access Control Gate, ACG);
- мониторы эксклюзивного доступа к памяти (Exclusive Access Monitor, EAM);
- процессорные подсистемы CPU0 и CPU1;
- подсистема отладки (Debug Subsystem);
- блоки статической памяти SRAM0-3, BKPSRAM;
- подсистема Flash памяти;
- ОТР память;
- контроллеры DMA, периферийные блоки и блоки регистров;
- мультиплексоры портов ввода-вывода (IOMUX);
- блок управления питанием (PMU);
- блок управления тактированием и сбросом.

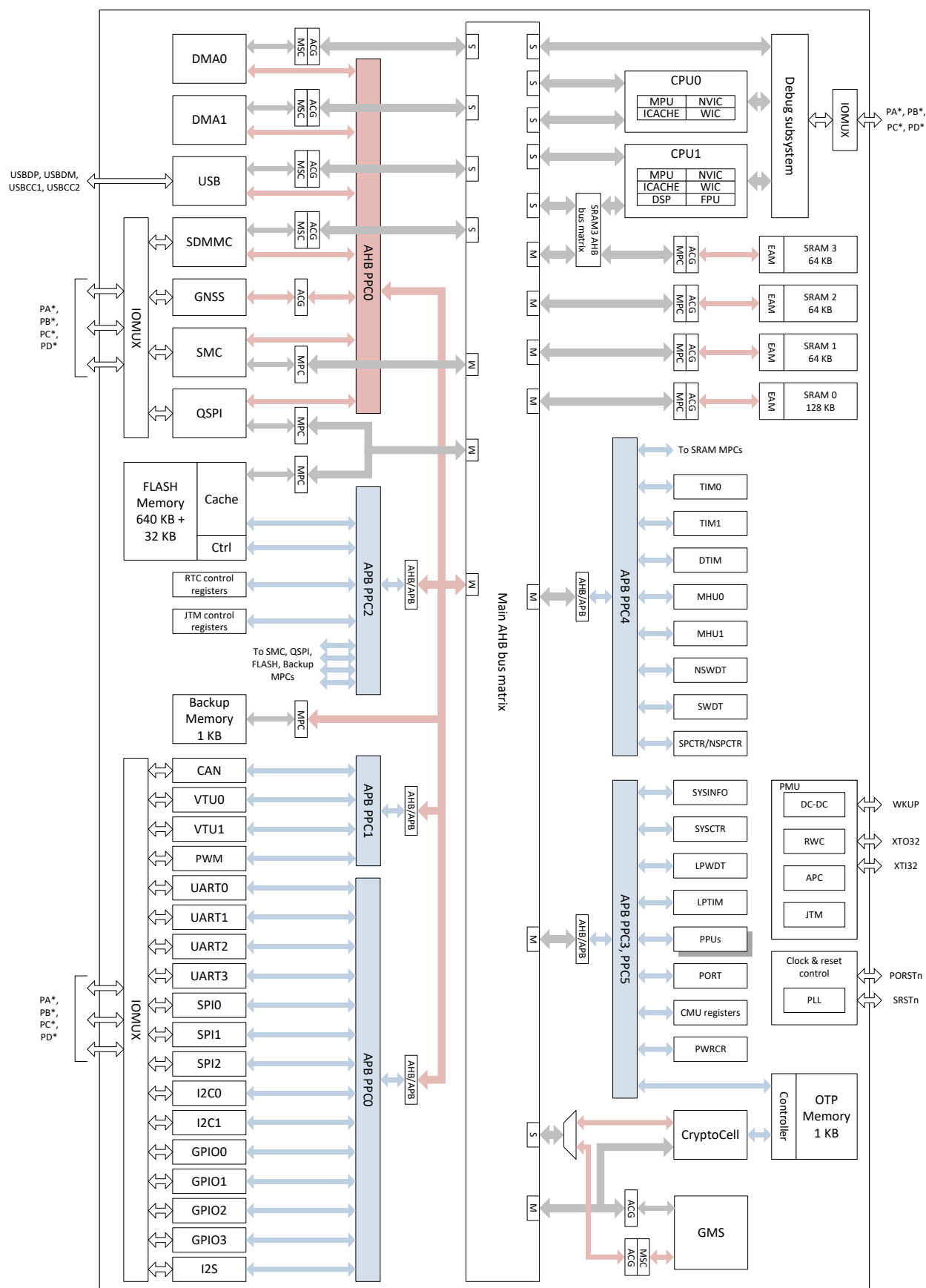
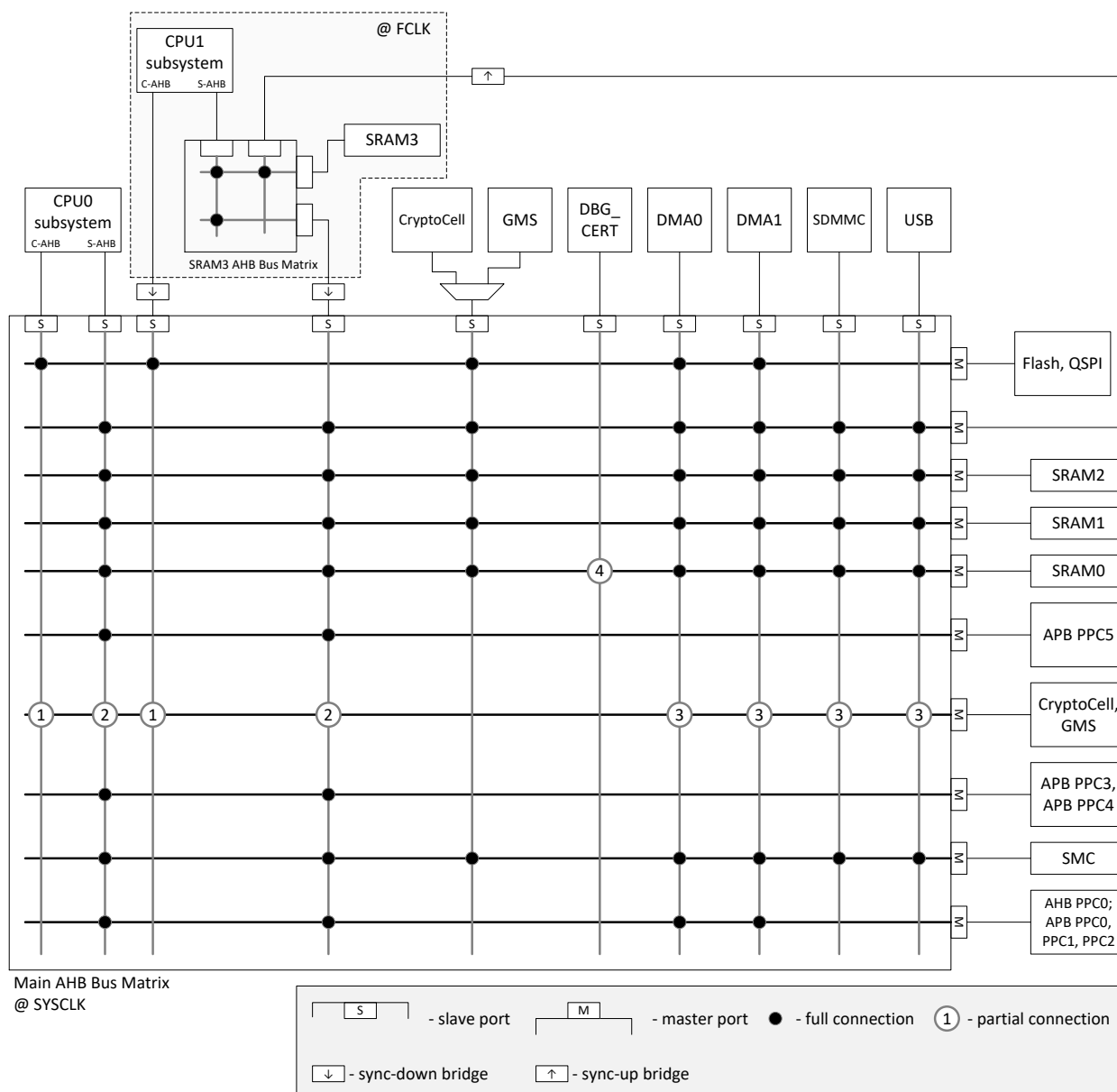


Рисунок 2.1. Структурная схема K1892BG1A

## 2.2 Матрица коммутации

На рисунке Рисунок 2.2 показаны взаимосвязи компонентов системы через основную матрицу коммутации, а также через дополнительную матрицу коммутации CPU1 и памяти SRAM3.



**Рисунок 2.2. Матрица коммутации**

Черный круг на пересечении вертикальной линии master-устройства и горизонтальной линии slave-устройства обозначает наличие доступа master-устройства ко всему диапазону адресов соответствующего slave-устройства. Круг с цифрой обозначает наличие доступа лишь к части адресного пространства slave-устройства:

- 1) Доступ к пользовательской области ОЗУ памяти (0x0E00\_0000 - 0x0E00\_3FFF; 0x1E00\_0000 - 0x1E00\_3FFF).
- 2) Доступ к регистрам блоков CryptoCell и GMS.

- 3) Доступ к регистрам блока GMS;
- 4) По интерфейсу DBG\_CERT доступно только 8 Кбайт памяти SRAM0 по адресам 0x3000\_0000 – 0x3000\_1FFF.

Характеристики основной матрицы коммутации:

- работает на частоте SYSCLK;
- многослойная структура – передача данных по непересекающимся каналам на максимальной скорости;
- round-robin арбитраж;
- нулевая задержка от выбранного master-устройства, один такт задержки при выборе другого master-устройства.

Основная матрица коммутации имеет 10 slave-портов, к которым подключены следующие master-устройства:

- кэшируемые интерфейсы (C-AHB) процессоров CPU0 и CPU1 для доступа к Code области памяти микросхемы;
- системный интерфейс S-AHB процессора CPU0;
- системный интерфейс S-AHB процессора CPU1 через дополнительную матрицу коммутации;
- DMA интерфейсы крипто-ускорителей CryptoCell и GMS (подключаются через мультиплексор);
- интерфейс DBG\_CERT отладочной подсистемы для загрузки сертификата;
- контроллеры прямого доступа в память DMA0 и DMA1;
- контроллер SDMMC;
- контроллер USB.

Основная матрица коммутации имеет 10 master-портов, к которым подключены следующие slave-устройства:

- подсистема Flash памяти и контроллер QSPI;
- банк памяти SRAM3 через дополнительную матрицу коммутации;
- отдельные порты для доступа к банкам памяти SRAM0, SRAM1, SRAM2;
- устройства, подключенные через контроллер APB PPC5;
- порт для доступа к регистрам блока CryptoCell и OTP памяти, а также к регистрам блока GMS;
- устройства, подключенные через контроллеры APB PPC3, PPC4;
- контроллер внешней памяти SMC;
- устройства, подключенные через контроллеры AHB PPC0, APB PPC0, APB PPC1, APB PPC2.

Дополнительная матрица служит для коммутации системного интерфейса CPU1, банка памяти SRAM3 и основной матрицы. Дополнительная матрица коммутации работает на частоте FCLK, так же как CPU1 и SRAM3. Мосты синхронизации между частотами

SYSCLK и FCLK (sync-up, sync-down bridge на рисунке) вносят задержку в несколько тактов:

- FCLK → SYSCLK: 3 такта SYSCLK + 2 такта FCLK;
- SYSCLK → FCLK:
  - чтение:
    - 4 такта FCLK + 1 такт SYCLK, если  $F_{FCLK}/F_{SYSCLK} < 4$ ;
    - 2 такта SYSCLK, если  $F_{FCLK}/F_{SYSCLK} \geq 4$ ;
  - запись:
    - 4 такта FCLK + 2 такта SYCLK, если  $F_{FCLK}/F_{SYSCLK} < 4$ ;
    - 3 такта SYSCLK, если  $F_{FCLK}/F_{SYSCLK} \geq 4$ .

Мост синхронизации sync-down bridge для интерфейса S-AHB CPU1 имеет буфер записи на одно слово. Буфер может сохранять одну буферизируемую неэксклюзивную транзакцию записи, если Slave-устройство занято. При этом CPU1 сразу же получает флаг готовности. Если буферизированная транзакция вызывает ошибку в slave устройстве, то генерируется прерывание BRG.

Дополнительная матрица не вносит задержку между CPU1 и SRAM3.



## 2.3 Подсистема CPU

Микросхема содержит два процессора Cortex-M33 – CPU0 и CPU1.

Cortex-M33 – это 32-х разрядное процессорное ядро, спроектированное для применения в микроконтроллерах с малым потреблением. Процессор обеспечивает высокую производительность, быструю обработку прерываний и широкие возможности отладки, включая точки останова и вывод трассы.

Основные возможности Cortex-M33 в составе K1892ВГ1Я:

- Технология Arm® TrustZone® и Secure расширение ARMv8-M.
- Блок Memory Protection Unit (MPU), поддерживающий 8 регионов для Secure приложений и 8 регионов для Non-secure приложений.
- Блок Secure Attribute Unit (SAU) поддерживающий 8 регионов памяти.
- Блок Floating Point Unit (FPU) для выполнения операций с плавающей точкой одинарной точности (только CPU1).
- DSP-расширение системы инструкций для операций цифровой обработки сигналов (только CPU1).
- Низкопотребляющие режимы работы.

На рисунке Рисунок 2.3 показана структурная схема подсистем процессоров CPU0 и CPU1. Модуль FPU имеется только в CPU1.

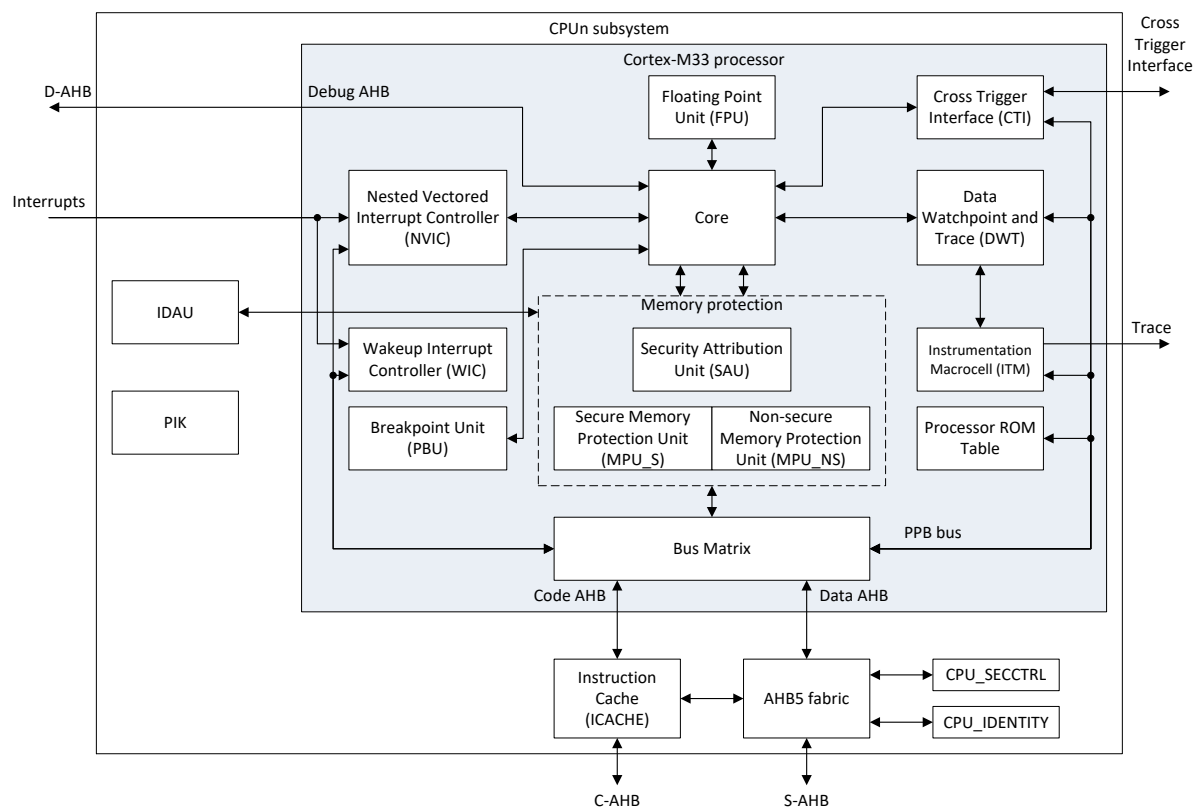


Рисунок 2.3. Структура подсистемы CPU

Подсистема процессора содержит следующие блоки:

- процессор Cortex-M33;
- кэш инструкций (ICACHE);
- блоки регистров CPU\_SECCTRL, CPU\_IDENTITY;
- локальный коммутатор интерфейсов АНВ5;
- блок Implementation Defined Attribution Unit (IDAU);
- блок управления питанием Power Integration Kit (PIK).

Основные интерфейсы подсистемы:

- Интерфейс System АНВ (S-АНВ). Используется для загрузки инструкций и доступа к данным, расположенным в областях SRAM, Peripheral, External RAM, External Device и System адресного пространства ARMv8-M.
- Интерфейс Code АНВ (C-АНВ). Используется для загрузки инструкций и доступа к данным, расположенным в области Code адресного пространства ARMv8-M.
- Интерфейс Debug АНВ (D-АНВ). Интерфейс предназначен для взаимодействия с устройствами отладочной подсистемы микросхемы. Обращения по интерфейсу D-АНВ распределяются внутренним коммутатором Cortex-M33 на интерфейсы S-АНВ, C-АНВ либо PPB в зависимости от адреса.
- Интерфейс прерываний от устройств системы.
- Интерфейс Trace для вывода трассы.
- Интерфейс Cross Trigger Interface для подключения к блоку CTI отладочной подсистемы.

Подробная информация о Cortex-M33 приведена в следующих документах:

- Arm® Cortex®-M33 Processor Technical Reference Manual.
- Arm®v8-M Architecture Reference Manual.

## 2.4 Встроенная SRAM память

Микросхема содержит 320 Кбайт SRAM памяти, разделенной на четыре банка, обозначаемых как SRAM<sub>n</sub> (n = 0, 1, 2, 3).

Память SRAM<sub>n</sub> имеет следующие особенности:

- Размеры банков: SRAM0 – 128 Кбайт, SRAM1-3 – 64 Кбайт.
- Банки SRAM0-2 работают на частоте SYSCLK, банк SRAM3 – на частоте FCLK.
- Все банки формируют непрерывный диапазон адресов в адресном пространстве системы.
- Каждый банк памяти независимо от других банков может быть переведен в малопотребляющий режим retention (с сохранением данных), либо выключен (с потерей данных).
- Поддерживаются обращения размером 8, 16 и 32 бита, а также невыровненные обращения.
- Каждый банк памяти поддерживает эксклюзивный доступ со стороны CPU с помощью инструкций LDREX, STREX.

Дополнительно в батарейном домене питания расположен банк памяти BKPSRAM объемом 1 Кбайт. Питание BKPSRAM осуществляется от вывода VDDA либо, при отсутствии напряжения на VDDA, от вывода VBAT через встроенный LDO регулятор напряжения. Память BKPSRAM поддерживает только выровненные 32-х разрядные обращения.

Примечание: в текущей версии микросхемы питание от VBAT не реализовано

## 2.5 Обзор Flash памяти

Микросхема имеет встроенную Flash память, состоящую из двух разделов:

- основной раздел – 640 Кбайт;
- системный раздел – 32 Кбайт.

Основной раздел предназначен для хранения инструкций и данных. Доступ к основному разделу (Secure/Non-secure) конфигурируется с помощью блока FLASH\_MPC.

Системный раздел предназначен для хранения инструкций и данных первичного доверенного загрузчика. Раздел доступен только для Secure обращений. Дополнительно, системный раздел может быть защищен от изменений с помощью бита FLASH\_SYS\_RO\_EN регистра HOST\_DCU\_EN0 блока CryptoCell.

Разделы Flash памяти состоят из страниц размером 8 Кбайт. Операция стирания может выполняться над одной страницей (ERASE) либо над всеми страницами (MASS ERASE).

Основной раздел Flash памяти имеет две резервные страницы для замещения страниц, содержащих поврежденные ячейки памяти. Информация о поврежденных страницах

заносятся в специальную область Flash памяти при технологическом тестировании. Считывание этой информации и замещение поврежденных страниц выполняется автоматически контроллером Flash.

## 2.6 Обзор ОТР памяти

Микросхема содержит блок однократно-программируемой (ОТР) памяти объемом 1 Кбайт.

ОТР память предназначена для хранения следующих данных:

- ключи и другая информации блока CryptoCell;
- настройки адреса загрузки и блокировки доступа;
- пользовательские данные (640 байт).

Подробное описание ОТР памяти см. в разделе 26 «ОТР: Однократно программируемая память».

## 2.7 Система обеспечения безопасности

Система безопасности микросхемы основана на технологии Arm® TrustZone® и Secure-расширении архитектуры ARMv8-M.

### 2.7.1 Атрибуты памяти

Адреса в памяти системы могут иметь один из следующих атрибутов безопасности:

- Secure – доверенные адреса памяти и периферии, которые доступны только для Secure программы или Secure master-устройства. Транзакции по таким адресам обозначаются как Secure на интерфейсе АНВ. Попытки доступа со стороны Non-secure программы блокируются и вызывают исключение.
- Non-secure Callable (NSC) – специальный тип Secure атрибута. Обозначает доверенную область памяти, в которую возможны переходы из Non-secure состояния CPU. Для этого точка входа в NSC области памяти должна содержать инструкцию SG.
- Non-secure (NS) – недоверенная область памяти, доступная для любого приложения, запущенного на CPU, а также для любого master-устройства в системе. Транзакции по таким адресам обозначаются как Non-secure на интерфейсе АНВ.

В процессоре атрибуты памяти назначаются с помощью блоков SAU и IDAU. Для остальных master-устройств в системе атрибуты назначаются блоками IDAU.

Блок SAU (Secure Attribution Unit) входит в состав MPU процессора и позволяет программно настроить атрибуты до 8 регионов памяти.

Блоки IDAU (Implementation Defined Attribution Unit) подключены к интерфейсам CPU, а также через блоки MSC к master-интерфейсам контроллеров DMA, SDMMC, USB, GMS. IDAU разделяет все адресное пространство системы на чередующиеся Non-secure и Secure области памяти размером 256 Мбайт (Таблица 2.1). Исключением являются диапазоны адресов, предназначенные для интерфейса PPB процессора, а также для отладочных устройств. Для этих адресов атрибуты безопасности не применимы. Разделение памяти с помощью IDAU фиксировано и не может быть изменено.

В таблице таблица 2.1 показана обобщенная карта памяти системы и соответствующие атрибуты IDAU.

**Таблица 2.1. Атрибуты безопасности IDAU**

Назначение области памяти	Диапазон адресов	Атрибут безопасности IDAU
Code	0x00000000 – 0x0FFFFFFF	Non-secure
	0x10000000 – 0x1FFFFFFF	Secure/NSC
SRAM	0x20000000 – 0x2FFFFFFF	Non-secure
	0x30000000 – 0x3FFFFFFF	Secure/NSC
Peripheral	0x40000000 – 0x4FFFFFFF	Non-secure
	0x50000000 – 0x5FFFFFFF	Secure
External RAM	0x60000000 – 0x6FFFFFFF	Non-secure
	0x70000000 – 0x7FFFFFFF	Secure
	0x80000000 – 0x8FFFFFFF	Non-secure
	0x90000000 – 0x9FFFFFFF	Secure
External Device	0xA0000000 – 0xAFFFFFFF	Non-secure
	0xB0000000 – 0xBFFFFFFF	Secure
	0xC0000000 – 0xCFFFFFFF	Non-secure
	0xD0000000 – 0xDFFFFFFF	Secure
System PPB	0xE0000000 – 0xE0FFFFFF	Атрибуты не применяются
System	0xE0100000 – 0xEFFFFFFF	Non-secure
System debug	0xF0000000 – 0xF0FFFFFF	Атрибуты не применяются
System	0xF0100000 – 0xFFFFFFFF	Secure

Настройка атрибутов NSC для областей Code и SRAM выполняется с помощью регистра NSCCFG блока SPCTR.

Для процессора атрибуты безопасности SAU и IDAU комбинируются. Результирующим является атрибут с более высоким уровнем безопасности (Таблица 2.2).

**Таблица 2.2. Комбинирование атрибутов SAU и IDAU**

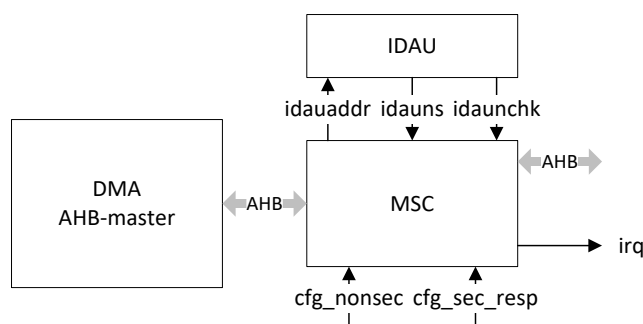
IDAU	SAU	Результат
S	X	S
X	S	S
NS	NSC	NSC
NS	NS	NS
NSC	NS	NSC
Обозначения: S – Secure NS – Non-secure NSC – Non-secure Callable X – не имеет значения		

Адресные пространства большинства блоков памяти и периферийных устройств повторяются в карте памяти два раза для Secure и Non-secure областей для возможности обращения к ним как с помощью Secure транзакций, так и с помощью Non-secure транзакций. Исключением являются некоторые устройства, к которым возможен доступ с помощью только Secure либо только Non-secure транзакций.

## 2.7.2 Безопасность контроллеров DMA

Микросхема содержит несколько контроллеров прямого доступа в память (DMA), являющихся master-устройствами на шине АНВ. Каждый контроллер DMA может формировать как Secure, так и Non-secure транзакций в зависимости от используемого адреса.

Транзакции от блоков DMA0, DMA1, а также DMA контроллеров в составе блоков SDMMC, USB, GMS обрабатываются с помощью блоков Master Security Controller (MSC) и IDAU, подключенных к соответствующим master-интерфейсам (Рисунок 2.4).



**Рисунок 2.4. Подключение MSC**

Результат обработки транзакции в блоке MSC (передается ли транзакция к slave-устройству или блокируется) зависит от значений сигналов **idaunchk**, **idauns**, **cfg\_nonsec** (Таблица 2.3).

Значение **idaunchk** показывает, применимы ли атрибуты безопасности к адресу транзакции (**idaunchk** = 1) или нет (**idaunchk** = 0).

Значение **idauns** обозначает атрибут адреса транзакции, назначенный блоком IDAU – Secure (**idauns** = 0) или Non-secure (**idauns** = 1).

Значение **cfg\_nonsec** задается в регистре SPCTR\_NSMSC и определяет уровень доступа master-устройства.

**Таблица 2.3. Обработка транзакции в MSC**

idaunchk	idauns	Результат	
		cfg_nonsec = 0	cfg_nonsec = 1
0	0	Транзакция передается, HNONSEC = 0	Транзакция блокируется
0	1	Транзакция передается, HNONSEC = 1	Транзакция передается, HNONSEC = 1
1	0	Транзакция передается, HNONSEC = 0	Транзакция передается, HNONSEC = 1
	1		

Параметр **cfg\_sec\_resp** задается в регистре SPCTR\_SECRESPCFG и определяет ответ, формируемый блоком MSC на интерфейсе АНВ при блокировке транзакции:

- если **cfg\_sec\_resp** = 1, то на интерфейсе АНВ возвращается ошибка (HRESP = 1);
- если **cfg\_sec\_resp** = 0, то транзакция обрабатывается как RAZ/WI (Read As Zero and Write Ignored) – на интерфейсе АНВ формируется нормальный ответ (HRESP = 0), запись игнорируется, при чтении возвращаются нули.

При блокировке транзакции блок MSC также генерирует прерывание. Проверка статуса и управление прерыванием выполняются с помощью регистров SPCTR\_SECMSCINTSTAT, SPCTR\_SECMSCINTCLR, SPCTR\_SECMSCINTEN.

Настройка параметров master-интерфейса блока CryptoCell выполняется с помощью регистров этого блока.

### 2.7.3 Безопасность периферийных устройств

Все блоки регистров и периферийные устройства K1892ВГ1Я можно разделить на несколько групп:

- Secure устройства с фиксированным уровнем доверенности. Такие устройства (например, ICACHE) доступны только для Secure обращений.
- Интерфейсы APB-R и APB-C блока CryptoCell, для которых уровень безопасности настраивается с помощью регистров блока CryptoCell.
- Блоки GPIOx (x = 0 - 3). Настройка безопасности доступа выполняется с помощью регистров соответствующего блока.
- Все остальные устройства подключены к основной матрице коммутации через блоки Peripheral Protection Controller (PPC).

В зависимости от типа интерфейса APB или AHB блоки PPC обозначаются как APB PPC и AHB PPC соответственно. К одному контроллеру PPC может быть подключено одно или несколько периферийных устройств. Соответствие блоков PPC и подключенных к ним устройств отображено в таблицах Таблица 2.6 – Таблица 2.9.

Уровень безопасности устройств, подключенных к PPC, настраивается с помощью регистров блока SPCTR. Некоторые устройства имеют фиксированные настройки и доступны только для Secure обращений (например, PPU) либо только для Non-secure обращений (например, NSWDT).

При несовпадении атрибута безопасности входящей транзакции с уровнем безопасности устройства блок PPC блокирует транзакцию и выполняет следующие действия:

- Если бит SECRESPCFG в регистре SPCTR\_SECRESPCFG установлен, то на интерфейсе AHB/APB возвращается ошибка (HRESP/PSLVERR = 1).
- Если бит SECRESPCFG в регистре SPCTR\_SECRESPCFG сброшен, то на интерфейсе AHB/APB формируется нормальный ответ (HRESP/PSLVERR = 0); при этом транзакция записи игнорируется, транзакция чтения возвращает нули.
- Устанавливается прерывание PPC. Проверка статуса и управление прерыванием выполняются с помощью регистров SPCTR\_SECPPCINTSTAT, SPCTR\_SECPPCINTCLR, SPCTR\_SECPPCINTEN.

Блок PPC также выполняет фильтрацию транзакций по уровню привилегированности. Непривилегированные транзакции (HPROT[1]/PPROT[0] = 0) для устройств, настроенных на прием привилегированных транзакций, блокируются – транзакции записи игнорируются, транзакции чтения возвращают нули. Прерывание PPC при этом не генерируется. Настройка привилегированности доступа для Secure обращений выполняется в блоке SPCTR, для Non-secure обращений – в блоке NSPCTR.

## 2.7.4 Безопасность блоков памяти

Для настройки параметров безопасности внутренней SRAM памяти и основного раздела Flash памяти, а также внешней памяти, подключаемой по интерфейсам QSPI и SMC, используются модули Memory Protection Controller (MPC).

Подробное описание MPC дано в разделе 25 «MPC: КОНТРОЛЛЕР БЕЗОПАСНОСТИ ПАМЯТИ».

Системный раздел Flash памяти доступен только для Secure транзакций.

Тип доступа к OTP памяти настраивается с помощью регистров блока CryptoCell (раздел 27 «CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCELL»).



### 2.7.5 Начальное состояние

После сброса все устройства системы, включая CPU, PPC, MSC, MPC, находятся в состоянии Secure. Блоки SAU выключены, при этом бит ALLNS = 0. Это означает, что все адресное пространство системы считается Secure для процессоров независимо от атрибутов IDAU.

### 2.7.6 Жизненный цикл микросхемы

Микросхема может находиться в одном из четырех состояний жизненного цикла (далее LCS, Life Cycle State):

- Chip Manufacturing (CM);
- Device Manufacturing (DM);
- Deployed/Secure Enabled (SECURE);
- Return Merchandise Authorization (RMA).

Управление LCS осуществляется блоком CryptoCell и основано на состоянии OTP памяти (см. раздел 27 «CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCELL»).

Значение LCS определяет параметры доступа к информации в OTP памяти, а также доступ к другим возможностям системы, например, к внешней отладке.

### 2.7.7 Загрузка сертификата

Подсистема отладки K1892BG1Я имеет выделенный АНВ интерфейс (далее интерфейс DBG\_CERT) для доступа к области памяти размером 8 Кбайт, расположенной в банке SRAM0 по адресам 0x3000\_0000 – 0x3000\_1FFF (далее окно памяти).

После «холодного» сброса интерфейс открыт только для записи в память.

Доступ на чтение указанного окна в памяти может быть разрешен с помощью бита CERTREADEN в регистре SYSCTR\_SCSECCTRL. С помощью бита CERTDISABLE того же регистра доступ к памяти может быть полностью закрыт.

После установки бита CERTDISABLE доступ по интерфейсу DBG\_CERT может быть восстановлен только после «холодного» сброса системы.

Интерфейс DBG\_CERT использует порт доступа CPU0 АНВ-АР в отладочной подсистеме. Перед использованием данного интерфейса необходимо включить ядро CPU0, используя блок CPU0 GPR, что также приведет к включению системного домена PD\_SYS.

Интерфейс DBG\_CERT позволяет внешнему отладчику, подключенному к микросхеме с помощью интерфейса JTAG/SWD (далее отладчик), разместить сертификат либо запрос данных в памяти SRAM0 перед загрузкой процессора. Для этого процессор должен

удерживаться в сбросе с помощью вывода SRSTn. После снятия сброса с процессора доверенный загрузчик может проверить содержимое памяти и определить наличие сертификата или запроса.

Данное руководство не описывает конкретную реализацию протокола общения внешнего отладчика и процессора через интерфейс DBG\_CERT, также, как и способ генерации и проверки сертификата.

Для генерирования сертификата может потребоваться считывание информации о системе. Далее приводится пример последовательности действий для запроса отладчиком идентификационной информации о системе:

1. Отладчик устанавливает низкий логический уровень на выводе SRSTn для сброса системы и удерживает SRSTn в таком состоянии для предотвращения загрузки процессора.
2. Отладчик записывает в окно SRAM0 запрос необходимой информации. При этом нужно использовать Secure запись.
3. Отладчик снимает сброс SRSTn для загрузки процессора.
4. Процессор проверяет окно SRAM0 и обнаруживает запрос данных.
5. Процессор стирает содержимое окна SRAM0 записью нулей.
6. Процессор записывает в память идентификатор системы и флаг готовности данных.
7. Процессор устанавливает бит CERTREADEN для разрешения чтения окна.
8. Процессор выполняет инструкцию WFI при выключенных прерываниях либо входит в бесконечный цикл.
9. Отладчик проверяет окно SRAM0 до тех пор, пока не будет установлен флаг готовности данных, а затем считывает идентификационную информацию.

Далее приводится пример последовательности действий для загрузки сертификата:

1. Отладчик устанавливает низкий логический уровень на выводе SRSTn для сброса системы и удерживает SRSTn в таком состоянии для предотвращения загрузки процессора.
2. Отладчик записывает в окно SRAM0 сертификат, сгенерированный с использованием ранее полученной идентификационной информации. При этом нужно использовать Secure запись.
3. Отладчик снимает сброс SRSTn для загрузки процессора.
4. Процессор проверяет окно SRAM0 и обнаруживает сертификат.
5. Процессор закрывает доступ отладчика к окну памяти установкой бита CERTDISABLE.
6. Процессор использует сертификат, например, для смены состояния LCS или настройки доступа к ресурсам отладки.
7. После этого процессор продолжает загрузку.

#### Примечание

Данный механизм может быть использован для взаимодействия отладчика и процессора сразу после снятия сброса SRSTn. Он не предназначен для использования после загрузки и во время выполнения основного приложения.

При работе с интерфейсом DBG\_CERT должны соблюдаться следующие правила:

- После включения процессора CPU0 и до начала проверки сертификата либо до продолжения загрузки необходимо закрыть доступ по интерфейсу DBG\_CERT путем установки бита CERTDISABLE и ожидания установки бита статуса CERTDISABLED.
- Домены PD\_SYS, PD\_CPU0 и PD\_SRAM0 должны оставаться в состоянии ON до установки CERTDISABLED. Процессор CPU0 не должен выполнять инструкцию WFI и переходить в режим SleepDeep.
- После проверки сертификата 8 Кбайт окно памяти должно быть стерто до того, как приложению будет позволено использовать данную память.
- Перед разрешением доступа на чтение содержимое окна памяти должно быть стерто.
- При использовании окна памяти для загрузки и проверки сертификата ни доверенное приложение, ни отладчик не должны инициировать сброс системы через регистр AIRCR процессора.

## 2.7.8 Контроль отладочного доступа

Для управления доступом внешнего отладчика к ресурсам системы используются сигналы DBGEN, NIDEN, SPIDEN, SPNIDEN.

В следующих таблицах показано влияние перечисленных сигналов на возможности инвазивной (точки останова, доступ к адресному пространству системы) и неинвазивной (трасса) отладки.

**Таблица 2.4. Управление инвазивной отладкой**

DBGEN	SPIDEN	Статус инвазивной отладки	Доступные устройства
0	X	Выключена	Нет
1	0	Включена	Только недоверенные
	1		Все

**Таблица 2.5. Управление неинвазивной отладкой**

DBGEN	NIDEN	SPIDEN	SPNIDEN	Доступные устройства
0	0	X	X	Нет
		0	0	Только недоверенные
			1	Все
			X	
1	X	0	0	Только недоверенные
		1	1	Все
			X	

Значения сигналов контроля отладочного доступа устанавливаются с помощью регистра CRYPTO\_HOST\_DCU\_EN0 блока CryptoCell. Исходное значение данного регистра зависит от текущего состояния LCS (см. раздел 27 «CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCELL»).

При LCS = SECURE отладочный доступ закрыт по умолчанию. Для разрешения доступа необходимо использовать механизм, описанный в пункте «Загрузка сертификата».

## 2.8 Карта памяти

### 2.8.1 Введение

В таблицах Таблица 2.6 – Таблица 2.10 приведена карта памяти верхнего уровня K1892ВГ1Я. Карта памяти разделена на области Code, SRAM, Peripheral, External RAM, External Device и System согласно архитектуре ARMv8-M. Каждая область памяти, кроме System, состоит из двух частей – для Secure и Non-secure доступа. также в таблицах приводятся названия контроллеров, отвечающих за безопасность блоков памяти или периферийных устройств.

Если не сказано другого, все запросы к неразмеченным (зарезервированным) областям памяти завершаются ошибкой шины. Любой запрос, приводящий к нарушению безопасности, обрабатывается как RAZ/WI (Read As Zero and Write Ignored) или завершается с ошибкой в зависимости от настройки блоков MPC или PPC.

### 2.8.2 Область Code

Область Code предназначена для хранения инструкций и данных. Доступ CPU к этой области выполняется через интерфейс C-AHB.

Область Code имеет следующий диапазон адресов:

- Non-secure: 0x0000\_0000 – 0x0FFF\_FFFF
- Secure: 0x1000\_0000 – 0x1FFF\_FFFF

Диапазон 0x1000\_0000 – 0x1FFF\_FFFF может быть сконфигурирован как Non-secure Callable с помощью регистра SPCTR\_NSCCFG.

**Таблица 2.6. Карта памяти K1892ВГ1Я (область Code)**

Диапазон адресов		Размер	Назначение	Доступ	Контроллер безопасности
От	До				
0x0000_0000	0x0009_FFFF	640KB	Flash память (основной раздел)	NS	FLASH_MPC
0x000A_0000	0x07FF_FFFF	-	Резерв	-	-
0x0800_0000	0x0BFF_FFFF	64MB	Внешняя QSPI-память	NS	QSPI_MPC
0x0C00_0000	0x0DFF_FFFF	-	Резерв	-	-
0x0E00_0000	0x0E00_3FFF	16KB	Пользовательская OTP память	NS	Встроен в CryptoCell
0x0E00_4000	0x0FFF_FFFF	-	Резерв	-	-
0x1000_0000	0x1009_FFFF	640KB	Flash память (основной раздел)	S	FLASH_MPC
0x100A_0000	0x101F_FFFF	-	Резерв	-	-
0x1020_0000	0x1020_7FFF	32KB	Flash память (системный раздел)	S	-
0x1020_8000	0x17FF_FFFF	-	Резерв	-	-
0x1800_0000	0x1BFF_FFFF	64MB	Внешняя QSPI-память	S	QSPI_MPC
0x1C00_0000	0x1DFF_FFFF	-	Резерв	-	-
0x1E00_0000	0x1E00_3FFF	16KB	Пользовательская OTP память	S	Встроен в CryptoCell
0x1E00_4000	0x1FFF_FFFF	-	Резерв	-	-
Примечания: S – Secure NS – Non-secure					

### 2.8.3 Область SRAM

Область SRAM предназначена для хранения инструкций и данных. Доступ CPU к этой области выполняется через интерфейс S-AHB.

Область SRAM имеет следующий диапазон адресов:

- Non-secure: 0x2000\_0000 – 0x2FFF\_FFFF
- Secure: 0x3000\_0000 – 0x3FFF\_FFFF

Диапазон 0x3000\_0000 – 0x3FFF\_FFFF может быть сконфигурирован как Non-secure Callable с помощью регистра SPCTR\_NSCCFG.

**Таблица 2.7. Карта памяти K1892ВГ1Я (область SRAM)**

Диапазон адресов		Размер	Назначение	Доступ	Контроллер безопасности
От	До				
0x2000_0000	0x2001_FFFF	128KB	SRAM0	NS	SRAM0_MPC
0x2002_0000	0x2002_FFFF	64KB	SRAM1	NS	SRAM1_MPC
0x2003_0000	0x2003_FFFF	64KB	SRAM2	NS	SRAM2_MPC
0x2004_0000	0x2004_FFFF	64KB	SRAM3	NS	SRAM3_MPC
0x2005_0000	0x2FFF_FFFF	-	Резерв	-	-
0x3000_0000	0x3001_FFFF	128KB	SRAM0	S	SRAM0_MPC
0x3002_0000	0x3002_FFFF	64KB	SRAM1	S	SRAM1_MPC
0x3003_0000	0x3003_FFFF	64KB	SRAM2	S	SRAM2_MPC
0x3004_0000	0x3004_FFFF	64KB	SRAM3	S	SRAM3_MPC
0x3005_0000	0x3FFF_FFFF	-	Резерв	-	-

Примечания:  
S – Secure  
NS – Non-secure

## 2.8.4 Область Peripheral

Область Peripheral имеет следующий диапазон адресов:

- Non-secure: 0x4000\_0000 – 0x4FFF\_FFFF
- Secure: 0x5000\_0000 – 0x5FFF\_FFFF

В таблице Таблица 2.8 приводятся смещения адреса относительно базового адреса 0x4000\_0000 для Non-secure запросов и 0x5000\_0000 для Secure запросов. Например, для Secure доступа к таймеру LPTIM со смещением 0x2\_F000 полный адрес равен 0x5002\_F000.

Транзакции по зарезервированным адресам, попадающим в диапазон адресов контроллеров APB PPC, обрабатываются как RAZ/WI.

**Таблица 2.8. Карта памяти K1892ВГ1Я (область Peripheral)**

Смещение адреса		Размер	Назначение		Контроллер безопасности
От	До		Non-secure	Secure	
0x000_0000	0x000_0FFF	4KB	TIM0		APB PPC5
0x000_1000	0x000_1FFF	4KB	TIM1		
0x000_2000	0x000_2FFF	4KB	DTIM		
0x000_3000	0x000_3FFF	4KB	MHU0		
0x000_4000	0x000_4FFF	4KB	MHU1		
0x000_5000	0x000_FFFF	-	Резерв		
0x001_0000	0x001_FFFF	64KB	Устройства подсистемы CPU		См. п. 2.8.8
0x002_0000	0x002_0FFF	4KB	SYSINFO		APB PPC5
0x002_1000	0x002_1FFF	4KB	-	SYSCTR	
0x002_2000	0x002_2FFF	4KB	-	SYS_PPU	
0x002_3000	0x002_3FFF	4KB	-	CPU0_PPU	
0x002_4000	0x002_4FFF	4KB	Резерв		
0x002_5000	0x002_5FFF	4KB	-	CPU1_PPU	
0x002_6000	0x002_6FFF	4KB	Резерв		
0x002_7000	0x002_7FFF	8KB	-	CRYPTO_PPU	
0x002_8000	0x002_8FFF	4KB	Резерв		
0x002_9000	0x002_9FFF	4KB	-	DEBUG_PPU	
0x002_A000	0x002_AFFF	4KB	-	SRAM0_PPU	

Смещение адреса		Размер	Назначение		Контроллер безопасности
От	До		Non-secure	Secure	
0x002_B000	0x002_BFFF	4KB	-	SRAM1_PPU	
0x002_C000	0x002_CFFF	4KB	-	SRAM2_PPU	
0x002_D000	0x002_DFFF	4KB	-	SRAM3_PPU	
0x002_E000	0x002_EFFF	4KB	-	LPWDT	
0x002_F000	0x002_FFFF	4KB	LPTIM		
0x003_0000	0x003_0FFF	4KB	IOCTR		APB PPC3
0x003_1000	0x003_1FFF	4KB	CLKCTR		
0x003_2000	0x003_2FFF	4KB	PWRCTR		
0x003_3000	0x003_3FFF	4KB	-	OTPCTR	
0x003_4000	0x003_4FFF	4KB	-	GMS_PPU	
0x003_5000	0x003_5FFF	4KB	-	GNSS_PPU	
0x003_5000	0x003_FFFF	-	Резерв		
0x004_0000	0x007_FFFF	-	Резерв		-
0x008_0000	0x008_0FFF	4KB	NSPCTR	SPCTR	APB PPC4
0x008_1000	0x008_1FFF	4KB	NSWDT	SWDT	
0x008_2000	0x008_2FFF	-	Резерв		
0x008_3000	0x008_3FFF	4KB	-	SRAM0 MPC	
0x008_4000	0x008_4FFF	4KB	-	SRAM1 MPC	
0x008_5000	0x008_5FFF	4KB	-	SRAM2 MPC	
0x008_6000	0x008_6FFF	4KB	-	SRAM3 MPC	
0x008_7000	0x008_7FFF	-	Резерв		Встроен в CryptoCell
0x008_8000	0x008_BFFF	16KB	Регистры CryptoCell		
0x008_C000	0x008_FFFF	-	Резерв		-
0x009_0000	0x009_0FFF	4KB	-	GMS	
0x009_1000	0x009_FFFF	-	Резерв		-
0x00A_0000	0x00F_FFFF	-	Резерв		-
0x010_0000	0x010_0FFF	4KB	UART0		APB PPC0
0x010_1000	0x010_1FFF	4KB	UART1		
0x010_2000	0x010_2FFF	4KB	UART2		
0x010_3000	0x010_3FFF	4KB	UART3		
0x010_4000	0x010_4FFF	4KB	SPI0		
0x010_5000	0x010_5FFF	4KB	SPI1		
0x010_6000	0x010_6FFF	4KB	SPI2		
0x010_7000	0x010_7FFF	4KB	I2C0		
0x010_8000	0x010_8FFF	4KB	I2C1		
0x010_9000	0x010_9FFF	4KB	I2S		
0x010_A000	0x010_AFFF	4KB	GPIO0		
0x010_B000	0x010_BFFF	4KB	GPIO1		
0x010_C000	0x010_CFFF	4KB	GPIO2		
0x010_D000	0x010_DFFF	4KB	GPIO3		
0x010_E000	0x010_EFFF	4KB	Резерв		
0x010_F000	0x010_FFFF	4KB	Резерв		
0x011_0000	0x011_0FFF	4KB	CAN		APB PPC1
0x011_1000	0x011_1FFF	4KB	PWM		
0x011_2000	0x011_2FFF	4KB	VTU0		
0x011_3000	0x011_3FFF	4KB	VTU1		
0x011_4000	0x01F_FFFF	-	Резерв		
0x020_0000	0x027_FFFF	512KB	USB		AHB PPC0
0x028_0000	0x028_0FFF	4KB	Регистры QSPI		
0x028_1000	0x028_1FFF	4KB	GNSS		
0x028_2000	0x028_3FFF	8KB	Регистры SMC		
0x028_4000	0x028_4FFF	4KB	SDMMC		
0x028_5000	0x028_5FFF	4KB	DMA0		



Смещение адреса		Размер	Назначение		Контроллер безопасности
От	До		Non-secure	Secure	
0x028_6000	0x028_6FFF	4KB	DMA1		
0x028_7000	0x02F_FFFF	-	Резерв		
0x030_0000	0x030_03FF	1KB	BKPSRAM		BKPSRAM_MPC
0x030_0400	0x03F_FFFF	-	Резерв		-
0x040_0000	0x040_1FFF	8KB	-	FCTR	APB PPC2
0x040_2000	0x040_2FFF	4KB	-	FCACHE	
0x040_3000	0x040_3FFF	4KB	RWC		
0x040_4000	0x040_4FFF	4KB	JTM		
0x040_5000	0x040_BFFF	-	Резерв		
0x040_C000	0x040_CFFF	4KB	-	FLASH_MPC	
0x040_D000	0x040_DFFF	4KB	-	QSPI_MPC	
0x040_E000	0x040_EFFF	4KB	-	SMC_MPC	
0x040_F000	0x040_FFFF	4KB	-	BKP_MPC	
0x041_0000	0x04F_FFFF	-	Резерв		
0x050_0000	0xFFF_FFFF	-	Резерв		-

## 2.8.5 Область External RAM

Таблица 2.9. Карта памяти K1892ВГ1Я (область External RAM)

Диапазон адресов		Размер	Назначение	Доступ	Контроллер безопасности
От	До				
0x6000_0000	0x63FF_FFFF	64MB	Внешняя память SMC	NS	SMC_MPC
0x6400_0000	0x6FFF_FFFF	-	Резерв	-	-
0x7000_0000	0x73FF_FFFF	64MB	Внешняя память SMC	S	SMC_MPC
0x7400_0000	0x7FFF_FFFF	-	Резерв	-	-
0x8000_0000	0x9FFF_FFFF	-	Резерв	-	-

Примечания:  
S – Secure  
NS – Non-secure

## 2.8.6 Область External Device

Диапазон адресов 0xA000\_0000 – 0xDFFF\_FFFF зарезервирован. Обращения по любому адресу из этого диапазона завершаются с ошибкой.

## 2.8.7 Область System

Таблица 2.10. Карта памяти K1892ВГ1Я (область System)

Диапазон адресов		Размер	Назначение
От	До		
0xE000_0000	0xE00F_FFFF	1MB	Интерфейс Private Peripheral Bus (PPB). Локальный для каждого процессора
0xE010_0000	0xEFFF_FFFF	255MB	Резерв
0xF000_0000	0xF00F_FFFF	1MB	Устройства подсистемы отладки. Доступны только через JTAG/SWD интерфейс
0xF010_0000	0xFFFF_FFFF	255MB	Резерв

## 2.8.8 Периферийные устройства подсистемы CPU

Для каждой из двух процессорных подсистем K1892BG1Я предусмотрен диапазон адресов для доступа к устройствам, индивидуальным для этой подсистемы.

Диапазон состоит из Non-secure области 0x4001\_0000 – 0x4001\_FFFF, и Secure области 0x5001\_0000 – 0x5001\_FFFF, как показано в следующей таблице.

**Таблица 2.11. Карта памяти K1892BG1Я (устройства подсистемы CPU)**

Диапазон адресов		Размер	Доступ	Назначение
От	До			
0x4001_0000	0x4001_EFFF	-	-	Резерв
0x4001_F000	0x4001_FFFF	4KB	NS	CPU_IDENTITY
0x5001_0000	0x5001_0FFF	4KB	S	ICACHE
0x5001_1000	0x5001_1FFF	4KB	S	CPU_SECCTRL
0x5001_2000	0x5001_EFFF	-	-	Резерв
0x5001_F000	0x5001_FFFF	4MB	S	CPU_IDENTITY
Примечания: S – Secure NS – Non-secure				

Данная область памяти не доступна для других master-устройств системы, но доступна для внешнего отладчика через интерфейс D-AHB.

Поддерживаются только 32-х разрядные операции записи.

Для блока CPU\_IDENTITY доступны как Secure, так и Non-secure обращения.

### 2.8.8.1 Блок CPU\_SECCTRL

Регистровый блок CPU\_SECCTRL предназначен для блокировки настроек безопасности CPU. Блок управляется тем же сигналом сброса, а также находится в том же домене питания, что и соответствующий процессор. Регистры сбрасываются при включении питания домена CPU.

**Таблица 2.12. Регистры CPU\_SECCTRL**

Условное обозначение	Описание	Исходное состояние	Смещение
SECCFG	Регистр блокировки параметров безопасности CPU	0x0000_0000	0x000
-	Резерв	0x0000_0000	0x004 – 0xFFC
PIDR4	Product ID 4.	0x0000_0004	0xFD0
PIDR5	Резерв.	0x0000_0000	0xFD4
PIDR6	Резерв.	0x0000_0000	0xFD8
PIDR7	Резерв.	0x0000_0000	0xFDC
PIDR0	Product ID 0.	0x0000_0059	0xFE0
PIDR1	Product ID 1.	0x0000_00B8	0xFE4
PIDR2	Product ID 2.	0x0000_000B	0xFE8
PIDR3	Product ID 3.	0x0000_0000	0xFEC
CIDR0	Component ID 0.	0x0000_000D	0xFF0
CIDR1	Component ID 1.	0x0000_00F0	0xFF4

CIDR2	Component ID 2.	0x0000_0005	0xFF8
CIDR3	Component ID 3.	0x0000_00B1	0xFFC

### 2.8.8.1.1 Регистр CPU\_SECCTRL\_SECCFG

Таблица 2.13. Регистр CPU\_SECCTRL\_SECCFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0	Резерв
1	LOCKSAU	WO	0	Управляет сигналом <b>LOCKSAU</b> процессора. При установке блокирует записи в регистры SAU_CTRL, SAU_RNR, SAU_RBAR и SAU_RLAR. После установки бит не может быть очищен до сброса.
0	LOCKSVTAIRCR	WO	0	Управляет сигналом <b>LOCKSVTAIRCR</b> процессора. При установке блокирует запись в регистры VTOS_S, AIRCR.PRIS, AIRCR.BFHFNMINS. После установки бит не может быть очищен до сброса.

### 2.8.8.2 Блок CPU\_IDENTITY

Таблица 2.14. Регистры CPU\_IDENTITY

Условное обозначение	Описание	Исходное состояние	Смещение
CPUID	Уникальный идентификационный номер CPU	0x0 для CPU0 0x1 для CPU1	0x000
-	Резерв	0x0000_0000	0x004 – 0xFFC
PIDR4	Product ID 4.	0x0000_0004	0xFD0
PIDR5	Резерв.	0x0000_0000	0xFD4
PIDR6	Резерв.	0x0000_0000	0xFD8
PIDR7	Резерв.	0x0000_0000	0xFDC
PIDR0	Product ID 0.	0x0000_0055	0xFE0
PIDR1	Product ID 1.	0x0000_00B8	0xFE4
PIDR2	Product ID 2.	0x0000_000B	0xFE8
PIDR3	Product ID 3.	0x0000_0000	0xFEC
CIDR0	Component ID 0.	0x0000_000D	0xFF0
CIDR1	Component ID 1.	0x0000_00F0	0xFF4
CIDR2	Component ID 2.	0x0000_0005	0xFF8
CIDR3	Component ID 3.	0x0000_00B1	0xFFC

## 2.9 Прерывания и исключения

Этот раздел описывает контроллер прерываний Nested Vectored Interrupt Controller (NVIC), карту сигналов прерываний и сигналы событий.

### 2.9.1 Контроллер прерываний NVIC

Основные возможности NVIC:

- 93 маскируемых прерывания.
- Одно немаскируемое прерывание (Non-Maskable Interrupt, NMI).

- Программируемый уровень приоритета (0-15) для каждого прерывания. Более высокий уровень соответствует меньшему приоритету.
- Детектирование прерываний по импульсу и уровню.
- Динамическое изменение приоритета прерываний.
- Объединение приоритетов в группы.
- Обработка прерываний в цепочке (tail-chaining).
- Поддержка внешнего, по отношению к NVIC, контроллера External Wake Up Controller (EWC) для обеспечения низкопотребляющих режимов работы.

Подробная информация о Cortex-M33 приведена в следующих документах:

- Arm® Cortex®-M33 Processor Technical Reference Manual.
- Arm® v8-M Architecture Reference Manual.

## 2.9.2 Прерывания

В Таблица 2.15 приведена таблица векторов прерываний для каждого процессора CUPn (n = 0,1). В таблице указаны источники прерываний, смещения адреса в таблице векторов прерываний, а также показано, какие прерывания используются как wake-up-событие для контроллера EWC каждого процессора.

**Таблица 2.15. Таблица векторов прерываний**

Номер IRQ	Secure Vector	Non-secure Vector	Описание	Смещение	Поддержка EWC
-	Initial SP		Начальное значение указателя стека	0x00	-
-	Reset		Обработчик сброса	0x04	-
-	NMI_S	NMI_NS	Немаскируемое прерывание – комбинированные прерывания SWDT и LPWDT	0x08	+
-	HardFault_S	HardFault_NS	Hard fault	0x0C	-
-	MemManage_S	MemManage_NS	Memory management	0x10	-
-	BusFault_S	BusFault_NS	Pre-fetch fault, memory access fault	0x14	-
-	UsageFault_S	UsageFault_NS	Undefined instruction or illegal state	0x18	-
-	SecureFault	-	Secure fault	0x1C	-
-	-		Резерв	0x20	-
-				0x24	-
-				0x28	-
-	SVCall_S	SVCall_NS	System service call via SWI instruction	0x2C	-
-	DebugMonitor	DebugMonitor	Debug monitor	0x30	-
-	-	-	Резерв	0x34	-
-	PendSV_S	PendSV_NS	Pendable request for system service	0x38	-
-	SysTick_S	SysTick_NS	System tick timer	0x3C	-
0	NSWDT_RST		Прерывание Non-secure Watchdog по запросу сброса	0x40	+
1	NSWDT		Прерывание Non-secure Watchdog	0x44	+

Номер IRQ	Secure Vector	Non-secure Vector	Описание	Смещение	Поддержка EWC
2		LPTIM	Прерывание LP Timer	0x48	+
3		TIM0	Прерывание Timer 0	0x4C	+
4		TIM1	Прерывание Timer 1	0x50	+
5		DTIM	Прерывание Dual Timer	0x54	+
6		MHU0	Прерывание MHU0 процессора CPU <sub>n</sub>	0x58	+
7		MHU1	Прерывание MHU1 процессора CPU <sub>n</sub>	0x5C	+
8		CRYPTO	Прерывание CryptoCell	0x60	+
9	-	MPC	Комбинированное прерывание MPC	0x64	+
10	-	PPC	Комбинированное прерывание PPC	0x68	+
11	-	MSC	Комбинированное прерывание MSC	0x6C	+
12	-	BRG	Прерывание ошибки Sync-Down моста	0x70	+
13	-	ICACHE <sub>n</sub>	Прерывание кэша инструкций процессора CPU <sub>n</sub>	0x74	-
14	-	-	Резерв	0x78	-
15	-	SYS_PPU	Прерывание SYS_PPU	0x7C	-
16	-	CPU0_PPU	Прерывание CPU0_PPU	0x80	-
17	-	CPU1_PPU	Прерывание CPU1_PPU	0x84	-
18	-	-	Резерв	0x88	-
19	-	-	Резерв	0x8C	-
20	-	CRYPTO_PPU	Прерывание CRYPTO_PPU	0x90	-
21	-	-	Резерв	0x94	-
22	-	RAM0_PPU	Прерывание RAM0_PPU	0x98	-
23	-	RAM1_PPU	Прерывание RAM1_PPU	0x9C	-
24	-	RAM2_PPU	Прерывание RAM2_PPU	0xA0	-
25	-	RAM3_PPU	Прерывание RAM3_PPU	0xA4	-
26	-	DEBUG_PPU	Прерывание DEBUG_PPU	0xA8	-
27	-	-	Резерв	0xAC	-
28		CPU <sub>n</sub> CTIIRQ0	Прерывание CPU <sub>n</sub> CTIIRQ0 процессора CPU <sub>n</sub>	0xB0	-
29		CPU <sub>n</sub> CTIIRQ1	Прерывание CPU <sub>n</sub> CTIIRQ1 процессора CPU <sub>n</sub>	0xB4	-
30	-	-	Резерв	0xB8	-
31	-	-	Резерв	0xBC	-
32		UART0	Прерывание UART0	0xC0	+
33		UART1	Прерывание UART1	0xC4	+
34		UART2	Прерывание UART2	0xC8	+
35		UART3	Прерывание UART3	0xCC	+
36		SPI0	Прерывание SPI0	0xD0	+
37		SPI1	Прерывание SPI1	0xD4	+
38		SPI2	Прерывание SPI2	0xD8	+
39		I2C0	Прерывание I2C0	0xDC	+
40		I2C1	Прерывание I2C1	0xE0	+
41		I2S	Прерывание I2S	0xE4	+
42		GPIO0_S	Secure прерывание GPIO0	0xE8	+
43		GPIO1_S	Secure прерывание GPIO1	0xEC	+
44		GPIO2_S	Secure прерывание GPIO2	0xF0	+
45		GPIO3_S	Secure прерывание GPIO3	0xF4	+
46		CAN	Прерывание CAN	0xF8	+
47		PWM	Прерывание PWM	0xFC	+
48		VTU0	Прерывание VTU0	0x100	+

Номер IRQ	Secure Vector	Non-secure Vector	Описание	Смещение	Поддержка EWC
49		VTU1	Прерывание VTU1	0x104	+
50	-	FCACHE	Прерывание Flash-кэш	0x108	-
51	-	FCTR	Прерывание Flash-контроллера	0x10C	+
52		QSPI	Прерывание QSPI	0x110	+
53		-	Резерв	0x114	-
54		GNSS	Прерывание GNSS	0x118	+
55		DMA0	Комбинированное прерывание DMA0	0x11C	+
56		DMA1	Комбинированное прерывание DMA1	0x120	+
57		SDMMC	Прерывание SDMMC	0x124	+
58		USB	Прерывание USB	0x128	+
59		JTM	Прерывание JTM	0x12C	+
60	-	GMS_URB	Прерывание GMS: блок URB	0x130	+
61	-	GMS_DMA	Прерывание GMS: блок DMA	0x134	+
62	-	GMS_PPU	Прерывание GMS PPU	0x138	-
63	-	GNSS_PPU	Прерывание GNSS PPU	0x13C	-
64		RWC_WKUP	Прерывание RWC: событие wakeup	0x140	+
65		RWC_TICK	Прерывание RWC: секундный строб	0x144	+
66		RWC_ALARM	Прерывание RWC: событие alarm	0x148	+
67		DMA0_CH0	Прерывание DMA0, канал 0	0x14C	+
68		DMA0_CH1	Прерывание DMA0, канал 1	0x150	+
69		DMA0_CH2	Прерывание DMA0, канал 2	0x154	+
70		DMA0_CH3	Прерывание DMA0, канал 3	0x158	+
71		DMA0_CH4	Прерывание DMA0, канал 4	0x15C	+
72		DMA0_CH5	Прерывание DMA0, канал 5	0x160	+
73		DMA0_CH6	Прерывание DMA0, канал 6	0x164	+
74		DMA0_CH7	Прерывание DMA0, канал 7	0x168	+
75		DMA1_CH0	Прерывание DMA1, канал 0	0x16C	+
76		DMA1_CH1	Прерывание DMA1, канал 1	0x170	+
77		DMA1_CH2	Прерывание DMA1, канал 2	0x174	+
78		DMA1_CH3	Прерывание DMA1, канал 3	0x178	+
79		DMA1_CH4	Прерывание DMA1, канал 4	0x17C	+
80		DMA1_CH5	Прерывание DMA1, канал 5	0x180	+
81		DMA1_CH6	Прерывание DMA1, канал 6	0x184	+
82		DMA1_CH7	Прерывание DMA1, канал 7	0x188	+
83		VMON	Прерывание монитора питания	0x18C	+
84		SDMMC_WKUP	Прерывание SDMMC: событие wakeup	0x190	+
85		GPIO0_NS	Non-secure прерывание GPIO0	0x194	+
86		GPIO1_NS	Non-secure прерывание GPIO1	0x198	+
87		GPIO2_NS	Non-secure прерывание GPIO2	0x19C	+
88		GPIO3_NS	Non-secure прерывание GPIO3	0x1A0	+
89		GPIO0_ACC	Прерывание по нарушению доступа GPIO0	0x1A4	+
90		GPIO1_ACC	Прерывание по нарушению доступа GPIO1	0x1A8	+
91		GPIO2_ACC	Прерывание по нарушению доступа GPIO2	0x1AC	+
92		GPIO3_ACC	Прерывание по нарушению доступа GPIO3	0x1B0	+

Каждый процессор получает прерывания только от своего локального кэша инструкций и STIRQ.

Прерывания от ICACHE и всех контроллеров PPU, а также прерывания MPC, PPC, MSC, BRG, GMS\_URB, GMS\_DMA, FCACHE, FCTR должны обрабатываться в Secure состоянии.

Сигнал прерывания недоверенного сторожевого таймера NSWDT и его запрос сброса используются в качестве прерываний процессора. Прерывание по запросу сброса должно обрабатываться как Secure прерывание с помощью Trusted Execution Environment (TEE). Таким образом, запрос сброса от NSWDT не сбрасывает систему напрямую. Для разрешения сброса системы с помощью NSWDT необходимо установить бит NSWDT\_EN в регистре SYSCTR\_RESET\_MASK.

Прерывания от доверенных сторожевых таймеров SWDT и LPWDT объединены в NMI-прерывание, которое может быть подключено к одному или обоим ядрам CPU с помощью регистра SYSCTR\_NMI\_ENABLE.

В системе предусмотрено два блока Message Handling Unit (MHU). Если используется TEE, то один блок MHU должен быть сконфигурирован как Secure, а второй – как Non-secure.

### 2.9.3 Сигналы событий

В дополнение к прерываниям, следующие сигналы событий могут быть использованы для управления исполнением инструкций:

- RXEV (входное событие) сигнал события, получаемый процессором Cortex-M33 и подключенный к выводу TXEV другого процессора в системе.
- TXEV (выходное событие) сигнал события, передаваемый процессором Cortex-M33 и подключенный к выводу RXEV другого процессора в системе.

Сигналы событий могут быть использованы совместно с инструкцией WFE.

## 2.10 Питание

### 2.10.1 Основные особенности

Организация питания K1892ВГ1Я имеет следующие основные особенности:

- Достаточно одного внешнего источника питания с номинальным напряжением 3.3 В
- Встроенные мониторы внешнего питания для формирования прерывания и Power-On сброса
- Встроенный DC-DC преобразователь для формирования напряжения питания ядра
- Два уровня напряжения питания ядра: 0.9 В, 1.1 В
- Несколько отключаемых доменов питания ядра для снижения энергопотребления
- Малопотребляющие режимы работы STANDBY и SHUTDOWN
- Батарейный/резервный домен питания и режим работы BACKUP
- Сенсоры температуры и напряжения

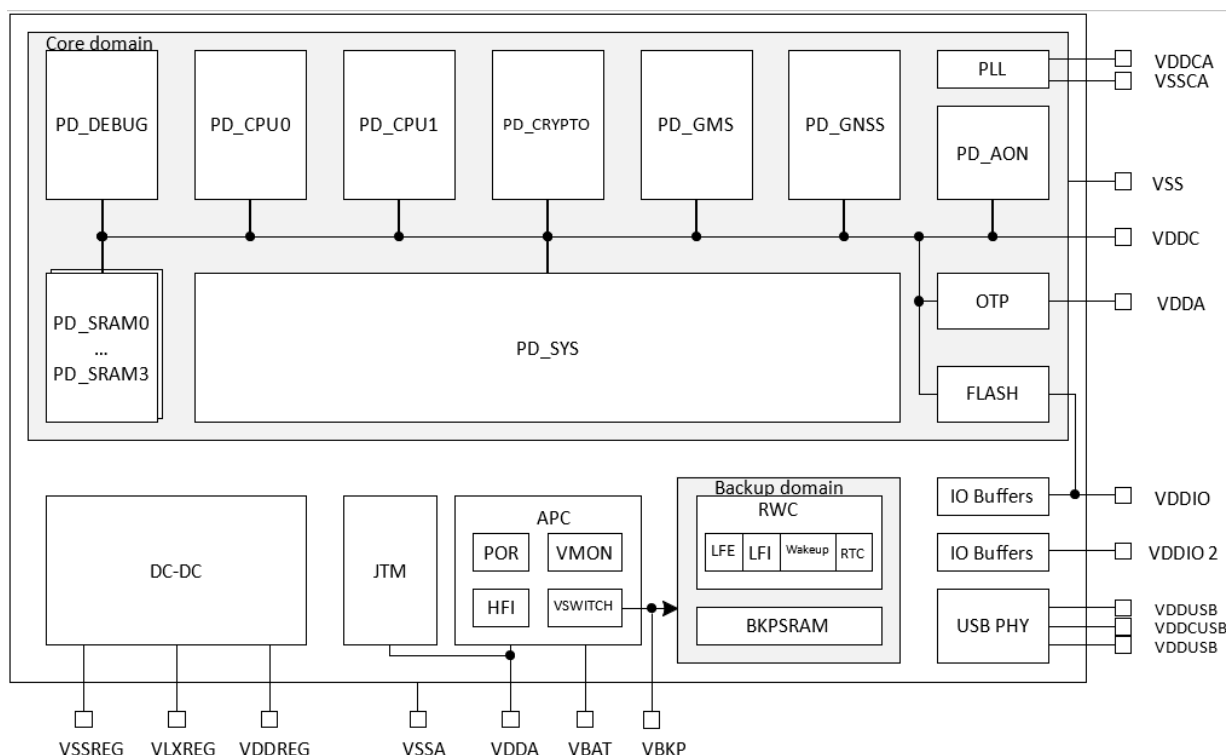
Примечание: в текущей версии микросхемы режим работы BACKUP не реализован

### 2.10.2 Обзор источников и доменов питания

Для питания микросхемы требуется минимум один источник с номинальным напряжением 3.3 В. Напряжение питания ядра (0.9 – 1.1 В) формируется с помощью внутреннего DC-DC преобразователя либо, при необходимости, с помощью внешнего источника.

На Рисунок 2.5 показаны выводы и домены питания K1892ВГ1Я.





**Рисунок 2.5. Выводы и домены питания микросхемы**

Микросхема имеет следующие выводы земли и питания:

- VDDIO: питание 1.8/3.3 В буферов ввода-вывода (кроме выводов PD6 - PD11) и Flash памяти.
- VDDIO2: питание 1.8/3.3 В буферов ввода-вывода PD6 - PD11.
- VDDA, VSSA: питание 3.3 В и земля аналоговых блоков APC, JTM; питание памяти OTP.
- VBAT: независимое резервное питание (1.6 - 3.3 В). Используется для получения напряжения питания RWC и BKPSRAM при отсутствии напряжения VDDA.
- VBKP: выходное напряжение внутреннего регулятора для питания памяти BKPSRAM. Используется для тестирования.
- VDDREG, VSSREG: входное напряжение 3.3 В и земля встроенного DC-DC регулятора.
- VLXREG: выходное напряжение встроенного DC-DC регулятора (см. раздел Регулятор DC-DC).
- VDDC: питание цифрового ядра (0.9/1.1 В).
- VDDCA: аналоговое питание PLL (0.9/1.1 В). Должно совпадать по напряжению с VDDC.
- VDDUSB: питание 3.3 В драйверов ввода вывода приемо-передатчика USB PHY. Если USB PHY не используется, то вывод VDDUSB может быть подключен к VSSUSB.
- VDDCUSB, VSSUSB: питание ядра (0.9/1.1 В) и земля приемо-передатчика USB PHY. Напряжение на выводе VDDCUSB должно совпадать с напряжением на VDDC.

Примечание: выше указано номинальное напряжение питания каждого вывода. Минимальные и максимальные допустимые значения приведены в разделе 36 «Справочная информация».

Питание микросхемы можно разделить на несколько категорий:

- Основное питание (далее обозначается как  $VDD_{MAIN}$ ). К источнику основного питания должны быть подключены выводы  $VDDA$ ,  $VDDREG$ . Опционально, могут быть подключены выводы  $VDDIO$ ,  $VDDIO2$ ,  $VDDUSB$ .
- Резервное (батарейное) питание. Подключается к выводу  $VBAT$ . При отключении основного питания используется для поддержания работы резервного (Backup) домена, куда входят блок  $RWC$  (счетчик реального времени, осцилляторы  $LFI$  и  $LFE$ , wake-up контроллер) и память  $BKPSRAM$ . Если резервное питание не требуется, то вывод  $VBAT$  необходимо подключить к основному источнику.
- Питание ядра (далее обозначается как  $VCORE$ ). Сюда входит питание блока  $PLL$  (вывод  $VDDCA$ ), питание цифрового ядра (вывод  $VDDC$ ) и питание  $USB PHY$  (вывод  $VDDCUSB$ ). Формируется с помощью внутреннего DC-DC конвертера. Также, питание ядра может формироваться с помощью внешнего регулятора, при этом встроенный DC-DC конвертер должен быть отключен (см. п. 2.10.4.3).
- Питание цифровых буферов ввода/вывода  $VDDIO$ ,  $VDDIO2$ . Могут быть подключены к независимым источникам питания 1.8/3.3 В.
- Питание встроенного трансивера  $USB PHY$ . Вывод  $VDDUSB$  может быть подключен к независимому источнику напряжения. Также указанный вывод может быть подключен к  $VSSUSB$ , если  $USB PHY$  не используется.

Примечание: в текущей версии микросхемы питание от  $VBAT$  не реализовано, вывод  $VBAT$  необходимо подключать к основному питанию.

### 2.10.3 Последовательность включения/выключения питания

При включении микросхемы питание ядра  $VCORE$  должно подаваться одновременно либо после подачи питания  $VDD_{MAIN}$ . При выключении микросхемы питание ядра  $VCORE$  должно отключаться одновременно с отключением питания  $VDD_{MAIN}$  либо перед этим. При использовании встроенного DC-DC конвертера эти условия выполняются автоматически.

Время нарастания напряжения питания  $VCORE/VDD_{MAIN}$  от 0 В до номинального значения, а также время спада от номинального значения до 0 В должно быть не менее 30 мкс.

Питание  $USB$  трансивера на вывод  $VDDUSB$  может подаваться независимо от питания  $VDD_{MAIN}$  и  $VCORE$ . Если встроенный  $USB$  трансивер не используется, то вывод  $VDDUSB$  может быть подключен к  $VSSUSB$ .

## 2.10.4 Регулятор DC-DC

### 2.10.4.1 Схема подключения

Для работы встроенного DC-DC регулятора требуются внешние катушка индуктивности  $L$  и конденсатор  $C$  (Рисунок 2.6). Типовые значения элементов:  $L = 4.7$  мкГн,  $C = 2 \times 10$  мкФ.

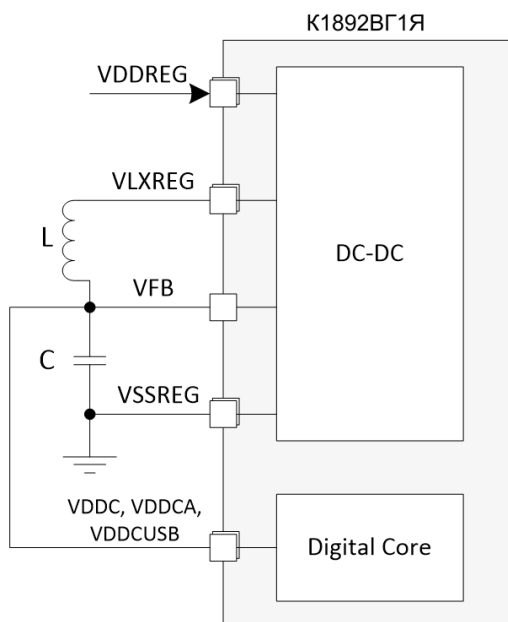


Рисунок 2.6. Схема подключения DC-DC регулятора

### 2.10.4.2 Регулировка напряжения

Напряжение питания ядра имеет три предустановленных уровня:

- VLEVEL0: 0.9 В;
- VLEVEL1: 1.0 В;
- VLEVEL2: 1.1 В.

Выбор уровня производится параметром DCDC\_VSEL регистра **PWRCTR\_RUNCFG** (в режиме RUN) и регистра **PWRCTR\_STBYCFG** (в режиме STANDBY). При первом включении питания, а также при «холодном» сбросе микросхемы используется уровень VLEVEL0.

Фактическое значение напряжения для каждого уровня задается в регистре **PWRCTR\_VLEVEL** и определяется из таблицы ниже.

Таблица 2.16. Выходное напряжение DC-DC регулятора

VLEVEL <sub>x</sub> , x = 0, 1, 2	Напряжение, В	VLEVEL <sub>x</sub> , x = 0, 1, 2	Напряжение, В
0x00	0.85	0x10	1.01
0x01	0.86	0x11	1.02
0x02	0.87	0x12	1.03
0x03	0.88	0x13	1.04

VLEVEL <sub>x</sub> , x = 0, 1, 2	Напряжение, В	VLEVEL <sub>x</sub> , x = 0, 1, 2	Напряжение, В
0x04	0.89	0x14	1.05
0x05 <sup>1</sup>	0.90	0x15	1.06
0x06	0.91	0x16	1.07
0x07	0.92	0x17	1.08
0x08	0.93	0x18	1.09
0x09	0.94	0x19 <sup>3</sup>	1.10
0x0A	0.95	0x1A	1.11
0x0B	0.96	0x1B	1.12
0x0C	0.97	0x1C	1.13
0x0D	0.98	0x1D	1.14
0x0E	0.99	0x1E	1.15
0x0F <sup>2</sup>	1.00	0x1F	1.16
Примечания: 1 – значение по сбросу для VLEVEL0 2 – значение по сбросу для VLEVEL1 3 – значение по сбросу для VLEVEL2			

Параметры VLEVEL0-2 должны выбираться так, чтобы выполнялись условия:

- VLEVEL0 < VLEVEL1 < VLEVEL2;
- V<sub>VLEVEL0</sub> < 0.99 В

#### 2.10.4.3 Использование внешнего источника

При необходимости, вместо встроенного DC-DC конвертера можно использовать внешний регулятор для формирования напряжения питания ядра.

В этом случае внешнее питание ядра подается на выводы VDDC, VDDCA, VDDCUSB, а также на вывод VFB. Выводы встроенного регулятора нужно подключить таким образом:

- VDDREG – к основному питанию;
- VLXREG – оставить неподключенным;
- VSSREG – к цепи цифровой земли.

Сразу после включения микросхемы необходимо отключить внутренний DC-DC регулятор путем записи '1' в бит DCDC\_DIS регистра **PWRCTR\_CFG**.

При использовании внешнего источника питания ядра параметр DCDC\_VSEL должен быть установлен в соответствии с уровнем напряжения. Значения параметров VLEVEL0-2 не важны.

### 2.10.5 Контроллер питания APC

Контроллер питания APC предназначен для организации питания и управления блоками DC-DC, JTM, RWC. В APC входят следующие компоненты:

- Мониторы питания POR, VMON.
- Осциллятор HFI.
- Схема формирования питания резервного домена.

- Генераторы опорного напряжения и тока для DC-DC, JTM.
- Схема управления регулятором DC-DC.

Для снижения собственного потребления блоков APC и DC-DC при низких токах потребления ядра в контроллере APC предусмотрены режимы ECO:

- Режим ECO регулятора. Включается при  $APC\_ECO = 0x1$ . В этом режиме контроллер APC отслеживает выходное напряжение регулятора на выводе VFB. Если оно падает ниже порога, задаваемого параметром  $APC\_ECOPROG$ , контроллер включает регулятор. При достижении напряжения на VFB заданного с помощью  $VLEVELx$  уровня контроллер выключает регулятор.
- Режим ECO регулятора и контроллера. Включается при  $APC\_ECO = 0x3$ . В этом режиме, дополнительно к описанному выше отключению регулятора, выключаются схемы формирования опорного напряжения и тока, а также тактовый генератор HFI.

Для дальнейшего уменьшения потребления при  $APC\_ECO = 0x3$  монитор питания регулятора может семплироваться с помощью низкочастотного тактового сигнала. Эта функция включается при  $APC\_ENLPCLK = 1$ .

В обоих случаях пороговое напряжение регулятора задается параметром  $APC\_ECOPROG$  в регистрах **PWRCTR\_RUNCFG** и **PWRCTR\_STDBYCFG** для режимов работы RUN и STANDBY соответственно.

**Таблица 2.17. Пороговое напряжение DC-DC регулятора**

APC_ECOPROG	Пороговое напряжение, В	APC_ECOPROG	Пороговое напряжение, В
0x0	0.60	0x8	0.76
0x1	0.62	0x9	0.78
0x2	0.64	0xA	0.80
0x3	0.66	0xB	0.82
0x4	0.68	0xC	0.84
0x5	0.70	0xD	0.86
0x6	0.72	0xE	0.88
0x7	0.74	0xF	0.90

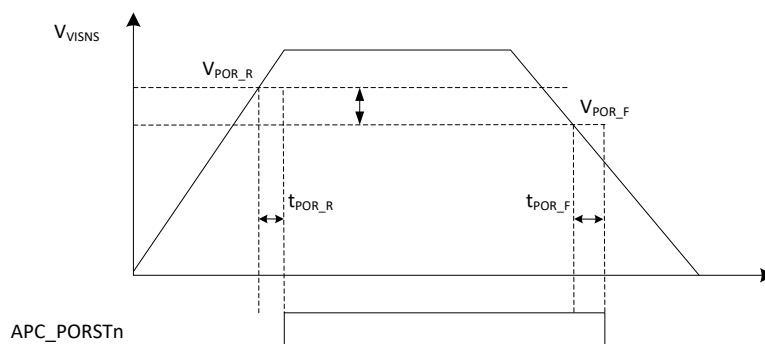
Рекомендуется устанавливать пороговое напряжение на 200 мВ меньше, чем выходное напряжение регулятора.

## 2.10.6 Мониторы основного питания

Мониторы POR и VMON предназначены для контроля входного напряжения питания с помощью вывода VISNS. Вывод VISNS должен быть подключен к цепи VDDBA.

При включении питания блок POR удерживает микросхему в состоянии сброса (Рисунок 2.9, сигнал  $APC\_PORSTn$ ), пока напряжение на VISNS не превысит порог  $V_{POR\_R}$

(примерно 2.5 В). При падении напряжения на VISNS ниже  $V_{POR\_F}$  (примерно 2.4 В) генерируется сброс APC\_PORSTn (Рисунок 2.7).



**Рисунок 2.7. Временная диаграмма работы монитора POR**

Монитор VMON может генерировать прерывание при падении напряжения на VISNS ниже  $V_{IRQ\_F}$  (примерно 2.8 В) либо при превышении порога  $V_{IRQ\_R}$  (примерно 2.9 В). Включение прерывания VMON выполняется с помощью параметров VMON\_FTI\_EN, VMON\_RTI\_EN регистра **PWRCTR\_CFG**. Сброс прерывания производится записью '1' в поле VMON\_INT\_CLR регистра **PWRCTR\_CFG**.

Подробнее о характеристиках мониторов см. в разделе справочной информации данного руководства.

## 2.10.7 Измерение температуры и напряжения, JTM

Блок JTM представляет из себя сигма-дельта АЦП и сенсор для измерения температуры. JTM имеет несколько измерительных каналов, подключенных следующим образом:

- канал 1 – температурный сенсор;
- каналы 2, 3, 4 не используются;
- канал 5 – внутреннее напряжение питания;
- каналы 6, 7 не используются;
- канал 8 – напряжение питания ядра с вывода VDDC.

Подробнее о JTM см. в соответствующем разделе данного руководства.

## 2.10.8 Батарейный домен

Контроллер APC содержит переключатель напряжения и внутренние LDO регуляторы для получения напряжения питания резервного домена, в который входят:

- Блок RWC, содержащий осцилляторы LFI, LFE; часы реального времени RTC; контроллер внешнего прерывания WKUP.
- Память BKPSRAM.

Вывод VDDA используется для питания батарейного домена, если напряжение на нем превышает 2.5 В (примерно). При уменьшении напряжения ниже 2.5 В происходит переключение на питание от вывода VBAT. Текущее напряжение питания памяти BKPSRAM доступно на выводе VBKP для мониторинга.

Примечание: в текущей версии микросхемы питание от VBAT не реализовано

## 2.10.9 Питание Flash памяти

Flash память имеет два режима пониженного потребления: Sleep Mode (SLM) и Power Down Mode (PDM). Управление режимами производится с помощью параметра FLASH\_PWR\_CTRL в регистрах **PWRCTR\_RUNCFG** (для режима работы микросхемы RUN) и **PWRCTR\_STDBYCFG** (для режима работы микросхемы STANDBY).

При включении режима микросхемы SHUTDOWN (питание ядра отключено) Flash память автоматически переходит в режим PDM.

Для перехода памяти из SLM или PDM в рабочий режим требуется время 5 мкс. В течение этого времени память не доступна. Текущее состояние памяти определяется полем FLASH\_PWR\_STAT регистра **PWRCTR\_STAT**.

Также, Flash память имеет режим чтения LVE при пониженном напряжении питания ядра (от 0.81 В до 0.99 В). Этот режим автоматически включается при уровне напряжения ядра DCDC\_VSEL=0. При другом уровне напряжения ядра режим LVE автоматически отключается.

Для управления режимом LVE предназначен параметр FLASH\_LVE регистра **PWRCTR\_CFG**. Текущее значение LVE определяется полем FLASH\_LVE\_STAT регистра **PWRCTR\_STAT**.

## 2.10.10 Домены питания ядра

### 2.10.10.1 Введение

Цифровое ядро микросхемы разделено на несколько доменов питания (Рисунок 2.5):

- PD\_CPU0, PD\_CPU1 – домены питания подсистем CPU0 и CPU1 соответственно. В них входят отключаемая часть процессора Cortex-M33 и кэш инструкций ICACHE.
- PD\_DEBUG – домен питания подсистемы отладки.
- PD\_CRYPT0 – домен питания крипто-ускорителя CryptoCell.
- PD\_GMS – домен питания крипто-ускорителя GMS.
- PD\_GNSS – домен питания навигационного ядра GNSS.
- PD\_SRAMn – домены питания банков памяти SRAMn, n = 0, 1, 2, 3.
- PD\_SYS – системный домен питания, включающий основную матрицу коммутации и большую часть периферийных блоков.

- PD\_AON – всегда включенный (при наличии VDDC) домен питания. Содержит контроллеры питания (PPU), контроллеры прерываний WIC и EWC, таймеры LPTIM и LPWDT, блоки регистров CLKCR, PWRCTR, PORTCR, SYSCTR и ряд других блоков.

Каждый домен питания ядра, кроме PD\_AON, управляется с помощью собственного контроллера Power Policy Unit (PPU) и может находиться в одном из следующих состояний:

- ON – питание включено;
- OFF – питание логики и блоков памяти выключено, состояние схемы теряется;
- MEM\_RET – режим пониженного энергопотребления памяти (retention) с сохранением данных (только для доменов PD\_SRAMn).

Домен PD\_AON всегда находится в состоянии ON при наличии внешнего питания VDDC.

Примечание: в текущей версии микросхемы снятие питания с домена не реализовано, при переводе PPU в режим OFF с домена снимается тактирование и формируется сигнал сброса.

#### 2.10.10.2 Управление с помощью PPU

PPU может управлять состоянием домена двумя способами:

- Статическое управление – домен может быть перманентно включен, выключен или переведен в режим сохранения данных (только для PD\_SRAMn).
- Динамическое управление – целевое состояние питания домена задается программно. Переход из состояния ON в заданное состояние и обратно происходит автоматически при выполнении определенных условий (см. далее). Возможны следующие динамические переходы:
  - ON → OFF
  - OFF → ON
  - ON → MEM\_RET
  - MEM\_RET → ON

Домены PD\_CRYPTO, PD\_GMS, PD\_GNSS поддерживают только статическое управление.

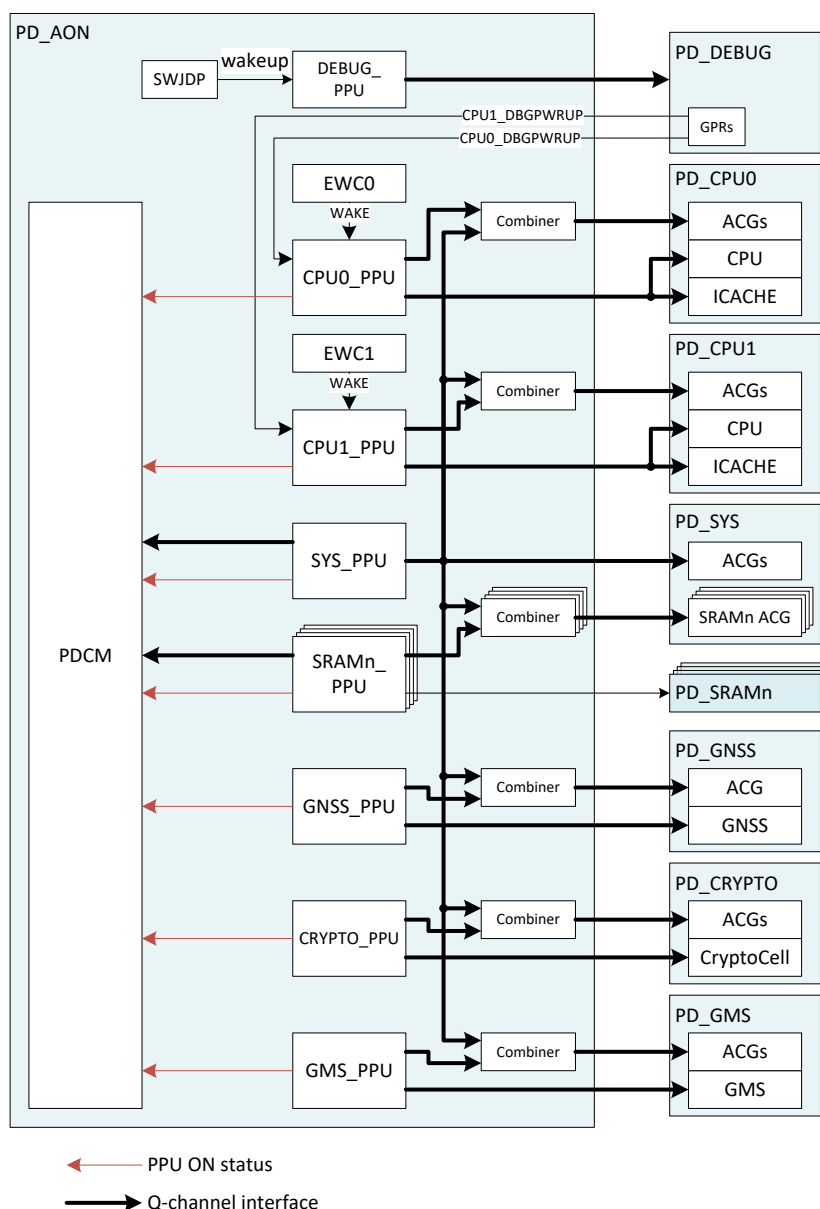
Установка режима управления и состояния домена выполняется с помощью регистра PPU\_PWPR. Подробнее см. в разделе «PPU: Контроллер управления питанием».

#### 2.10.10.3 Переход в состояние OFF/MEM\_RET, PDCM

Для корректного переключения домена из состояния ON в состояние OFF или MEM\_RET контроллеры PPU учитывают готовность отключаемых устройств к переходу в малопотребляющее состояние. Для этого каждый блок PPU подключен по интерфейсу Q-channel к контролируемым им устройствам. Переход в OFF/MEM\_RET выполняется, только если PPU получит подтверждение от каждого устройства.

На рисунке Рисунок 2.8 показана упрощенная схема взаимодействия PPU с устройствами в отключаемых доменах, а также с PDCM.





### Рисунок 2.8. Схема взаимодействия RPU

Блоки Access Control Gate (ACG) отслеживают активность на интерфейсах АНВ устройств в системе (см. Рисунок 2.3). ACG дает подтверждение на запрос выключения домена при условии, что на соответствующем интерфейсе АНВ нет активных транзакций в течение некоторого времени.

Блоки Combiner необходимы для объединения запросов на закрытие ACG, поступающих от SYS PPU, с аналогичными запросами от других PPU.

Дополнительно, возможность выключения доменов PD\_SYS и PD\_SRAMn зависит от состояний других доменов. Это достигается с помощью блока Power Dependency Control Matrix (PDCM). В следующей таблице показана матрица взаимосвязей контроллеров PPU и соответствующих доменов, реализуемая в PDCM.

**Таблица 2.18. Матрица взаимосвязей PPU**

Домен	Контролируемый домен				
	PD_SYS	PD_SRAM0	PD_SRAM1	PD_SRAM2	PD_SRAM3
PD_SYS	+/-	+/-	+/-	+/-	+/-
PD_CPU0	+	+/-	+/-	+/-	+/-
PD_CPU1	+	+/-	+/-	+/-	+/-
PD_SRAM0	+	+/-	-	-	-
PD_SRAM1	+	-	+/-	-	-
PD_SRAM2	+	-	-	+/-	-
PD_SRAM3	+	-	-	-	+/-
PD_CRYPT0	+	-	-	-	-
PD_GMS	+	-	-	-	-
PD_GNSS	+	-	-	-	-

В левом столбце таблицы приведены домены, от состояния которых зависит состояние контролируемого домена (PD\_SYS, PD\_SRAMn). Знаком «+» обозначено наличие фиксированной зависимости, знаком «+/-» – конфигурируемая зависимость. Знак «-» обозначает отсутствие зависимости.

Контролируемый домен остается включенным до тех пор, пока хотя бы один из доменов-зависимостей включен:

- Домен PD\_SYS остается в состоянии ON, пока включен любой другой домен, кроме PD\_DEBUG.
- Если включена зависимость PD\_SYS->PD\_SYS, то после включения домен PD\_SYS остается в состоянии ON.
- Домены PD\_SRAMn могут быть сконфигурированы так, чтобы оставаться включенными, пока включены домены PD\_CPU0, PD\_CPU1, PD\_SYS.
- Если включена зависимость PD\_SRAMn->PD\_SRAMn, то после включения домен PD\_SRAMn остается в состоянии ON.

Таким образом, блок PDCM определяет условие перехода контролируемого домена в малопотребляющее состояние. Однако, PDCM не предназначен для включения доменов, находящихся в состоянии OFF или MEM\_RET.

Взаимосвязи состояний доменов настраиваются с помощью регистров SYSCTR\_PDCM\_SYS\_SENSE, SYSCTR\_PDCM\_SRAMn\_SENSE.

#### 2.10.10.4 Переход в состояние ON

Домены, поддерживающие динамическое управление, могут быть включены по сигналу от определенных устройств в системе.

#### EWC и GPR

Каждый процессор Cortex-M33 в K1892BГ1Я содержит контроллер WIC (Wakeup Interrupt Controller) со следующими функциями:

- позволяет переходить процессору в режим DEEPSLEEP с отключением тактирования ядра и контроллера прерываний NVIC;

- сохраняет маску прерываний из NVIC для обработки последующих прерываний;
- если маска прерываний позволяет, подает запрос на пробуждение процессора из DEPPSLEEP.

Однако WIC не поддерживает пробуждение процессора, если питание ядра выключено. Для этого предназначен блок EWC (External Wakeup Controller). EWC использует маску прерываний, сохраненную в WIC, для удержания прерываний, происходящих при выключенном питании CPU, а также для включения питания и пробуждения CPU.

Прерывания, поддерживаемые EWC, отмечены в разделе 2.9.2.

Если планируется использовать прерывания для включения питания домена PD\_CPU<sub>n</sub>, то перед переходом домена в состояние OFF необходимо включить соответствующие контроллеры WIC и EWC (обязательно в таком порядке), используя регистры **SYSCTR\_WICCTRL** и **SYSCTR\_EWCCTRL** соответственно. При этом, после записи бита \*EN\_SET необходимо дождаться обновления бита \*EN\_STAT. После пробуждения и загрузки обработчика прерываний процессор должен сбросить EWC с помощью бита EWCnEN\_CLR.

Если питание процессора было отключено без настройки соответствующего EWC, то процессор не может быть включен с помощью прерываний. В этом случае питание домена PD\_CPU<sub>n</sub> может быть включено одним из способов:

- установив состояние питания ON в контроллере CPU<sub>n</sub>\_PPU;
- с помощью внешнего отладчика, используя соответствующий блок GPR.

EWC не использует сигналы событий и не поддерживает пробуждение процессора по событию.

Пробуждение процессора с помощью EWC или GPR также приводит к включению системного домена (PD\_SYS).

## ACG

Включение домена PD\_SRAM<sub>n</sub> происходит с помощью блока ACG при обращении к соответствующему банку памяти SRAM<sub>n</sub>.

## SWJDP

Блок SWJDP используется для включения домена PD\_DEBUG с помощью внешнего отладчика.

## 2.10.10.5 Обзор характеристик доменов

В следующей таблице приведены состав и характеристики всех доменов ядра.

**Таблица 2.19. Сводная таблица характеристик доменов ядра**

Домен	Состав	Поддерживаемые состояния	Исходное состояние	Динамическое включение
PD_SYS	MPC, MSC, PPC, TIM*, DTIM, MHU*, NSWD, SWDT, SPCTR, NSPCTR, DMA*, USB, SDMMC, SMC, QSPI, FCACHE, FCTR, CAN, VTU*, PWM, UART*, SPI*, I2C*, I2S, GPIO*	ON, OFF	ON	CPU0 GPR, CPU1 GPR, CPU0 EWC, CPU1 EWC
PD_CPU0	Подсистема CPU0: ядро CPU, MPU, NVIC, ICACHE	ON, OFF	ON	CPU0 GPR, CPU0 EWC
PD_CPU1	Подсистема CPU1: ядро CPU, MPU, NVIC, FPU, ICACHE	ON, OFF	OFF	CPU1 GPR, CPU1 EWC
PD_DEBUG	ROM-таблицы, GPR, CTI, TPIU	ON, OFF	ON	SWJDP
PD_SRAM0	SRAM0	ON, OFF, MEM_RET	ON	ACG
PD_SRAM1	SRAM1	ON, OFF, MEM_RET	ON	ACG
PD_SRAM2	SRAM2	ON, OFF, MEM_RET	ON	ACG
PD_SRAM3	SRAM3	ON, OFF, MEM_RET	ON	ACG
PD_CRYPT O	CRYPTO	ON, OFF	ON	-
PD_GMS	GMS	ON, OFF	OFF	-
PD_GNSS	GNSS	ON, OFF	OFF	-
PD_AON	Все PPU, PDCM, EWC, WIC, SWJDP, SYSINFO, SYSCTR, LPTIM, LPWDT, IOCTR, CLKCTR, PWRCTR, OTP	ON	ON	-

## 2.10.11 Режимы работы микросхемы

### 2.10.11.1 Введение

Режим работы микросхемы определяется:

- наличием основного питания VDD<sub>MAIN</sub>;
- состоянием встроенного DC-DC регулятора;
- состояниями отдельных доменов питания ядра PD\_\*.

После «холодного» сброса микросхема находится в режиме RUN, который является основным рабочим режимом микросхемы. Дополнительно предусмотрено два малопотребляющих режима – STANDBY и SHUTDOWN. Выбор режима определяется исходя из целей по потреблению и времени для возвращения в режим RUN.

При выключении основного питания  $VDD_{MAIN}$  микросхема переключается на резервное питание от вывода VBAT, что соответствует режиму BACKUP. В этом режиме функционирует только резервный домен.

В следующей таблице перечислены режимы работы микросхемы и соответствующие состояния источника питания, регулятора и доменов ядра.

**Таблица 2.20. Режимы работы микросхемы**

Режим	$VDD_{MAIN}$	DC-DC	PD_AON	PD_SYS	PD_CPU <sub>n</sub>	PD_DEBUG	PD_SRAM <sub>n</sub>	PD_CRYPTO	PD_GMS	PD_GNSS
RUN	ON	ON	ON	ON	OFF/ ON	OFF/ ON	OFF/ ON/ MEM_RET	OFF/ ON	OFF/ ON	OFF/ ON
STANDBY	ON	ON	ON	OFF	OFF	OFF/ ON	OFF/ MEM_RET	OFF	OFF	OFF
SHUTDOWN	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
BACKUP	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

В следующей таблице перечислены устройства, доступные в каждом из режимов, а также сигналы и устройства, которые могут быть использованы для возвращения в режим RUN.

**Таблица 2.21. Обзор доступных устройств в различных режимах работы**

Режим	CPU <sub>n</sub>	Flash	SRAM <sub>n</sub>	Источники тактирования	Периферийные устройства	Переход в RUN
RUN	ON / ON-SLEEP / ON-DEEPSLEEP (WIC) / OFF-DEEPSLEEP (WIC+EWC) / OFF	ON / SLM / PDM	OFF / ON / MEM_RET	LFI, LFE, HFI, XTI, PLL	Все	-
STANDBY	OFF	ON / SLM / PDM	OFF / MEM_RET	LFI, LFE, HFI, XTI, PLL	LPTIM, LPWDT, RTC, BKPSRAM	Выход SRST <sub>n</sub> ; GPR, RTC, LPTIM, LPWDT, VMON
SHUTDOWN	OFF	PDM	OFF	LFE, LFI	RTC, BKPSRAM	Выход WKUP; RTC
BACKUP	OFF	OFF	OFF	LFE, LFI	RTC, BKPSRAM	-

Примечание: в текущей версии микросхемы режим работы BACKUP не реализован

### 2.10.11.2 Режим RUN

Микросхема находится в режиме RUN при следующих условиях: включено основное питание VDD<sub>MAIN</sub> и питание ядра, домен PD\_SYS находится в состоянии ON. Состояние других доменов может быть произвольным. Поэтому режим RUN имеет большой диапазон по производительности и потребляемой мощности.

Методы снижения потребления в режиме RUN:

- снижение тактовой частоты системы;
- динамическое управление тактированием (см. раздел “Методы снижения энергопотребления”);
- снижение напряжения питания ядра;
- отключение неиспользуемых доменов;
- динамическое управление питанием доменов памяти и процессоров;
- использование режима ECO регулятора DC-DC и контроллера APC;
- использование режимов SLM, PDM Flash памяти.

### 2.10.11.3 Режим STANDBY

Режим характеризуется наличием основного питания и питания ядра, при этом домен PD\_SYS находится в состоянии OFF.

Для перехода в режим STANDBY нужно:

- установить LP\_MODE = 0 в регистре **PWRCTR\_CFG**;
- настроить параметры DC-DC, APC и Flash памяти в регистре **PWRCTR\_STDBYCFG**;
- при необходимости, выбрать HFI или XTI в качестве источника тактирования и выключить PLL;
- если планируется запуск системы с помощью запроса EWC, то как минимум один процессор и соответствующие контроллеры WIC, EWC должны быть настроены для перехода в состояние OFF-DEEPSLEEP (WIC+EWC);
- выключить статические домены PD\_GMS, PD\_GNSS, PD\_CRYPT0;
- включить динамический переход доменов памяти PD\_SRAMn в состояние OFF или MEM\_RET;
- включить динамический переход доменов PD\_DEBUG, PD\_CPU<sub>n</sub>, PD\_SYS в OFF;
- выполнить инструкцию WFI.

После перехода в STANDBY применяются настройки питания из регистра **PWRCTR\_STDBYCFG**.

В режиме STANDBY продолжают функционировать таймеры LPTIM, LPWDT; часы реального времени RTC и резервная память BKPSRAM. Домен PD\_DEBUG может оставаться включенным, если есть обращения от внешнего отладчика в отладочную подсистему.

Настройки многофункциональных портов ввода-вывода сохраняются. Буферы всех портов кроме PA[10:15] выключаются (состояние High-Z). Буферы портов PA[10:15] также выключаются, если настроены не на функцию AF0.

Выход в режим RUN возможен:

- по прерыванию от RTC, LPTIM, LPWDT, VMON;
- по системному сбросу SRSTn;
- по запросу внешнего отладчика с помощью GPR.

При переходе в RUN применяются настройки из регистра **PWRCTR\_RUNCFG**. Настройки источников тактирования такие же, как и при входе режим STANDBY.

#### 2.10.11.4 Режим SHUTDOWN

В этом режиме основное питание включено, а регулятор DC-DC и, соответственно, питание ядра выключены.

Для перехода в режим SHUTDOWN нужно:

- установить LP\_MODE = 1 в регистре **PWRCTR\_CFG**;
- выключить статические домены PD\_GMS, PD\_GNSS, PD\_CRYPT0;
- включить динамический переход доменов PD\_SRAMn, PD\_DEBUG, PD\_CPUUn, PD\_SYS в OFF;
- выполнить инструкцию WFI.

Переход в режим будет выполнен после выключения домена PD\_SYS.

В режиме SHUTDOWN продолжают функционировать часы реального времени RTC и резервная память BKPSRAM.

Буферы всех портов PA, PB, PC, PD выключаются (состояние High-Z).

Выход в режим RUN возможен:

- по прерыванию от RTC;
- по внешнему сигналу WKUP.

При переходе в RUN выполняется «холодный» сброс всей системы.

#### 2.10.11.5 Режим BACKUP

Микросхема находится в режиме BACKUP при работе только от резервного источника питания (вывод VBAT). В этом режиме функционирует RTC и резервная память.

Выход в RUN возможен только при включении основного питания.

Примечание: в текущей версии микросхемы режим работы BACKUP не реализован

## 2.10.12 Регистры управления питанием

Управление питанием (кроме PPU) производится с помощью регистрового блока PWRCTR. Поддерживаются только 32-х разрядные операции записи и чтения регистров.

### 2.10.12.1 Перечень регистров PWRCTR

Таблица 2.22. Регистры PWRCTR

Условное обозначение	Описание	Исходное состояние	Смещение
PWRCTR_CFG	Регистр общих настроек и управления	0x0000_0000	0x000
PWRCTR_STAT	Регистр статуса питания	0x0000_0000	0x004
PWRCTR_RUNCFG	Регистр конфигурации питания в основном (RUN) режиме работы	0x0000_0100	0x008
PWRCTR_STDBYCFG	Регистр конфигурации питания в режиме STANDBY	0x0000_0100	0x00C
PWRCTR_VLEVEL	Регистр настройки выходного напряжения DC-DC регулятора	0x0019_0F05	0x010
PWRCTR_TRIM	Регистр подстройки параметров PMU	0x0000_0000	0x014
PWRCTR_TEST	Регистр для доступа к тестовым режимам PMU	0x0000_0000	0x018
PWRCTR_SRAM_TRIM 0	Регистр подстройки памяти SRAM при DCDC_VSEL = 0x0	0x0000_00C9	0x020
PWRCTR_SRAM_TRIM 1	Регистр подстройки памяти SRAM при DCDC_VSEL = 0x1	0x0000_00A5	0x024
PWRCTR_SRAM_TRIM 2	Регистр подстройки памяти SRAM при DCDC_VSEL = 0x2	0x0000_0087	0x028

### 2.10.12.2 Регистр PWRCTR\_CFG

Таблица 2.23. Регистр PWRCTR\_CFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:13	-	RO	0	Резерв
12	FLASH_LVE	RW	1	Управление входом LVE (Low Voltage read Enable) Flash памяти при напряжении питания ядра $0.81\text{ В} \leq VDDC < 0.99\text{ В}$ (DCDC_VSEL = 0). При DCDC_VSEL $\neq 0$ вход LVE Flash памяти аппаратно сбрасывается в 0 независимо от значения поля FLASH_LVE. Значение на входе LVE изменяется только после завершения активной операции чтения из Flash памяти. Текущее состояние входа LVE отображается в регистре PWRCTR_STAT
11	-	RO	0	Резерв
10	VMON_INT_CLR	WO	0	Запись в это поле '1' сбрасывает прерывание от монитора питания VMON
9	VMON_FTI_EN	RW	0	Управление прерыванием по спадающему фронту выходного сигнала монитора питания VMON: 0x0: прерывание выключено; 0x1: прерывание включено



Номер бита	Название	Тип доступа	Исходное состояние	Назначение
8	VMON_RTI_EN	RW	0	Управление прерыванием по нарастающему фронту выходного сигнала монитора питания VMON: 0x0: прерывание выключено; 0x1: прерывание включено
7:5	-	RO	0	Резерв
4	DCDC_DIS	RW	0	Выключение встроенного DC-DC регулятора (см. п. 2.10.4.3)
3:1	-	RO	0	Резерв
0	LP_MODE	RW	0	Выбор режима пониженного энергопотребления при выключении домена PD_SYS: 0x0: STANDBY; 0x1: SHUTDOWN.

### 2.10.12.3 Регистр PWRCTR\_STAT

Таблица 2.24. Регистр PWRCTR\_STAT

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:23	-	RO	0	Резерв
22	FLASH_LVE_STAT	RO	0	Текущее состояние входа LVE Flash памяти
21:20	FLASH_PWR_STAT	RO	0	Текущий режим работы Flash памяти: 0x0: рабочий режим; 0x1: Sleep; 0x2: Power Down; 0x3: резерв
19:5	-	RO	0	Резерв
4	DCDC_PG	RO	0	Индикатор готовности выходного напряжения DC-DC. Устанавливается в 1 при напряжении на выводе VFB более 95% от запрограммированного уровня VLEVELx
3	-	RO	0	Резерв
2	VMON_FTI_STAT	RO	0	Статус прерывания по спадающему фронту выходного сигнала монитора питания VMON
1	VMON_RTI_STAT	RO	0	Статус прерывания по нарастающему фронту выходного сигнала монитора питания VMON
0	VMON_STAT	RO	0	Статус монитора питания VDDA: 0x0: напряжение VDDA выше порогового (примерно 2.9 В); 0x1: напряжение VDDA ниже порогового (примерно 2.8 В).

### 2.10.12.4 Регистр PWRCTR\_RUNCFG

Регистр содержит настройки питания, применяемые в режиме RUN.

Таблица 2.25. Регистр PWRCTR\_RUNCFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:22	-	RO	0	Резерв
21:20	FLASH_PWR_CTR	RW	0	Управление режимом работы Flash памяти: 0x0: рабочий режим; 0x1: Sleep; 0x2: Power Down; 0x3: резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
19:16	APC_ECOPROG	RW	0	Параметр задает порог для монитора выходного напряжения DC-DC регулятора. Используется только в ECO режиме (APC_ECO=0x1 или APC_ECO=0x3). APC включает регулятор каждый раз, когда выходное напряжение регулятора на выводе VFB падает ниже заданного порога (см. п. 2.10.5).
15:14	-	RO	0	Резерв
13	APC_ENLPCLK	RW	0	Значение 1 включает семплирование монитора питания регулятора низкочастотным тактовым сигналом для дополнительного снижения потребления.
12:11	APC_ECO	RW	0	Управление режимом ECO регулятора DC-DC и блока APC: 0x0: режим ECO выключен; 0x1: включен режим ECO регулятора DC-DC; 0x2: резерв; 0x3: включен режим ECO регулятора DC-DC и контроллера APC.
10	DCDC_ENSTDBY	RW	0	При записи 1 включает низкопотребляющий режим конвертера. Улучшает эффективность преобразования при низком токе нагрузки. При этом повышается пульсация выходного напряжения
9	DCDC_ENCCMTRAN	RW	0	При включении (значение 0b1) генерируется 16 ССМ-импульсов каждый раз при снижении тока нагрузки. Это позволяет минимизировать всплески выходного напряжения. Однако при этом снижается эффективность конвертера
8	DCDC_DISSINK	RW	1	Управление резистором подтяжки выхода DC-DC регулятора к земле: 0b0: включена подтяжка VLXREG к земле через резистор 100 Ом; 0b1: подтяжка выключена При нормальной работе DCDC DISSINK = 1
7:4	DCDC_SWDRV	RW	0	Power driver strength control
3:2	DCDC_MODE	RW	0	Режим работы встроенного DC-DC регулятора: 0b00: авто PWM/PFM; 0b01: PWM; 0b10: PWM FCCM; 0b11: PFM (рекомендуется для режима ECO)
1:0	DCDC_VSEL	RW	0	Выбор выходного напряжения встроенного DC-DC регулятора: 0x0: 0.9 В (VLEVEL0); 0x1: 1.0 В (VLEVEL1); 0x2: 1.1 В (VLEVEL2); 0x3: резерв Точное значение напряжения в каждом случае устанавливается с помощью регистра PWRCTR_VLEVEL

### 2.10.12.5 Регистр PWRCTR\_STDBYCFG

Регистр содержит настройки питания, применяемые в режиме STANDBY.

Таблица 2.26. Регистр PWRCTR\_STDBYCFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:22	-	RO	0	Резерв
21:20	FLASH_PWR_CTR	RW	0	Управление режимом работы Flash памяти: 0x0: рабочий режим; 0x1: Sleep; 0x2: Power Down; 0x3: резерв
19:16	APC_ECOPROG	RW	0x0	Параметр задает порог для монитора выходного напряжения DC-DC регулятора. Используется только в ECO режиме (APC_ECO=0x1 или APC_ECO=0x3). APC включает регулятор каждый раз, когда выходное напряжение регулятора на выводе VFB падает ниже заданного порога (см. п. 2.10.5).
15:14	-	RO	0	Резерв
13	APC_ENLPCLK	RW	0	Значение 1 включает семплирование монитора питания регулятора низкочастотным тактовым сигналом для дополнительного снижения потребления.
12:11	APC_ECO	RW	0x0	Управление режимом ECO регулятора DC-DC и блока APC: 0x0: режим ECO выключен; 0x1: включен режим ECO регулятора DC-DC; 0x2: резерв; 0x3: включен режим ECO регулятора DC-DC и контроллера APC.
10	DCDC_ENSTDBY	RW	0	При записи 1 включает низкопотребляющий режим конвертера. Улучшает эффективность преобразования при низком токе нагрузки. При этом повышается пульсация выходного напряжения
9	DCDC_ENCCMTRAN	RW	0	При включении (значение 0b1) генерируется 16 ССМ-импульсов каждый раз при снижении тока нагрузки. Это позволяет минимизировать всплески выходного напряжения. Однако при этом снижается эффективность конвертера
8	DCDC DISSINK	RW	1	Управление резистором подтяжки выхода DC-DC регулятора к земле: 0b0: включена подтяжка VLXREG к земле через резистор 100 Ом; 0b1: подтяжка выключена При нормальной работе DCDC DISSINK = 1
7:4	DCDC_SWDRV	RW	0x0	Power driver strength control
3:2	DCDC_MODE	RW	0b00	Режим работы встроенного DC-DC регулятора: 0b00: авто PWM/PFM; 0b01: PWM; 0b10: PWM FCCM; 0b11: PFM (рекомендуется для режима ECO)

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
1:0	DCDC_VSEL	RW	0	Выбор выходного напряжения встроенного DC-DC регулятора: 0x0: 0.9 В (VLEVEL0); 0x1: 1.0 В (VLEVEL1); 0x2: 1.1 В (VLEVEL2); 0x3: резерв Точное значение напряжения в каждом случае устанавливается с помощью регистра PWRCTR_VLEVEL

## 2.10.12.6 Регистр PWRCTR\_VLEVEL

Таблица 2.27. Регистр PWRCTR\_VLEVEL

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	VLEVEL2	RW	0x19	Выходное напряжение встроенного DC-DC регулятора при DCDC_VSEL = 0x2 (Таблица 2.16)
15:13	-	RO	0	Резерв
12:8	VLEVEL1	RW	0x0F	Выходное напряжение встроенного DC-DC регулятора при DCDC_VSEL = 0x1 (Таблица 2.16)
7:5	-	RO	0	Резерв
4:0	VLEVEL0	RW	0x05	Выходное напряжение встроенного DC-DC регулятора при DCDC_VSEL = 0x0 (Таблица 2.16)

## 2.10.12.7 Регистр PWRCTR\_TRIM

Регистр предназначен для подстройки параметров блоков APC и DC-DC. Параметры подстройки подбираются в процессе тестирования и не требуют коррекции со стороны пользователя микросхемы.

Таблица 2.28. Регистр PWRCTR\_TRIM

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:24	DCDC_TRIMLC	RW	0x0	Опции параметров контроля для оптимизации под разные конфигурации выходного фильтра.
23	-	RO	0	Резерв
22:20	DCDC_IMIN	RW	0x0	Опция синхронной подстройки выпрямления По умолчанию "000" = без выпрямления
19	-	RO	0	Резерв
18:16	DCDC_IMAX	RW	0x0	Опция подстройки пикового тока катушки. По умолчанию "000" = без выпрямления
15:14	-	RO	0	Резерв
13	APC_FORCETRIM	RW	0	Включает подстройку источника опорного напряжения сверхнизкого энергопотребления. Данное действие необходимо выполнять только при нормальной работе блока APC. "forcetrim" необходимо удерживать в значении 1 не менее 1 мкс, чтобы обеспечить запуск последовательности подстройки. В ходе процесса подстройки APC будет игнорировать вывод есо.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
12:8	APC_VREF_VT	RW	0x0	Подстройка источника опорного напряжения. После сброса значение регистра автоматически обновляется из памяти OTP.
7	-	RO	0	Резерв
6:4	APC_VREF_TT	RW	0x0	Подстройка источника опорного напряжения. После сброса значение регистра автоматически обновляется из памяти OTP.
3	-	RO	0	Резерв
2:0	APC_VREF_IT	RW	0x0	Подстройка источника опорного напряжения. После сброса значение регистра автоматически обновляется из памяти OTP.

### 2.10.12.8 Регистр PWRCTR\_TEST

Регистр предназначен для включения тестовых режимов блоков APC, DC-DC, JTM и RWC.

Таблица 2.29. Регистр PWRCTR\_TEST

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:29	-	RO	0	Резерв
28:24	JTM_TEST	RW	0	Выбор режима тестирования JTM
23:21	-	RO	0	Резерв
20:16	RWC_TEST	RW	0	Выбор режима тестирования RWC
15:13	-	RO	0	Резерв
12:8	DCDC_TEST	RW	0	Выбор режима тестирования DC-DC
7:4	-	RO	0	Резерв
3:0	APC_TEST	RW	0	Выбор режима тестирования APC

Тестирование выполняется с помощью цифрового вывода PMU\_DTB (альтернативная функция вывода PA9) и аналогового вывода ANATEST.

### 2.10.12.9 Регистр PWRCTR\_SRAM\_TRIM0

Таблица 2.30. Регистр PWRCTR\_SRAM\_TRIM0

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0x0	Резерв
10:8	WPULSE	RW	0x0	Настройка WPULSE
7:5	WA	RW	0x6	Настройка WA
4:3	RA	RW	0x1	Настройка RA
2:1	RM	RW	0x0	Настройка RM
0	RME	RW	0x1	Настройка RME

### 2.10.12.10 Регистр PWRCTR\_SRAM\_TRIM1

Таблица 2.31. Регистр PWRCTR\_SRAM\_TRIM1

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0x0	Резерв
10:8	WPULSE	RW	0x0	Настройка WPULSE
7:5	WA	RW	0x5	Настройка WA
4:3	RA	RW	0x0	Настройка RA

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
2:1	RM	RW	0x2	Настройка RM
0	RME	RW	0x1	Настройка RME

### 2.10.12.11 Регистр PWRCTR\_SRAM\_TRIM2

Таблица 2.32. Регистр PWRCTR\_SRAM\_TRIM2

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0x0	Резерв
10:8	WPULSE	RW	0x0	Настройка WPULSE
7:5	WA	RW	0x4	Настройка WA
4:3	RA	RW	0x0	Настройка RA
2:1	RM	RW	0x3	Настройка RM
0	RME	RW	0x1	Настройка RME

## 2.11 Сброс

На следующем рисунке показана схема сброса микросхемы.

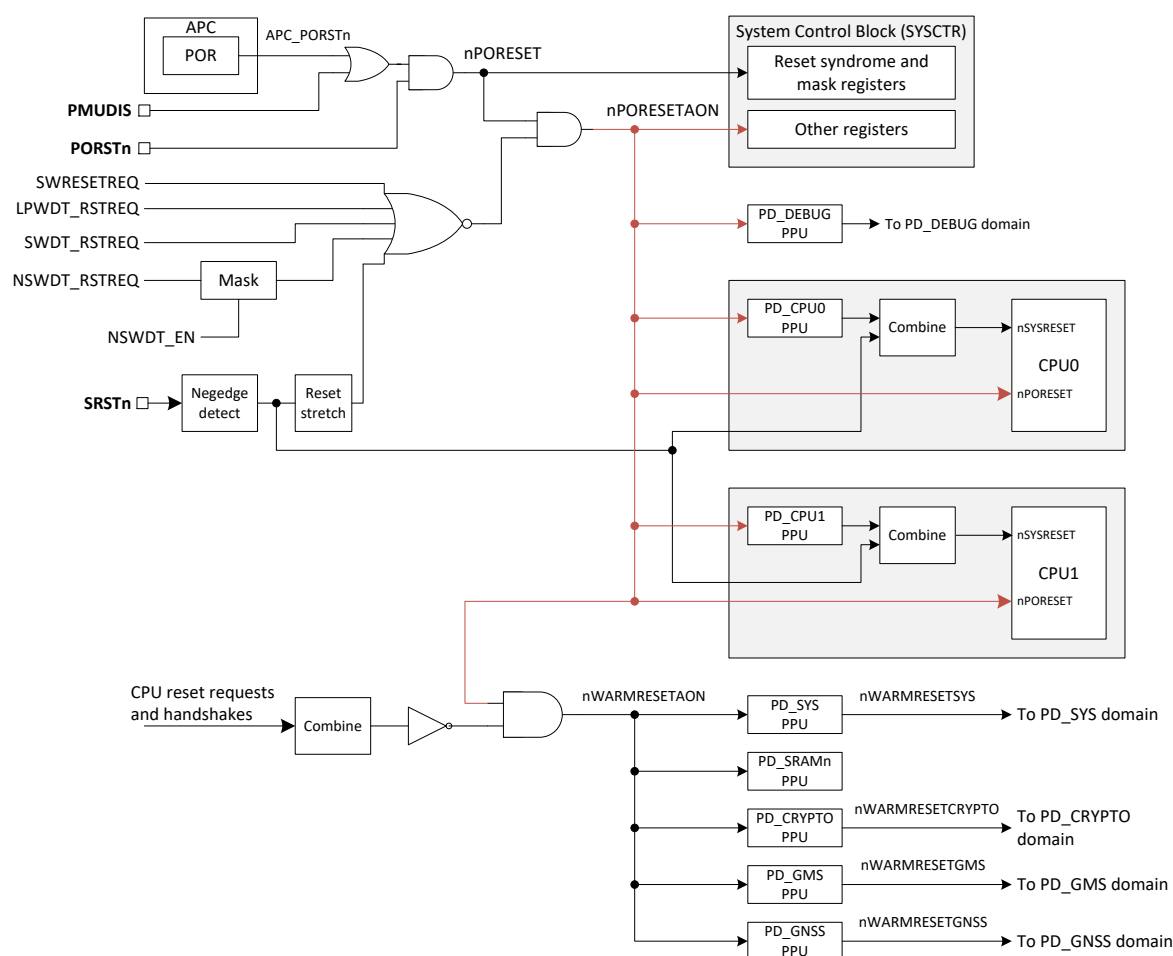


Рисунок 2.9. Схема сброса K1892BG1Я

### 2.11.1 Источники сброса

Система имеет несколько источников сброса:

- APC\_PORSTn – сигнал сброса от монитора питания в блоке APC (2.10.5). При необходимости, деактивируется установкой на выводе PMUDIS логической единицы.
- PORSTn – сигнал сброса от внешнего монитора питания. Может быть использован вместо APC\_PORSTn или совместно с ним.
- SWRESETREQ – программный сброс. Иницируется через регистр **SYSCTR\_SWRESET**.
- LPWDT\_RSTREQ – сброс от сторожевого таймера LPWDT.
- SWDT\_RSTREQ – сброс от сторожевого таймера SWDT.
- NSWDT\_RSTREQ – сброс от сторожевого таймера NSWDT. Сброс разрешен при установке бита NSWDT\_EN в регистре **SYSCTR\_RESET\_MASK**.

- SRSTn – внешний системный сброс.
- Запросы системного сброса от процессоров CPU0 и CPU1. Иницируются установкой бита SYSRESETREQ в регистре Application Interrupt and Reset Control Register (AIRCR).

### 2.11.2 Power-on («холодный») сброс

Сигналы «холодного» сброса nPORESET и nPORESETAON формируются от внутреннего или внешнего мониторов питания при включении микросхемы и при падении напряжения питания ниже определенного уровня.

Сигнал nPORESET напрямую сбрасывает регистры **RESET\_SYNDROME** и **RESET\_MASK** в блоке SYSTR.

Сигнал nPORESET также комбинируется с программным сбросом SWRESETREQ, запросами сброса от сторожевых таймеров и импульсом системного сброса nSRST. Полученный сигнал nPORESETAON используется для сброса всей остальной части системы.

Сигнал nPORESETAON сбрасывает все контроллеры PPU для каждого отключаемого домена. Затем каждый PPU формирует сброс для своего контролируемого домена.

nPORESETAON также подается на выводы nPORESET процессоров Cortex-M33.

### 2.11.3 nWARMRESETAON

Сигнал nWARMRESETAON выполняет «теплый» сброс системы. Этот сигнал также сбрасывает контроллеры PPU доменов PD\_SYS, PD\_SRAMn, PD\_CRYPT0, PD\_GMS, PD\_GNSS, которые затем выполняют сброс соответствующих доменов с помощью сигналов nWARMRESET\*.

### 2.11.4 SRSTn

Сигнал SRSTn предназначен для сброса системы с помощью внешнего отладчика. Этот сигнал проходит через схему детектирования фронта (Negedge detect) и схему формирования импульса (Reset stretch). Сигнал с выхода первой схемы служит для сброса процессоров и удержания их в этом состоянии, пока на вывод SRSTn подается низкий логический уровень. Вторая схема формирует импульс сброса длительностью 3 такта частоты LPCLK, используемый в качестве «холодного» сброса всей системы.

Вывод SRSTn может удерживаться в низком уровне для предотвращения загрузки процессоров во время выполнения отладочных манипуляций (например, программирование Flash памяти, загрузка сертификата безопасности в SRAM0 и т. д.).

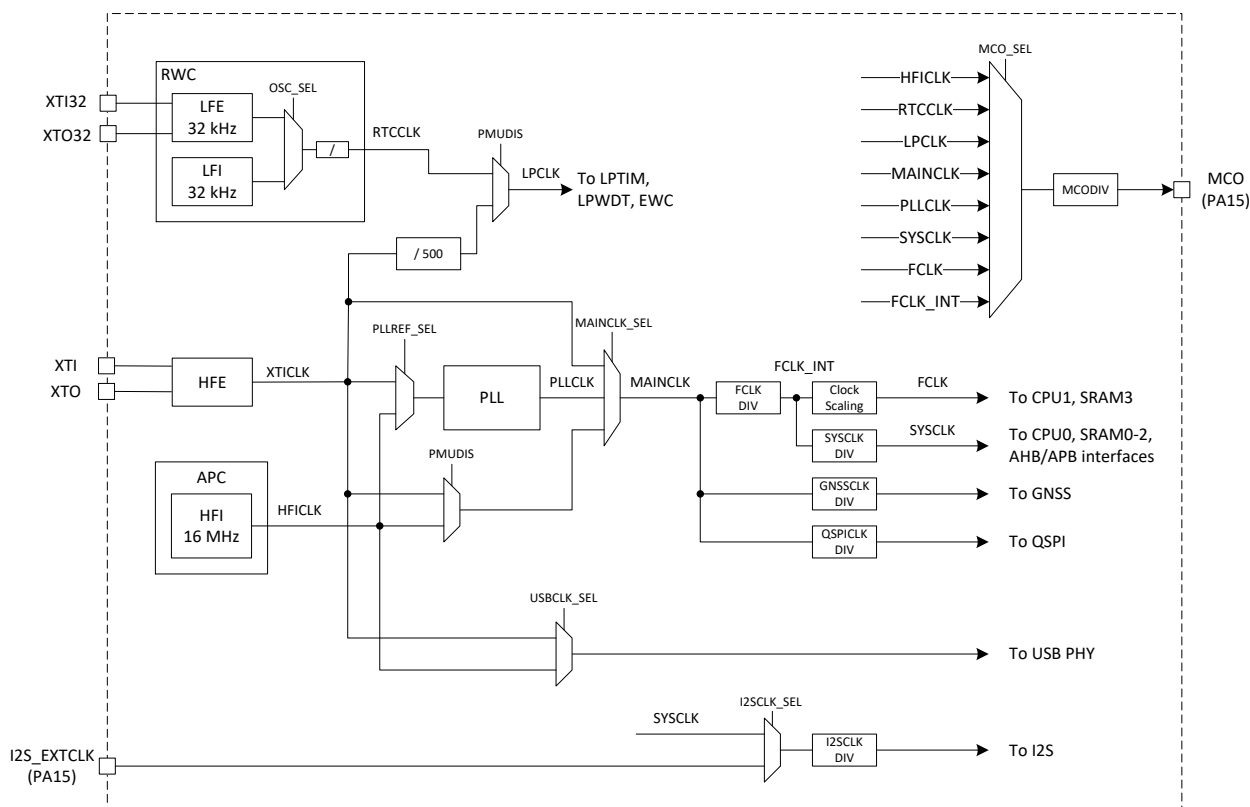


Длительность удержания вывода SRSTn в низком состоянии должна быть не менее трех тактов частоты LPCLK.

## 2.12 Тактирование

### 2.12.1 Обзор

На Рисунок 2.10 показана схема формирования тактовых сигналов K1892ВГ1Я.



**Рисунок 2.10. Схема формирования тактовых сигналов K1892ВГ1Я**

Микросхема имеет четыре источника тактовых сигналов:

- высокочастотный осциллятор HFE, используемый совместно с внешним кварцевым резонатором с максимальной частотой работы 48 МГц. Осциллятор также может работать в качестве буфера для подключения к выводу XTI внешнего генератора с частотой до 50 МГц;
- внутренний высокочастотный RC-осциллятор (HFI) на частоту 16 МГц. Точность установки частоты составляет  $\pm 5\%$ , имеется возможность программной подстройки частоты;
- внутренний низкочастотный RC-осциллятор (LFI) на частоту 32768 Гц. Точность установки частоты –  $\pm 2\%$ , имеется возможность программной подстройки частоты;
- внутренний низкочастотный осциллятор (LFE), используемый совместно с внешним кварцевым резонатором. Частота – 32768 Гц. Осциллятор также может работать в качестве буфера для подключения внешнего генератора.

В качестве основного системного тактового сигнала MAINCLK может использоваться:

- HFICLK (выбран после «холодного» сброса микросхемы при PMUDIS = 0);
- XTICLK;
- PLLCLK.

В качестве опорной частоты блока PLL могут быть выбраны тактовые сигналы:

- HFICLK;
- XTICLK.

После «холодного» сброса микросхемы вход опорной частоты PLL подключен к HFICLK.

MAINCLK служит для получения основных тактовых сигналов в системе:

- FCLK: используется для тактирования CPU1 и банка памяти SRAM3. Получается путем деления MAINCLK на коэффициент FCLKDIV ( $1 \div 32$ );
- SYSCLK: тактирование CPU0, основной матрицы коммутации, интерфейсов AHB и APB и большей части периферийных блоков. Получается путем деления FCLK на коэффициент SYSCLKDIV ( $1 \div 32$ ).

Тактирование всех периферийных блоков выполняется от SYSCLK за следующим исключением:

- тактовый сигнал блока GNSS получается путем деления MAINCLK на коэффициент GNSSCLKDIV ( $1 \div 8$ );
- тактовый сигнал интерфейса QSPI получается путем деления MAINCLK на коэффициент QSPICLKDIV ( $1 \div 32$ );
- опорная тактовая частота USB трансивера может быть получена от:
  - HFICLK;
  - XTICLK с частотой 16.0, 19.2, 20.0, 24.0, 26.0 или 32.0 МГц. Стабильность тактового сигнала должна быть не хуже  $\pm 400$  ppm, джиттер – не более  $\pm 100$  пс.
- тактовая частота интерфейса I2S получается делением на коэффициент I2SCLKDIV ( $1 \div 32$ ) одного из следующих тактовых сигналов:
  - SYSCLK;
  - внешний тактовый сигнал, подключенный к многофункциональному выводу с функцией I2S\_EXTCLK.

Сигнал RTCCLK формируется в батарейном домене питания и может быть получен:

- от осциллятора LFI (по умолчанию);
- от осциллятора LFE.

Сигнал RTCCLK используется в батарейном домене питания для тактирования часов реального времени RTC и контроллера внешнего прерывания WKUP.

Сигнал LPCLK в нормальном режиме (при PMUDIS = 0) подключен к RTCCLK и служит для тактирования таймеров LPTIM, LPWDT, контроллеров EWC. При установке высокого логического уровня на выводе PMUDIS сигнал LPCLK формируется делением на 500 внешнего тактового сигнала XTICKL.

Для отладочных целей предусмотрен выход MCO, к которому через делитель с программируемым коэффициентом MCODIV можно подключить один из тактовых сигналов HFICKL, RTCCLK, LPCLK, MAINCLK, FCLK, FCLK\_INT, SYSCCLK.

Периферийные блоки SMC и SDMMC имеют встроенные делители для формирования сигналов тактирования интерфейсов (см. соответствующие разделы данного руководства).

## 2.12.2 Осциллятор HFI

Осциллятор HFI, встроенный в контроллер APC, генерирует частоту 16 МГц с точностью  $\pm 5\%$ .

Тактовый сигнал осциллятора HFICKL может быть использован в качестве опорного сигнала PLL, тактового сигнала MAINCLK и USBCLK.

Работа трансивера USB при тактировании от HFICKL возможна с некоторыми ограничениями.

Осциллятор имеет возможность калибровки частоты с помощью параметра TRIM регистра **CLKCTR\_HFITRIM**. Величины подстройки частоты  $\Delta F_{\text{HFICKL}}$  от номинального значения 16 МГц, приведенные в следующей таблице, являются приблизительными и могут быть уточнены при испытаниях микросхемы. После сброса значение параметра TRIM автоматически обновляется из памяти OTP.

**Таблица 2.33. Параметры для подстройки частоты осциллятора HFI**

TRIM	$\Delta F_{\text{HFICKL}}$ , %	TRIM	$\Delta F_{\text{HFICKL}}$ , %	TRIM	$\Delta F_{\text{HFICKL}}$ , %	TRIM	$\Delta F_{\text{HFICKL}}$ , %
0x00	0.0	0x08	27.5	0x1F	-3.0	0x17	-41.0
0x01	3.5	0x09	30.5	0x1E	-9.0	0x16	-46.0
0x02	7.0	0x0A	34.0	0x1D	-13.5	0x15	-50.5
0x03	10.0	0x0B	37.5	0x1C	-18.5	0x14	-55.0
0x04	13.5	0x0C	41.0	0x1B	-23.0	0x13	-59.5
0x05	17.0	0x0D	44.5	0x1A	-27.5	0x12	-64.0
0x06	20.5	0x0E	47.5	0x19	-32.0	0x11	-68.5
0x07	24.0	0x0F	51.0	0x18	-36.5	0x10	-73.5

## 2.12.3 Осцилляторы LFI и LFE

Кварцевый осциллятор LFE и RC-осциллятор LFI встроены в блок RWC, расположенный в резервном домене питания, что позволяет им функционировать в любом режиме работы микросхемы. Оба осциллятора генерируют тактовые сигналы с частотой 32768 Гц, необходимые для работы часов реального времени RTC и контроллера внешнего

прерывания WKUP (сигнал RTCCLK), а также таймеров LPWDT, LPTIM, контроллеров EWC и некоторых служебных схем в домене PD\_AON (сигнал LPCLK).

Выбор используемого осциллятора производится параметром OSC\_SEL.

Блок RWC имеет делитель для понижения частоты RTCCLK и LPCLK вплоть до 1/64 Гц.

Кварцевый осциллятор LFE имеет встроенные конденсаторы и требует подключения только внешнего резонатора. Также LFE может использоваться как буфер для внешнего тактового сигнала.

Осциллятор LFI имеет возможность калибровки частоты в диапазоне  $\pm 512$  ppm. После сброса значение калибровочного параметра автоматически обновляется из памяти OTP.

Все настройки осцилляторов LFI и LFE выполняются в блоке RWC (см. соответствующий раздел данного руководства).

## 2.12.4 PLL

Блок фазовой автоподстройки частоты (PLL) может быть использован для формирования основной частоты системы MAINCLK. В качестве опорного сигнала PLL может быть выбран HFICLK (по умолчанию) или XTICLK. Частота опорного тактового сигнала PLL должна быть не менее 30 кГц.

Выходная частота PLL определяется по формуле:

$$F_{PLLCLK} = F_{REFCLK} \times \frac{NF}{NR \times OD}$$

где  $F_{REFCLK}$  – частота опорного сигнала PLL,  $NR$  – входной делитель опорной частоты,  $NF$  – делитель в петле обратной связи,  $OD$  – выходной делитель.

Настройка выходной частоты PLL может выполняться двумя способами:

1. С помощью предустановок – делители  $NR$ ,  $NF$ ,  $OD$  выбираются автоматически в зависимости от значения параметра SEL регистра **CLKCTR\_PLLCFG**. При этом параметр MAN = 0.
2. Делители  $NR$ ,  $NF$ ,  $OD$  задаются параметрами NR\_MAN, NF\_MAN, OD\_MAN соответственно. При этом MAN = 1, SEL  $\neq$  0.

При установке SEL = 0 блок PLL переходит в малопотребляющий режим, при этом на выход передается опорная частота.

После изменения параметров PLL необходимо дождаться стабилизации выходной частоты. Это можно сделать с помощью проверки бита LOCK регистра **CLKCTR\_PLLCFG**.

Корректное функционирование PLL гарантируется при выходной частоте в диапазоне 30 МГц – 375 МГц (без учета выходного делителя, OD = 1).

### 2.12.5 Настройка MAINCLK, FCLK, SYSCLK

MAINCLK служит для получения системных тактовых сигналов FCLK и SYSCLK, а также сигналов QSPICK и GNSSCLK. Сигнал MAINCLK, в свою очередь, может быть подключен к внутреннему осциллятору LFI, внешнему выводу XT1 либо выходу PLL.

Источник MAINCLK может быть выбран с помощью параметра MAINCLK\_SEL без остановки работы процессоров, при переключении сигнал MAINCLK не будет иметь паразитных импульсов. Важно учесть, что тактовый сигнал, на который переключается MAINCLK должен быть включен и стабилен.

Перед изменением конфигурации PLL мультиплексор MAINCLK должен быть переключен на стабильный тактовый сигнал XTICK или HFICK. После установки бита LOCK в регистре **CLKCTR\_PLLCFG** сигнал PLLCLK может быть использован снова.

Перед переключением MAINCLK делители производных от MAINCLK тактовых сигналов FCLK, SYSCLK, QSPICK и GNSSCLK должны быть настроены так, чтобы частоты производных сигналов после переключения не превышали максимально допустимых значений, указанных в разделе “Справочная информация”.

Для изменения значения делителя FCLKDIV нужно выполнить такую последовательность действий:

- изменить значение SYSCLKDIV в регистре **CLKCTR\_SYSCLKDIV** с учетом новой частоты FCLK;
- дождаться обновления значения SYSCLKDIV\_CUR;
- изменить значение FCLKDIV в регистре **CLKCTR\_FCLKDIV**;
- дождаться обновления значения FCLKDIV\_CUR.

### 2.12.6 Методы снижения энергопотребления

Для уменьшения энергопотребления в контексте системы тактирования предусмотрены следующие возможности.

- Выключение тактового сигнала домена при переходе в состояние OFF/MEM\_RET. Выполняется контроллером PPU соответствующего домена.
- Динамическое управление тактированием – выключение тактового сигнала при отсутствии активности устройства и возобновление тактирования при необходимости.

Применяется для следующих устройств в системе:

- Блоки ACG и различные периферийные устройства домена PD\_SYS, тактируемые с помощью SYSCLK и FCLK. Функция включается битами

**SYS\_SYSCLK\_FORCE** и **SYS\_FCLK\_FORCE** регистра **CLKCTR\_CLKFORCE** соответственно.

- Банки памяти SRAM0-2 (бит **SRAM\_SYSCLK\_FORCE**) и SRAM3 (бит **SRAM\_FCLK\_FORCE**).
- Блоки ACG, мосты синхронизации и кэш инструкций в подсистемах CPU. Функция включается битами **CPU\_SYSCLK\_FORCE** для подсистемы CPU0 и **CPU\_FCLK\_FORCE** для подсистемы CPU1.
- Блок CryptoCell (бит **CRYPTO\_SYSCLK\_FORCE**).
- Блок GMS (бит **GMS\_SYSCLK\_FORCE**).
- Интерфейсный тактовый сигнал контроллера SMC (бит **SMCCLK\_FORCE**).
- Тактовый сигнал контроллеров PPU доменов **PD\_CPU<sub>n</sub>** и **PD\_DEBUG** управляется битом **CPUDBG\_PIKCLK\_FORCE**.
- Тактовый сигнал контроллеров PPU доменов **PD\_SYS**, **PD\_SRAM<sub>n</sub>**, **PD\_CRYPTO**, **PD\_GNSS** и **PD\_GMS** управляется битом **BASE\_PIKCLK\_FORCE**.

Функция динамического управления тактированием включается при записи в соответствующий бит **\*CLK\_FORCE** значения 0. По умолчанию, все биты **\*CLK\_FORCE** равны 1, и тактирование перечисленных выше блоков не останавливается.

- Автоматическое изменение частоты FCLK. При установке бита **FCLK\_SCALE\_EN = 1** в регистре **CLKCTR\_CFG** частота FCLK автоматически снижается до частоты SYSCLK при выключении домена **PD\_CPU1**. При включении домена восстанавливается исходная частота FCLK.

Для уменьшения задержки доступа к банку памяти SRAM3 при выключенном процессоре CPU1 бит **FCLK\_SCALE\_EN** должен быть равен 0.

## 2.12.7 Регистры управления тактированием

Регистры управления тактированием расположены в блоке CLKCTR. Поддерживаются обращения с разрядностью 8, 16 и 32 бит.

### 2.12.7.1 Перечень регистров CLKCTR

**Таблица 2.34. Регистры CLKCTR**

Условное обозначение	Название регистра	Исходное состояние	Смещение
CLKCTR_PLLCFG	Регистр настройки PLL	0x0000_0000	0x000
CLKCTR_PLLDIAG	Регистр диагностики PLL	0x0000_0000	0x004
CLKCTR_CFG	Регистр общих настроек тактирования	0x0000_0000	0x008
CLKCTR_CLKFORCE	Регистр настройки динамического управления тактированием	0x0000_0FFF	0x00C
CLKCTR_FCLKDIV	Регистр настройки делителя сигнала FCLK	0x0000_0000	0x010
CLKCTR_SYSCLKDIV	Регистр настройки делителя сигнала SYSCLK	0x0000_0000	0x014
CLKCTR_QSPICLKDIV	Регистр настройки делителя сигнала QSPICLK	0x0000_0000	0x018
CLKCTR_I2SCLKDIV	Регистр настройки делителя сигнала I2SCLK	0x0000_0000	0x01C
CLKCTR_MCODIV	Регистр настройки делителя сигнала MCOCLK	0x0000_0000	0x020
CLKCTR_GNSSCLKDIV	Регистр настройки делителя сигнала GNSSCLK	0x0000_0000	0x024
-	Резерв	0x0000_0000	0x028 - 0x02C
CLKCTR_HFITRIM	Регистр подстройки частоты осциллятора HFI	0x0000_0000	0x030

### 2.12.7.2 Регистр CLKCTR\_PLLCFG

**Таблица 2.35. Регистр CLKCTR\_PLLCFG**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	LOCK	RO	0	Бит указывает окончание настройки PLL на новую заданную частоту работы. При начале процедуры настройки либо при выключенной PLL бит сбрасывается в 0
30:27	NR_MAN	RW	0	Коэффициент NR настройки PLL: $NR = NR\_MAN + 1$
26:14	NF_MAN	RW	0	Коэффициент NF настройки PLL: $NF = NF\_MAN + 1$
13:10	OD_MAN	RW	0	Коэффициент OD настройки PLL: $OD = OD\_MAN + 1$
9	MAN	RW	0	Способ настройки выходной частоты PLL: 0x0: делители NR, NF, OD определяются полем SEL данного регистра; 0x1: делители задаются полями NR_MAN, NF_MAN, OD_MAN данного регистра.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
8:0	SEL	RW	0	<p>Определяет выходную частоту и режим работы PLL:</p> <p>0x000: опорная частота REFCLK. Блок PLL выключен;</p> <p>0x001: <math>F_{REFCLK} * 2</math>;</p> <p>0x002: <math>F_{REFCLK} * 3</math>;</p> <p>0x003: <math>F_{REFCLK} * 4</math>;</p> <p>...</p> <p>0x176 - 0x1FF: <math>F_{REFCLK} * 375</math>.</p> <p>При SEL &gt; 0 и MAN = 1, конфигурация PLL определяется полями NR_MAN, NF_MAN, OD_MAN данного регистра.</p>

### 2.12.7.3 Регистр CLKCTR\_PLLDIAG

Регистр предназначен для тестирования блока PLL.

Таблица 2.36. Регистр CLKCTR\_PLLDIAG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв
4	MACRO_LOCK	RO	0	Состояние выхода LOCK блока PLL
3	-	RO	0	Резерв
2	FASTEN	RW	0	Установка этого бита разрешает механизм fast lock. В нормальном режиме работы бит равен 0
1	ENSAT	RW	0	Установка этого бита разрешает насыщение. В нормальном режиме работы бит равен 0
0	TEST	RW	0	Установка этого бита включает тестовый режим работы PLL

### 2.12.7.4 Регистр CLKCTR\_CFG

Таблица 2.37. Регистр CLKCTR\_CFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:20	-	RO	0	Резерв
19:18	HFE_FREQ	RW	0x2	<p>Диапазон частоты осциллятора HFE при работе с кварцевым резонатором:</p> <p>0x0: 1.0 – 4.0 МГц;</p> <p>0x1: 4.1 – 12.0 МГц;</p> <p>0x2: 12.1 – 24.0 МГц;</p> <p>0x3: 24.1 – 48.0 МГц.</p>
17:16	HFE_MODE	RW	0x2	<p>Режим работы кварцевого осциллятора HFE:</p> <p>0x0: осциллятор выключен;</p> <p>0x1: режим работы с кварцевым резонатором или внешним дифференциальным тактовым сигналом;</p> <p>0x2: режим работы с внешним тактовым сигналом;</p> <p>0x3: резерв.</p>
15:14	-	RO	0	Резерв
13	MCO_EN	RW	0	Включение тактового сигнала MCO
12	-	RO	0	Резерв



Номер бита	Название	Тип доступа	Исходное состояние	Назначение
11:9	MCO_SEL	RW	0	Выбор источника тактового сигнала MCO: 0x0: HFICLK; 0x1: RTCCLK; 0x2: LPCLK; 0x3: MAINCLK; 0x4: PLLCLK; 0x5: SYSCLK; 0x6: FCLK_INT; 0x7: FCLK
8	I2SCLK_SEL	RW	0	Выбор тактового сигнала интерфейса I2S: 0: SYSCLK; 1: I2S_EXTCLK
7	USBCLK_SEL	RW	0	Выбор опорного тактового сигнала трансивера USB PHY: 0: HFICLK; 1: XTICLK
6	PLLREF_SEL	RW	0	Выбор опорного тактового сигнала PLL: 0: HFICLK; 1: XTICLK
5:4	MAINCLK_SEL	RW	0	Выбор источника тактового сигнала MAINCLK: 0x0: HFICLK; 0x1: XTICLK; 0x2: PLLCLK; 0x3: резерв
3:1	-	RO	0	Резерв
0	FCLK_SCALE_EN	RW	0	Значение 1 включает функцию автоматического понижения частоты FCLK при выключении питания процессора CPU 1

### 2.12.7.5 Регистр CLKCTR\_CLKFORCE

Таблица 2.38. Регистр CLKCTR\_CLKFORCE

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:12	-	RO	0	Резерв
11	GMS_SYSCLK_FORCE	RW	1	Управление динамическим включением/выключением тактирования (см. п. 2.12.6): 0: функция активна; 1: функция не активна, непрерывное тактирование
10	SMCCLK_FORCE	RW	1	
9	BASE_PIKCLK_FORCE	RW	1	
8	CPUDBG_PIKCLK_FORCE	RW	1	
7	CRYPTO_SYSCLK_FORCE	RW	1	
6	CPU_FCLK_FORCE	RW	1	
5	CPU_SYSCLK_FORCE	RW	1	
4	SRAM_FCLK_FORCE	RW	1	
3	SRAM_SYSCLK_FORCE	RW	1	
2	SYS_FCLK_FORCE	RW	1	
1	SYS_SYSCLK_FORCE	RW	1	
0	-	RO	1	Резерв

### 2.12.7.6 Регистр CLKCTR\_FCLKDIV

Таблица 2.39. Регистр CLKCTR\_FCLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	FCLKDIV_CUR	RW	0	Текущее значение делителя FCLKDIV
15:5	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
4:0	FCLKDIV	RW	0	Задаёт значение делителя для тактового сигнала FCLK как (FCLKDIV + 1)

### 2.12.7.7 Регистр CLKCTR\_SYSCCLKDIV

Таблица 2.40. Регистр CLKCTR\_SYSCCLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	SYSCCLKDIV_CUR	RW	0	Текущее значение делителя SYSCCLKDIV
15:5	-	RO	0	Резерв
4:0	SYSCCLKDIV	RW	0	Задаёт значение делителя для тактового сигнала SYSCCLK как (SYSCCLKDIV + 1)

### 2.12.7.8 Регистр CLKCTR\_QSPICLKDIV

Таблица 2.41. Регистр CLKCTR\_QSPICLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	QSPICLKDIV_CUR	RW	0	Текущее значение делителя QSPICLKDIV
15:5	-	RO	0	Резерв
4:0	QSPICLKDIV	RW	0	Задаёт значение делителя для тактового сигнала QSPICLK как (QSPICLKDIV + 1)

### 2.12.7.9 Регистр CLKCTR\_I2SCLKDIV

Таблица 2.42. Регистр CLKCTR\_I2SCLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	I2SCLKDIV_CUR	RW	0	Текущее значение делителя I2SCLKDIV
15:5	-	RO	0	Резерв
4:0	I2SCLKDIV	RW	0	Задаёт значение делителя для тактового сигнала I2SCLK как (I2SCLKDIV + 1)

### 2.12.7.10 Регистр CLKCTR\_MCODIV

Таблица 2.43. Регистр CLKCTR\_MCODIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	MCODIV_CUR	RW	0	Текущее значение делителя MCODEV
15:5	-	RO	0	Резерв
4:0	MCODIV	RW	0	Задаёт значение делителя для тактового сигнала MCODEV как (MCODEV + 1)

### 2.12.7.11 Регистр CLKCTR\_GNSSCLKDIV

Таблица 2.44. Регистр CLKCTR\_GNSSCLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:19	-	RO	0	Резерв
18:16	GNSSCLKDIV_CUR	RW	0	Текущее значение делителя GNSSCLKDIV

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:3	-	RO	0	Резерв
2:0	GNSSCLKDIV	RW	0	Задаёт значение делителя для тактового сигнала GNSSCLK как (GNSSCLKDIV + 1)

### 2.12.7.12 Регистр CLKCTR\_HFITRIM

Таблица 2.45. Регистр CLKCTR\_HFITRIM

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
4:0	TRIM	RW	0	Подстройка частоты осциллятора HFI (Таблица 2.33). После сброса автоматически обновляется из памяти OTP.

## 2.13 Загрузка

### 2.13.1 Адрес загрузки

После сброса процессоров начальный адрес загрузки определяется регистрами SYSCTR\_INITSVTOR0 и SYSCTR\_INITSVTOR1 для ядер CPU0 и CPU1 соответственно. Адрес указывает на начало таблицы векторов прерываний и после сброса заносится в регистр VTOR\_S каждого процессора (см. Arm® Cortex®-M33 Processor Technical Reference Manual).

После «холодного» сброса (сигнал nPORESETAON) значение регистров SYSCTR\_INITSVTORx устанавливается в зависимости от бита OTP\_BOOT\_ADDR\_SEL в соответствии со следующей таблицей.

Таблица 2.46. Определение начального адреса загрузки

OTP_BOOT_ADDR_SEL	SYSCTR_INITSVTOR0, SYSCTR_INITSVTOR1	Тип памяти
0	0x1020_0000	Системный раздел FLASH
1	OTP_BOOT_ADDR << 7	FLASH, OTP, SRAMn

Если бит OTP\_BOOT\_ADDR\_SEL установлен, то адрес загрузки задается параметром OTP\_BOOT\_ADDR. Загрузка может производиться из встроенной памяти FLASH, пользовательского раздела OTP или одного из банков SRAMn.

При OTP\_BOOT\_ADDR\_SEL = 0 (исходное состояние OTP) адрес загрузки равен 0x1020\_0000, что соответствует системному разделу FLASH памяти.

Начальный адрес загрузки одинаков для обоих ядер CPU и должен быть выровнен на 128 слов (512 байт).

Параметры OTP\_BOOT\_ADDR\_SEL и OTP\_BOOT\_ADDR расположены в OTP памяти по адресу 0x2C (см. раздел 26 «OTP: Однократно программируемая память»). Допускается только однократное изменение указанных параметров и только в состояниях LCS = CM, DM.

Регистры SYSCTR\_INITSVTORx могут быть изменены программно перед выполнением «теплого» сброса (с использованием CPU<sub>n</sub>\_PPU) либо с помощью внешнего отладчика при удержании SRST<sub>n</sub> = 0. Допускаются только Secure обращения к регистрам.

Согласно требованию TrustZone для архитектуры ARMv8-M загрузка должна выполняться из доверенной области памяти. Затем, после настройки параметров безопасности системы, может выполняться недоверенная программа. При «холодном» старте вся память системы является доверенной. Для работы недоверенного приложения необходимо настроить блоки MPC.

### 2.13.2 Отложенная загрузка

Каждое ядро Cortex-M33 имеет вход CPUWAIT с помощью которого исполнение инструкций может быть отложено. Для управления входом CPUWAIT в регистре SYSCTR\_CPUWAIT предусмотрен соответствующий бит для каждого процессора. Исходное состояние после снятия сброса nPORESETAON следующее: CPU0WAIT = 0, CPU1WAIT = 1. Т.е. после «холодного» сброса микросхемы загрузку выполняет только ядро CPU0. Загрузка ядра CPU1, а также включение питания домена PD\_CPU1 заблокированы.

Параметры CPU0WAIT и CPU1WAIT могут быть программно изменены перед выполнением «теплого» сброса либо с помощью внешнего отладчика при удержании SRST<sub>n</sub> = 0.

## 2.14 Идентификация микросхемы

Для идентификации микросхемы предусмотрен регистровый блок SYSINFO.

### 2.14.1 Перечень регистров SYSINFO

Перечень регистров SYSINFO приведен в следующей таблице.

**Таблица 2.47. Регистры SYSINFO**

Условное обозначение	Описание	Исходное состояние	Смещение
SYSINFO_SYS_VERSION	Регистр версии системы	0x03600219	0x000
SYSINFO_UID0	Регистр уникального идентификатора, биты [31:0]	0x00000000 <sup>1</sup>	0x010
SYSINFO_UID1	Регистр уникального идентификатора, биты [63:32]	0x00000000 <sup>1</sup>	0x014

SYSINFO_UID2	Регистр уникального идентификатора, биты [95:64]	0x00000000 <sup>1</sup>	0x018
SYSINFO_UID3	Регистр уникального идентификатора, биты [127:96]	0x00000000 <sup>1</sup>	0x01C
Примечания 1. Значения регистров уникального идентификатора SYSINFO_UID0 - SYSINFO_UID3 после сброса автоматически обновляются из памяти OTP.			

## 2.14.2 Регистр SYSINFO\_SYS\_VERSION

Формат регистра SYSINFO\_SYS\_VERSION приведен в следующей таблице.

**Таблица 2.48. Поля регистра SYSINFO\_SYS\_VERSION**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:28	CONFIGURATION	RO	0x0	Резерв
27:24	MAJOR_REVISION	RO	0x3	Старшая часть номера ревизии
23:20	MINOR_REVISION	RO	0x6	Младшая часть номера ревизии
19:16	-	RO	0x0	Резерв
15:0	PART_NUMBER	RO	0x0219	Номер компонента

## 3. SYSCTR: РЕГИСТРЫ УПРАВЛЕНИЯ СИСТЕМОЙ

### 3.1 Введение

Блок SYSCTR содержит регистры для управления различными системными функциями, такими как безопасность, сброс, питание, загрузка.

Поддерживается только доверенный привилегированный доступ к регистрам.

Поддерживается только запись 32-х разрядными словами.

Все регистры находятся в домене PD\_AON. Регистры SYSCTR\_RESET\_SYNDROME и SYSCTR\_RESET\_MASK сбрасываются по сигналу nPORESET. Все остальные регистры блока сбрасываются по сигналу nPORESETAON, если не указано иного.

### 3.2 Регистры

#### 3.2.1 Перечень регистров SYSCTR

Таблица 3.1. Регистры SYSCTR

Условное обозначение	Описание	Исходное состояние	Смещение
SYSCTR_SECDBGSTA T	Регистр состояния сигналов контроля отладочного доступа	0x0000_0000	0x000
-	Резерв	0x0000_0000	0x004
-	Резерв	0x0000_0000	0x008
SYSCTR_SCSECCTRL	Регистр управления безопасностью системы.	0x0000_0000	0x00C
-	Резерв	0x0000_0000	0x010 – 0x0FC
SYSCTR_RESET_ SYNDROME	Регистр синдрома сброса.	00000_0001	0x100
SYSCTR_RESET_MAS K	Регистр маски сброса.	0x0000_0000	0x104
SYSCTR_SWRESET	Регистр программного сброса.	0x0000_0000	0x108
SYSCTR_GRETRREG	Регистр общего назначения.	0x0000_0000	0x10C
SYSCTR_INITSVTOR0	Регистр начального значения адреса таблицы векторов для CPU0.	0x1020_0000	0x110
SYSCTR_INITSVTOR1	Регистр начального значения адреса таблицы векторов для CPU1.	0x1020_0000	0x114
SYSCTR_CPUWAIT	Регистр управления отложенной загрузкой CPU.	0x0000_0002	0x118
SYSCTR_NMI_ENABL E	Регистр управления немаскируемым прерыванием NMI.	0x0000_0001	0x11C
SYSCTR_WICCTRL	Регистр управления контроллерами WIC.	0x0000_0000	0x120
SYSCTR_EWCCTRL	Регистр управления контроллерами EWC.	0x0000_0000	0x124
-	Резерв	0x0000_0000	0x128 – 0x1FF
SYSCTR_PDCM_ SYS_SENSE	Регистр настройки зависимостей PDCM для домена PD_SYS.	0x0000_007F	0x200
-	Резерв	0x0000_0000	0x204
-	Резерв	0x0000_0000	0x208

SYSCTR_PDCM_SRAM0_SENSE	Регистр настройки зависимостей PDCM для домена PD_SRAM0.	0x0000_0000	0x20C
SYSCTR_PDCM_SRAM1_SENSE	Регистр настройки зависимостей PDCM для домена PD_SRAM1.	0x0000_0000	0x210
SYSCTR_PDCM_SRAM2_SENSE	Регистр настройки зависимостей PDCM для домена PD_SRAM2.	0x0000_0000	0x214
SYSCTR_PDCM_SRAM3_SENSE	Регистр настройки зависимостей PDCM для домена PD_SRAM3.	0x0000_0000	0x218

### 3.2.2 Регистр SYSCTR\_SECDBGSTAT

Регистр отображает состояние сигналов DBGEN, NIDEN, SPIDEN и SPNIDEN, контролирующих доступ внешнего отладчика к ресурсам системы. Управление этими сигналами производится с помощью регистров блока CryptoCell.

**Таблица 3.2. Регистр SYSCTR\_SECDBGSTAT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	-	RO	0	Резерв.
6	SPNIDEN_STATUS	RO	*	Состояние сигнала SPNIDEN.
5	-	RO	0	Резерв.
4	SPIDEN_STATUS	RO	*	Состояние сигнала SPIDEN.
3	-	RO	0	Резерв.
2	NIDEN_STATUS	RO	*	Состояние сигнала NIDEN.
1	-	RO	0	Резерв.
0	DBGEN_STATUS	RO	*	Состояние сигнала DBGEN.

Примечание: \* – исходное состояние зависит от значения LCS. См. раздел 27 «CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCELL».

### 3.2.3 Регистр SYSCTR\_SCSECCTRL

Регистр предназначен для управления доступом через интерфейс загрузки сертификата (DBG\_CERT), а также для блокировки изменения некоторых регистров.

**Таблица 3.3. Регистр SYSCTR\_SCSECCTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	CERTREADENAB LED	RO	0	Бит показывает состояние доступа на чтение через интерфейс DBG_CERT: 1: Чтение разрешено. 0: Чтение запрещено. Этот бит также сбрасывается в 0, когда: <ul style="list-style-type: none"> <li>• домен PD_DEBUG находится в состоянии OFF.</li> <li>• CERTDISABLED = 1.</li> </ul>
16	CERTDISABLED	RO	0	Бит показывает состояние доступа на запись через интерфейс DBG_CERT: 1: Запись запрещена. 0: Запись разрешена. Этот бит также устанавливается в 1 при выключении домена PD_DEBUG.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:3	-	RO	0	Резерв
2	SCSECCFGLOCK	Write 1 to set	0	Control to disable writes to security-related control registers in this register block. When set to 1, writes to SYSCTR_INITSVTOR0 and SYSCTR_INITSVTOR1 are ignored. The bit is only cleared to zero by a power-on reset.
1	CERTREADEN	RW	0	Управление доступом на чтение через интерфейс DBG_CERT: 1: Доступ на чтение разрешен, если CERTDISABLE = 0. 0: Доступ на чтение запрещен.
0	CERTDISABLE	Write 1 to set	0	Установка значения 1 отключает доступ через интерфейс DBG_CERT. Этот бит также аппаратно устанавливается в 1: <ul style="list-style-type: none"> <li>При выключении домена PD_DEBUG.</li> <li>При выключении домена PD_CPU0.</li> </ul> Запись 0 игнорируется. Бит может быть сброшен только по сигналу nPORESETAON.

### 3.2.4 Регистр SYSCTR\_RESET\_SYNDROME

Регистр показывает причину последнего события сброса.

Регистр очищается по сигналу nPORESET либо записью 0 в каждый бит.

Запись 1 не изменяет значение бита.

Биты LOCKUP0 и LOCKUP1 обозначают не сброс, а блокировку (lock-up) соответствующего процессора.

**Таблица 3.4. Регистр SYSCTR\_RESET\_SYNDROME**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:10	-	RO	0	Резерв
9	SWRESETREQ	Write 0 to clear	0	Программный сброс с помощью регистра SWRESET.
8	-	RO	0	Резерв
7	LOCKUP1	Write 0 to clear	0	Статус блокировки CPU1.
6	LOCKUP0	Write 0 to clear	0	Статус блокировки CPU0.
5	SYSRSTREQ1	Write 0 to clear	0	Системный сброс с помощью CPU1.
4	SYSRSTREQ0	Write 0 to clear	0	Системный сброс с помощью CPU0.
3	LPWDT	Write 0 to clear	0	Сброс от сторожевого таймера LPWDT.
2	SWDT	Write 0 to clear	0	Сброс от доверенного сторожевого таймера SWDT.
1	NSWDT	Write 0 to clear	0	Сброс от недоверенного сторожевого таймера NSWDT.
0	PoR	Write 0 to clear	1	Сброс по включению питания (Power-on).



### 3.2.5 Регистр SYSCTR\_RESET\_MASK

Регистр SYSCTR\_RESET\_MASK позволяет управлять влиянием источников сброса на формирование сигнала сброса системы nPORESETAON.

Установка каждого бита в 1 включает соответствующий источник сброса.

Сброс бита в 0 предотвращает сброс системы от данного источника, а также установку соответствующего бита регистра RESET\_SYNDROME.

**Таблица 3.5. Регистр SYSCTR\_RESET\_MASK**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:6	-	RO	0	Резерв
5	SYSRSTREQ1_EN	RW	0	Разрешение сброса по запросу от CPU1.
4	SYSRSTREQ0_EN	RW	0	Разрешение сброса по запросу от CPU0.
3:2	-	RO	0	Резерв
1	NSWDT_EN	RW	0	Разрешение сброса от NSWDT.
0	-	RO	0	Резерв

### 3.2.6 Регистр SYSCTR\_SWRESET

Регистр SWRESET позволяет выполнить программный сброс системы.

**Таблица 3.6. Регистр SYSCTR\_SWRESET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:10	-	RO	0	Резерв
9	SWRESETREQ	WO	0	Запись 1 приводит к «холодному» сбросу системы.
8:0	-	RO	0	Резерв

### 3.2.7 Регистр SYSCTR\_GRETREG

Регистр общего назначения. Сохраняет свое значение в режиме STANDBY.

**Таблица 3.7. Регистр SYSCTR\_GRETREG**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	GRETREG	RW	0	Регистр общего назначения.

### 3.2.8 Регистры SYSCTR\_INITSVTOR0, SYSCTR\_INITSVTOR1

Регистры содержат значение адреса доверенной таблицы векторов (VTOR\_STBLOFF[31:7]) для соответствующего ядра CPU<sub>n</sub>, n = 0, 1.

Исходное значение регистров после сброса зависит от флага OTP\_BOOT\_ADDR\_SEL в OTP памяти (см. раздел 2.13 «Загрузка»).

Адреса должны задаваться с выравниванием по 128 слов (512 байт).

В следующих таблицах показано назначение разрядов регистров.

**Таблица 3.8. Регистр SYSCTR\_INITSVTOR0**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	INITSVTOR0	RW	0x20_4000	Начальное значение адреса таблицы векторов для CPU0.
6:0	-	RO	0	Резерв

**Таблица 3.9. Регистр SYSCTR\_INITSVTOR1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	INITSVTOR1	RW	0x20_4000	Начальное значение адреса таблицы векторов для CPU1.
6:0	-	RO	0	Резерв

### 3.2.9 Регистр SYSCTR\_CPUWAIT

Регистр позволяет отложить исполнение инструкций и приостановить загрузку процессоров сразу после выполнения сброса. Это дает возможность внешнему отладчику выполнить необходимые действия до загрузки процессоров.

После «холодного» сброса микросхемы регистр также предотвращает включение питания домена PD\_CPU1. После выключения и последующего включения питания доменов PD\_CPU<sub>n</sub> без сброса системы регистр CPUWAIT не влияет на включение питания.

**Таблица 3.10. Регистр SYSCTR\_CPUWAIT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0	Резерв
1	CPU1WAIT	RW	1	Управление загрузкой CPU1: 0: загрузка выполняется в обычном режиме. 1: загрузка не выполняется до сброса данного бита. После «холодного» сброса этот бит также управляет включением домена PD_CPU1: 1: питание не включается до сброса данного бита. 0: питание включается.
0	CPU0WAIT	RW	0	Управление загрузкой CPU0: 0: загрузка выполняется в обычном режиме. 1: загрузка не выполняется до сброса данного бита.

### 3.2.10 Регистр SYSCTR\_NMI\_ENABLE

Регистр позволяет включать и выключать источники немаскируемого прерывания отдельно для каждого ядра.

**Таблица 3.11. Регистр SYSCTR\_NMI\_ENABLE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0	Резерв
1	CPU1_NMI_ENABLE	RW	0	Управление немаскируемым прерыванием NMI ядра CPU1: 1: прерывание разрешено. 0: прерывание запрещено.
0	CPU0_NMI_ENABLE	RW	1	Управление немаскируемым прерыванием NMI ядра CPU0: 1: прерывание разрешено. 0: прерывание запрещено.

### 3.2.11 Регистр SYSCTR\_WICCTRL

Регистр предназначен для программного управления контроллерами прерываний WIC каждого ядра процессора.

Регистр сбрасывается по сигналу nWARMRESETAON.

**Таблица 3.12. Регистр SYSCTR\_WICCTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	CPU1WICRDY	RO	0	Установка этого бита в 1 подтверждает включение CPU1 WIC.
16	CPU0WICRDY	RO	0	Установка этого бита в 1 подтверждает включение CPU0 WIC.
15:10	-	RO	0	Резерв
9	CPU1WICEN_CLR	WO	0	Запись 1 устанавливает запрос на выключение CPU1 WIC.
8	CPU0WICEN_CLR	WO	0	Запись 1 устанавливает запрос на выключение CPU0 WIC.
7:6	-	RO	0	Резерв
5	CPU1WICEN_SET	WO	0	Запись 1 устанавливает запрос на включение CPU1 WIC.
4	CPU0WICEN_SET	WO	0	Запись 1 устанавливает запрос на включение CPU0 WIC.
3:2	-	RO	0	Резерв
1	CPU1WICEN_STATUS	RO	0	Статус запроса на включение CPU1 WIC. Устанавливается в 1 после записи 1 в поле CPU1WICEN_SET. Сбрасывается в 0 после записи 1 в поле CPU1WICEN_CLR.
0	CPU0WICEN_STATUS	RO	0	Статус запроса на включение CPU0 WIC. Устанавливается в 1 после записи 1 в поле CPU0WICEN_SET. Сбрасывается в 0 после записи 1 в поле CPU0WICEN_CLR.

### 3.2.12 Регистр SYSCTR\_EWCCTRL

Регистр используется для программного управления контроллерами прерываний EWC каждого ядра процессора, предназначенными для пробуждения соответствующего ядра из режимов пониженного потребления.

Если контроллер EWC<sub>n</sub> включен, то при переходе соответствующего ядра CPU<sub>n</sub> в режим DeepSleep при включенном WIC0 все разрешенные прерывания будут фиксироваться с помощью EWC<sub>n</sub>.

Регистр сбрасывается по сигналу nWARMRESETAON.

**Таблица 3.13. Регистр SYSCTR\_EWCCTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:10	-	RO	0	Резерв
9	EWC1EN_CLR	WO	0	Запись 1 сбрасывает бит EWC1EN_STATUS.
8	EWC0EN_CLR	WO	0	Запись 1 сбрасывает бит EWC0EN_STATUS.
7:6	-	RO	0	Резерв
5	EWC1EN_SET	WO	0	Запись 1 устанавливает бит EWC1EN_STATUS.
4	EWC0EN_SET	WO	0	Запись 1 устанавливает бит EWC0EN_STATUS.
3:2	-	RO	0	Резерв
1	EWC1EN_STATUS	RO	0	Статус контроллера EWC1: 0: контроллер выключен; 1: контроллер включен.
0	EWC0EN_STATUS	RO	0	Статус контроллера EWC0: 0: контроллер выключен; 1: контроллер включен.

### 3.2.13 Регистр SYSCTR\_PDCM\_SYS\_SENSE

Регистр предназначен для настройки взаимосвязей в блоке PDCM. Каждый бит регистра определяет, блокирует ли состояние соответствующего домена выключение домена PD\_SYS или нет.

Регистр сбрасывается по сигналу nWARMRESETAON.

**Таблица 3.14. Регистр SYSCTR\_PDCM\_SYS\_SENSE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	S_PD_GNSS_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_GNSS находится в состоянии ON.
16	S_PD_GMS_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_GMS находится в состоянии ON.
15:13	-	RO	0	Резерв
12	S_PD_CRYPTON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_CRYPTON находится в состоянии ON.
11:7	-	RO	0	Резерв
6	S_PD_SRAM3_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_SRAM3 находится в состоянии ON.
5	S_PD_SRAM2_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_SRAM2 находится в состоянии ON.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
4	S_PD_SRAM1_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_SRAM1 находится в состоянии ON.
3	S_PD_SRAM0_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_SRAM0 находится в состоянии ON.
2	S_PD_CPU1_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_CPU1 находится в состоянии ON.
1	S_PD_CPU0_ON	RO	1	Домен PD_SYS всегда остается включенным, если домен PD_CPU0 находится в состоянии ON.
0	S_PD_SYS_ON	RW	1	Установка бита в 1 предотвращает выключение домена PD_SYS после его включения.

### 3.2.14 Регистры SYSCTR\_PDCM\_PD\_SRAM<N>\_SENSE

Регистр предназначен для настройки взаимосвязей в блоке PDCM. Каждый бит регистра определяет, блокирует ли состояние соответствующего домена выключение домена PD\_SRAM<N> (N = 0, 1, 2, 3) или нет.

Регистр сбрасывается по сигналу nWARMRESETAON.

**Таблица 3.15. Регистр SYSCTR\_PDCM\_PD\_SRAM<N>\_SENSE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	S_PD_GNSS_ON	RW	1	Домен PD_SRAM<N> всегда остается включенным, если домен PD_GNSS находится в состоянии ON.
16	S_PD_GMS_ON	RW	1	Домен PD_SRAM<N> всегда остается включенным, если домен PD_GMS находится в состоянии ON.
15:7	-	RO	0	Резерв
6	S_PD_SRAM3_ON	RO/RW	0	При N = 3: Доступ RW. Установка бита в 1 предотвращает выключение домена PD_SRAM3 после его включения. При N != 3: Доступ RO. Состояние PD_SRAM3 игнорируется.
5	S_PD_SRAM2_ON	RO/RW	0	При N = 2: Доступ RW. Установка бита в 1 предотвращает выключение домена PD_SRAM2 после его включения. При N != 2: Доступ RO. Состояние PD_SRAM2 игнорируется.
4	S_PD_SRAM1_ON	RO/RW	0	При N = 1: Доступ RW. Установка бита в 1 предотвращает выключение домена PD_SRAM1 после его включения. При N != 1: Доступ RO. Состояние PD_SRAM1 игнорируется.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
3	S_PD_SRAM0_ON	RO/RW	0	При N = 0: Доступ RW. Установка бита в 1 предотвращает выключение домена PD_SRAM0 после его включения. При N != 0: Доступ RO. Состояние PD_SRAM0 игнорируется.
2	S_PD_CPU1_ON	RW	0	Установка бита в 1 предотвращает выключение домена PD_SRAM<N>, если домен PD_CPU1 находится в состоянии ON.
1	S_PD_CPU0_ON	RW	0	Установка бита в 1 предотвращает выключение домена PD_SRAM<N>, если домен PD_CPU0 находится в состоянии ON.
0	S_PD_SYS_ON	RW	0	Установка бита в 1 предотвращает выключение домена PD_SRAM<N>, если домен PD_SYS находится в состоянии ON.

## 4. SPCTR/NSPCTR: БЛОКИ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ

### 4.1 Введение

Блок регистров SPCTR расположен по адресу 0x5008\_0000 и доступен только для привилегированных Secure обращений. Основные функции блока:

- Управление контроллерами безопасности PPC, MSC
- Отображение статуса и управление прерываниями PPC, MSC, MPC, BRG
- Управление функцией Non-Secure Callable областей памяти Code и SRAM

Блок регистров NSPCTR расположен по адресу 0x4008\_0000 и доступен только для привилегированных Non-secure обращений. Основное назначение блока – управление привилегированностью доступа Non-secure устройств, подключенных к контроллерам PPC.

### 4.2 Регистры SPCTR

#### 4.2.1 Перечень регистров SPCTR

Перечень регистров SPCTR приведен в следующей таблице:

**Таблица 4.1. Перечень программно-доступных регистров SPCTR**

Условное обозначение	Название	Исходное состояние	Смещение
SPCTR_SPCSECCTRL	Регистр блокировки настроек	0x0	0x000
SPCTR_BUSWAIT	Регистр блокировки ACG	0x1	0x004
SPCTR_SECRESPCFG	Регистр настройки ответа PPC при нарушении безопасности	0x0	0x010
SPCTR_NSCCFG	Регистр управления функцией Non-Secure Callable	0x0	0x014
SPCTR_MPCINTSTAT	Регистр статуса прерываний MPC	0x0	0x01C
SPCTR_PPCINTSTAT	Регистр статуса прерываний PPC	0x0	0x020
SPCTR_PPCINTCLR	Регистр сброса прерываний PPC	0x0	0x024
SPCTR_PPCINTEN	Регистр разрешения прерываний PPC	0x0	0x028
SPCTR_MSCINTSTAT	Регистр статуса прерываний MSC	0x0	0x030
SPCTR_MSCINTCLR	Регистр сброса прерываний MSC	0x0	0x034
SPCTR_MSCINTEN	Регистр разрешения прерываний MSC	0x0	0x038
SPCTR_BRGINTSTAT	Регистр статуса прерываний мостов	0x0	0x040

SPCTR_BRGINTCLR	Регистр сброса прерываний мостов	0x0	0x044
SPCTR_BRGINTEN	Регистр разрешения прерываний мостов	0x0	0x048
SPCTR_AHBPPC0_NS	Регистр настройки безопасности доступа АHBPPC0	0x0	0x060
SPCTR_APBPPC4_NS	Регистр настройки безопасности доступа APBPPC4	0x0	0x070
SPCTR_APBPPC5_NS	Регистр настройки безопасности доступа APBPPC5	0x0	0x074
SPCTR_APBPPC0_NS	Регистр настройки безопасности доступа APBPPC0	0x0	0x080
SPCTR_APBPPC1_NS	Регистр настройки безопасности доступа APBPPC1	0x0	0x084
SPCTR_APBPPC2_NS	Регистр настройки безопасности доступа APBPPC2	0x0	0x088
SPCTR_APBPPC3_NS	Регистр настройки безопасности доступа APBPPC3	0x0	0x08C
SPCTR_AHBPPC0_SP	Регистр настройки привилегированности Secure доступа АHBPPC0	0x0	0x0A0
SPCTR_APBPPC4_SP	Регистр настройки привилегированности Secure доступа APBPPC4	0x0	0x0B0
SPCTR_APBPPC5_SP	Регистр настройки привилегированности Secure доступа APBPPC5	0x0	0x0B4
SPCTR_APBPPC0_SP	Регистр настройки привилегированности Secure доступа APBPPC0	0x0	0x0C0
SPCTR_APBPPC1_SP	Регистр настройки привилегированности Secure доступа APBPPC1	0x0	0x0C4
SPCTR_APBPPC2_SP	Регистр настройки привилегированности Secure доступа APBPPC2	0x0	0x0C8
SPCTR_APBPPC3_SP	Регистр настройки привилегированности Secure доступа APBPPC3	0x0	0x0CC
SPCTR_MSC_NS	Регистр настройки безопасности доступа MSC	0x0	0x0D0

#### 4.2.2 Описание регистра SPCTR\_SPCSECCTRL

Регистр SPCTR\_SPCSECCTRL предназначен для блокировки изменения некоторых настроек, содержащихся в SPCTR.

Формат регистра SPCTR\_SPCSECCTRL приведен в следующей таблице:



**Таблица 4.2. Поля регистра SPCTR\_SPCSECCTRL**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	CFGLOCK	RW	0	Запись значения 1 блокирует изменение следующих регистров: SPCTR_NSCCFG SPCTR_*PPC*_SP SPCTR_*PPC*_NS SPCTR_MSC_NS Запись 0 игнорируется.

### 4.2.3 Описание регистра SPCTR\_BUSWAIT

Регистр позволяет заблокировать блоки ACG на master-интерфейсах контроллеров DMA, USB, SDMMC.

Формат регистра SPCTR\_BUSWAIT приведен в следующей таблице:

**Таблица 4.3. Поля регистра SPCTR\_BUSWAIT**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:17	-	RO	0	Резерв
16	ACC_WAITN_STAT	RO	0	Статус доступа через блоки ACG: 0x0: доступ запрещен; 0x1: доступ разрешен. Параметр влияет на блоки ACG, подключенные к контроллерам DMA, USB, SDMMC
15:1	-	RO	0	Резерв
0	ACC_WAITN	RW	1	Запрос на блокировку доступа через блоки ACG: 0x0: доступ запрещен; 0x1: доступ разрешен

### 4.2.4 Описание регистра SPCTR\_SECRESPCFG

Регистр предназначен для настройки реакции контроллеров PPC на нарушение безопасности.

Формат регистра SPCTR\_SECRESPCFG приведен в следующей таблице:

**Таблица 4.4. Поля регистра SPCTR\_SECRESPCFG**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	SECRESPCFG	RW	0	Реакция контроллеров PPC на нарушение безопасности: 0x0: чтение возвращает нули, запись игнорируется; 0x1: ошибка шины.

#### 4.2.5 Описание регистра SPCTR\_NSCCFG

Регистр позволяет управлять функцией Non-Secure Callable для областей памяти Code и SRAM.

Формат регистра SPCTR\_NSCCFG приведен в следующей таблице:

**Таблица 4.5. Поля регистра SPCTR\_NSCCFG**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:2	-	RO	0	Резерв
1	RAMNSC	RW	0	Управление функцией Non-Secure Callable для области SRAM (0x3000_0000 - 0x3FFF_FFFF): 0x0: Non-secure вызовы запрещены; 0x1: Non-secure вызовы разрешены
0	CODENSC	RW	0	Управление функцией Non-Secure Callable для области Code (0x1000_0000 - 0x1FFF_FFFF): 0x0: Non-secure вызовы запрещены; 0x1: Non-secure вызовы разрешены

#### 4.2.6 Описание регистра SPCTR\_MPCINTSTAT

Регистр содержит статус прерываний от всех контроллеров MPC в системе.

Формат регистра SPCTR\_MPCINTSTAT приведен в следующей таблице:

**Таблица 4.6. Поля регистра SPCTR\_MPCINTSTAT**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:20	-	RO	0	Резерв
19	BKPSRAM_MPC_STAT	RO	0	Статус прерывания контроллера BKPSRAM_MPC

Диапазон	Название	Тип доступа	Исходное состояние	Описание
18	SMC_MPC_INTSTAT	RO	0	Статус прерывания контроллера SMC_MPC
17	QSPI_MPC_INTSTAT	RO	0	Статус прерывания контроллера QSPI_MPC
16	FLASH_MPC_INTSTAT	RO	0	Статус прерывания контроллера FLASH_MPC
15:4	-	RO	0	Резерв
3	SRAM3_MPC_INTSTAT	RO	0	Статус прерывания контроллера SRAM3_MPC
2	SRAM2_MPC_INTSTAT	RO	0	Статус прерывания контроллера SRAM2_MPC
1	SRAM1_MPC_INTSTAT	RO	0	Статус прерывания контроллера SRAM1_MPC
0	SRAM0_MPC_INTSTAT	RO	0	Статус прерывания контроллера SRAM0_MPC

#### 4.2.7 Описание регистра SPCTR\_PPCINTSTAT

При нарушении безопасности на одном из контроллеров PPC генерируется комбинированное прерывание PPC. Регистры SPCTR\_PPCINTEN, SPCTR\_PPCINTSTAT, SPCTR\_PPCINTCLR предназначены управления разрешением прерывания, проверки статуса и сброса прерывания соответственно.

Формат регистра SPCTR\_PPCINTSTAT приведен в следующей таблице:

**Таблица 4.7. Поля регистра SPCTR\_PPCINTSTAT**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	АНВPPC0_INTSTAT	RO	0	Статус прерывания контроллера АНВPPC0
19:8	-	RO	0	Резерв
7	АРВPPC3_INTSTAT	RO	0	Статус прерывания контроллера АРВPPC3
6	АРВPPC2_INTSTAT	RO	0	Статус прерывания контроллера АРВPPC2
5	АРВPPC1_INTSTAT	RO	0	Статус прерывания контроллера АРВPPC1

Диапазон	Название	Тип доступа	Исходное состояние	Описание
4	APBPPC0_INTSTAT	RO	0	Статус прерывания контроллера APBPPC0
3:2	-	RO	0	Резерв
1	APBPPC5_INTSTAT	RO	0	Статус прерывания контроллера APBPPC5
0	APBPPC4_INTSTAT	RO	0	Статус прерывания контроллера APBPPC4

#### 4.2.8 Описание регистра SPCTR\_PPCINTCLR

Формат регистра SPCTR\_PPCINTCLR приведен в следующей таблице:

**Таблица 4.8. Поля регистра SPCTR\_PPCINTCLR**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	АНВPPC0_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера АНВPPC0
19:8	-	RO	0	Резерв
7	APBPPC3_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC3
6	APBPPC2_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC2
5	APBPPC1_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC1
4	APBPPC0_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC0
3:2	-	RO	0	Резерв
1	APBPPC5_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC5
0	APBPPC4_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера APBPPC4

#### 4.2.9 Описание регистра SPCTR\_PPCINTEN

Формат регистра SPCTR\_PPCINTEN приведен в следующей таблице:

**Таблица 4.9. Поля регистра SPCTR\_PPCINTEN**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	АНВPPC0_INTEN	RW	0	Запись 1 разрешает прерывание контроллера АНВPPC0
19:8	-	RO	0	Резерв
7	APBPPC3_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC3
6	APBPPC2_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC2
5	APBPPC1_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC1
4	APBPPC0_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC0
3:2	-	RO	0	Резерв
1	APBPPC5_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC5
0	APBPPC4_INTEN	RW	0	Запись 1 разрешает прерывание контроллера APBPPC4

#### 4.2.10 Описание регистра SPCTR\_MSCINTSTAT

При нарушении безопасности на одном из контроллеров MSC генерируется комбинированное прерывание MSC. Регистры SPCTR\_MSCINTEN, SPCTR\_MSCINTSTAT, SPCTR\_MSCINTCLR предназначены управления разрешением прерывания, проверки статуса и сброса прерывания соответственно.

Формат регистра SPCTR\_MSCINTSTAT приведен в следующей таблице:

**Таблица 4.10. Поля регистра SPCTR\_MSCINTSTAT**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	GMS_MSC_INTSTAT	RO	0	Статус прерывания контроллера GMS_MSC
19	USB_MSC_INTSTAT	RO	0	Статус прерывания контроллера USB_MSC
18	SDMMC_SMC_INTSTAT	RO	0	Статус прерывания контроллера SDMMC_MSC

Диапазон	Название	Тип доступа	Исходное состояние	Описание
17	DMA1_MSC_INTSTAT	RO	0	Статус прерывания контроллера DMA1_MSC
16	DMA0_MSC_INTSTAT	RO	0	Статус прерывания контроллера DMA0_MSC
15:0	-	RO	0	Резерв

#### 4.2.11 Описание регистра SPCTR\_MSCINTCLR

Формат регистра SPCTR\_MSCINTCLR приведен в следующей таблице:

**Таблица 4.11. Поля регистра SPCTR\_MSCINTCLR**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	GMS_MSC_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера GMS_MSC
19	USB_MSC_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера USB_MSC
18	SDMMC_SMC_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера SDMMC_MSC
17	DMA1_MSC_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера DMA1_MSC
16	DMA0_MSC_INTCLR	WO	0	Запись 1 сбрасывает прерывание контроллера DMA0_MSC
15:0	-	RO	0	Резерв

#### 4.2.12 Описание регистра SPCTR\_MSCINTEN

Формат регистра SPCTR\_MSCINTEN приведен в следующей таблице:

**Таблица 4.12. Поля регистра SPCTR\_MSCINTEN**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	GMS_MSC_INTEN	RW	0	Запись 1 разрешает прерывание контроллера GMS_MSC

Диапазон	Название	Тип доступа	Исходное состояние	Описание
19	USB_MSC_INTEN	RW	0	Запись 1 разрешает прерывание контроллера USB_MSC
18	SDMMC_MSC_INTEN	RW	0	Запись 1 разрешает прерывание контроллера SDMMC_MSC
17	DMA1_MSC_INTEN	RW	0	Запись 1 разрешает прерывание контроллера DMA1_MSC
16	DMA0_MSC_INTEN	RW	0	Запись 1 разрешает прерывание контроллера DMA0_MSC
15:0	-	RO	0	Резерв

### 4.2.13 Описание регистра SPCTR\_BRGINTSTAT

При возникновении ошибки в буфере моста-синхронизатора генерируется прерывание BRG. Регистры SPCTR\_BRGINTEN, SPCTR\_BRGINTSTAT, SPCTR\_BRGINTCLR предназначены управления разрешением прерывания, проверки статуса и сброса прерывания соответственно.

Формат регистра SPCTR\_BRGINTSTAT приведен в следующей таблице:

**Таблица 4.13. Поля регистра SPCTR\_BRGINTSTAT**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	BRG_CPU1SYS_INTSTAT	RO	0	Статус прерывания sync-down моста CPU1

### 4.2.14 Описание регистра SPCTR\_BRGINTCLR

Формат регистра SPCTR\_BRGINTCLR приведен в следующей таблице:

**Таблица 4.14. Поля регистра SPCTR\_BRGINTCLR**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	BRG_CPU1SYS_INTCLR	WO	0	Запись 1 сбрасывает прерывание sync-down моста CPU1

#### 4.2.15 Описание регистра SPCTR\_BRGINTEN

Формат регистра SPCTR\_BRGINTEN приведен в следующей таблице:

**Таблица 4.15. Поля регистра SPCTR\_BRGINTEN**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	BRG_CPU1SYS_INTEN	RW	0	Запись 1 разрешает прерывание sync-down моста CPU1

#### 4.2.16 Описание регистра SPCTR\_AHBPCC0\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру АНВРРС0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только Secure доступ;

0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_AHBPCC0\_NS приведен в следующей таблице:

**Таблица 4.16. Поля регистра SPCTR\_AHBPCC0\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:7	-	RO	0	Резерв
6	DMA1_NS	RW	0	Настройка доступа для DMA1.
5	DMA0_NS	RW	0	Настройка доступа для DMA0.
4	SDMMC_NS	RW	0	Настройка доступа для SDMMC.
3	SMC_NS	RW	0	Настройка доступа для SMC.
2	GNSS_NS	RW	0	Настройка доступа для GNSS.
1	QSPI_NS	RW	0	Настройка доступа для QSPI.
0	USB_NS	RW	0	Настройка доступа для USB.

#### 4.2.17 Описание регистра SPCTR\_APBPPC4\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру АРВРРС4. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только Secure доступ;

0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC4\_NS приведен в следующей таблице:



**Таблица 4.17. Поля регистра SPCTR\_APBPPC4\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:5	-	RO	0	Резерв
4	MHU1_NS	RW	0	Настройка доступа для MHU1.
3	MHU0_NS	RW	0	Настройка доступа для MHU0.
2	DTIM_NS	RW	0	Настройка доступа для DTIM.
1	TIM1_NS	RW	0	Настройка доступа для TIM1.
0	TIM0_NS	RW	0	Настройка доступа для TIM0.

#### 4.2.18 Описание регистра SPCTR\_APBPPC5\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру APBPPC5. Каждое поле соответствует одному из устройств и кодируется следующим образом:

- 0x0: разрешен только Secure доступ;
- 0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC5\_NS приведен в следующей таблице:

**Таблица 4.18. Поля регистра SPCTR\_APBPPC5\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	LPTIM_NS	RW	0	Настройка доступа для LPTIM.

#### 4.2.19 Описание регистра SPCTR\_APBPPC0\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру APBPPC0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

- 0x0: разрешен только Secure доступ;
- 0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC0\_NS приведен в следующей таблице:

**Таблица 4.19. Поля регистра SPCTR\_APBPPC0\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:10	-	RO	0	Резерв
9	I2S_NS	RW	0	Настройка доступа для I2S.

Диапазон	Название	Тип доступа	Исходное состояние	Описание
8	I2C1_NS	RW	0	Настройка доступа для I2C1.
7	I2C0_NS	RW	0	Настройка доступа для I2C0.
6	SPI2_NS	RW	0	Настройка доступа для SPI2.
5	SPI1_NS	RW	0	Настройка доступа для SPI1.
4	SPI0_NS	RW	0	Настройка доступа для SPI0.
3	UART3_NS	RW	0	Настройка доступа для UART3.
2	UART2_NS	RW	0	Настройка доступа для UART2.
1	UART1_NS	RW	0	Настройка доступа для UART1.
0	UART0_NS	RW	0	Настройка доступа для UART0.

#### 4.2.20 Описание регистра SPCTR\_APBPPC1\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру APBPPC1. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только Secure доступ;

0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC1\_NS приведен в следующей таблице:

**Таблица 4.20. Поля регистра SPCTR\_APBPPC1\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	VTU1_NS	RW	0	Настройка доступа для VTU1.
2	VTU0_NS	RW	0	Настройка доступа для VTU0.
1	PWM_NS	RW	0	Настройка доступа для PWM.
0	CAN_NS	RW	0	Настройка доступа для CAN.

#### 4.2.21 Описание регистра SPCTR\_APBPPC2\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру APBPPC2. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только Secure доступ;

0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC2\_NS приведен в следующей таблице:

**Таблица 4.21. Поля регистра SPCTR\_APBPPC2\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	JTM_NS	RW	0	Настройка доступа для JTM.
2	RWC_NS	RW	0	Настройка доступа для RWC.
1:0	-	RO	0	Резерв

#### 4.2.22 Описание регистра SPCTR\_APBPPC3\_NS

Регистр позволяет управлять уровнем безопасности устройств, подключенных к контроллеру APBPPC3. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только Secure доступ;

0x1: разрешен только Non-secure доступ.

Формат регистра SPCTR\_APBPPC3\_NS приведен в следующей таблице:

**Таблица 4.22. Поля регистра SPCTR\_APBPPC3\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:3	-	RO	0	Резерв
2	PWRCTR_NS	RW	0	Настройка доступа для PWRCTR.
1	CLKCTR_NS	RW	0	Настройка доступа для CLKCTR.
0	IOCTR_NS	RW	0	Настройка доступа для IOCTR.

#### 4.2.23 Описание регистра SPCTR\_AHBPPC0\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру AHBPPC0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_AHBPPC0\_SP приведен в следующей таблице:

**Таблица 4.23. Поля регистра SPCTR\_AHBPPC0\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:7	-	RO	0	Резерв

Диапазон	Название	Тип доступа	Исходное состояние	Описание
6	DMA1_SP	RW	0	Настройка доступа для DMA1.
5	DMA0_SP	RW	0	Настройка доступа для DMA0.
4	SDMMC_SP	RW	0	Настройка доступа для SDMMC.
3	SMC_SP	RW	0	Настройка доступа для SMC.
2	GNSS_SP	RW	0	Настройка доступа для GNSS.
1	QSPI_SP	RW	0	Настройка доступа для QSPI.
0	USB_SP	RW	0	Настройка доступа для USB.

#### 4.2.24 Описание регистра SPCTR\_APBPPC4\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC4. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC4\_SP приведен в следующей таблице:

**Таблица 4.24. Поля регистра SPCTR\_APBPPC4\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:5	-	RO	0	Резерв
4	MHU1_SP	RW	0	Настройка доступа для MHU1.
3	MHU0_SP	RW	0	Настройка доступа для MHU0.
2	DTIM_SP	RW	0	Настройка доступа для DTIM.
1	TIM1_SP	RW	0	Настройка доступа для TIM1.
0	TIM0_SP	RW	0	Настройка доступа для TIM0.

#### 4.2.25 Описание регистра SPCTR\_APBPPC5\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC5. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC5\_SP приведен в следующей таблице:

**Таблица 4.25. Поля регистра SPCTR\_APBPPC5\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	LPTIM_SP	RW	0	Настройка доступа для LPTIM.

#### 4.2.26 Описание регистра SPCTR\_APBPPC0\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC0\_SP приведен в следующей таблице:

**Таблица 4.26. Поля регистра SPCTR\_APBPPC0\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:14	-	RO	0	Резерв
13	GPIO3_SP	RW	0	Настройка доступа для GPIO3.
12	GPIO2_SP	RW	0	Настройка доступа для GPIO2.
11	GPIO1_SP	RW	0	Настройка доступа для GPIO1.
10	GPIO0_SP	RW	0	Настройка доступа для GPIO0.
9	I2S_SP	RW	0	Настройка доступа для I2S.
8	I2C1_SP	RW	0	Настройка доступа для I2C1.
7	I2C0_SP	RW	0	Настройка доступа для I2C0.
6	SPI2_SP	RW	0	Настройка доступа для SPI2.
5	SPI1_SP	RW	0	Настройка доступа для SPI1.
4	SPI0_SP	RW	0	Настройка доступа для SPI0.
3	UART3_SP	RW	0	Настройка доступа для UART3.
2	UART2_SP	RW	0	Настройка доступа для UART2.
1	UART1_SP	RW	0	Настройка доступа для UART1.
0	UART0_SP	RW	0	Настройка доступа для UART0.

#### 4.2.27 Описание регистра SPCTR\_APBPPC1\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC1. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC1\_SP приведен в следующей таблице:

**Таблица 4.27. Поля регистра SPCTR\_APBPPC1\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	VTU1_SP	RW	0	Настройка доступа для VTU1.
2	VTU0_SP	RW	0	Настройка доступа для VTU0.
1	PWM_SP	RW	0	Настройка доступа для PWM.
0	CAN_SP	RW	0	Настройка доступа для CAN.

#### 4.2.28 Описание регистра SPCTR\_APBPPC2\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC2. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC2\_SP приведен в следующей таблице:

**Таблица 4.28. Поля регистра SPCTR\_APBPPC2\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	JTM_SP	RW	0	Настройка доступа для JTM.
2	RWC_SP	RW	0	Настройка доступа для RWC.
1:0	-	RO	0	Резерв

#### 4.2.29 Описание регистра SPCTR\_APBPPC3\_SP

Регистр задает требуемый уровень привилегированности Secure обращений к устройствам, подключенным к контроллеру APBPPC3. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Secure доступ

Формат регистра SPCTR\_APBPPC3\_SP приведен в следующей таблице:

**Таблица 4.29. Поля регистра SPCTR\_APBPPC3\_SP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:3	-	RO	0	Резерв
2	PWRCTR_SP	RW	0	Настройка доступа для PWRCTR.
1	CLKCTR_SP	RW	0	Настройка доступа для CLKCTR.
0	IOCTR_SP	RW	0	Настройка доступа для IOCTR.

#### 4.2.30 Описание регистра SPCTR\_MSC\_NS

Регистр позволяет управлять уровнем безопасности контроллеров MSC, подключенных к устройствам DMA, SDMMC, USB.

Формат регистра SPCTR\_MSC\_NS приведен в следующей таблице:

**Таблица 4.30. Поля регистра SPCTR\_MSC\_NS**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20	GMS_MSC_NS	RW	0	Уровень безопасности контроллера GMS_MSC: 0x0: Secure; 0x1: Non-secure
19	USB_MSC_NS	RW	0	Уровень безопасности контроллера USB_MSC: 0x0: Secure; 0x1: Non-secure
18	SDMMC_SMC_NS	RW	0	Уровень безопасности контроллера SDMMC_MSC: 0x0: Secure; 0x1: Non-secure
17	DMA1_MSC_NS	RW	0	Уровень безопасности контроллера DMA1_MSC: 0x0: Secure; 0x1: Non-secure
16	DMA0_MSC_NS	RW	0	Уровень безопасности контроллера DMA0_MSC: 0x0: Secure; 0x1: Non-secure
15:0	-	RO	0	Резерв

## 4.3 Регистры NSPCTR

### 4.3.1 Перечень регистров NSPCTR

Перечень регистров NSPCTR приведен в следующей таблице:

**Таблица 4.31. Перечень программно-доступных регистров NSPCTR**

Условное обозначение	Название	Исходное состояние	Смещение
NSPCTR_AHBPCC0_NSP	Регистр настройки привилегированности Non-secure доступа AHBPCC0	0x0	0x0A0
NSPCTR_APBPPC4_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC4	0x0	0x0B0
NSPCTR_APBPPC5_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC5	0x0	0x0B4
NSPCTR_APBPPC0_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC0	0x0	0x0C0
NSPCTR_APBPPC1_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC1	0x0	0x0C4
NSPCTR_APBPPC2_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC2	0x0	0x0C8
NSPCTR_APBPPC3_NSP	Регистр настройки привилегированности Non-secure-доступа APBPPC3	0x0	0x0CC

### 4.3.2 Описание регистра NSPCTR\_AHBPCC0\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру AHBPCC0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_AHBPCC0\_NSP приведен в следующей таблице:

**Таблица 4.32. Поля регистра NSPCTR\_AHBPCC0\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:7	-	RO	0	Резерв
6	DMA1_NSP	RW	0	Настройка доступа для DMA1.
5	DMA0_NSP	RW	0	Настройка доступа для DMA0.
4	SDMMC_NSP	RW	0	Настройка доступа для SDMMC.
3	SMC_NSP	RW	0	Настройка доступа для SMC.



Диапазон	Название	Тип доступа	Исходное состояние	Описание
2	GNSS_NSP	RW	0	Настройка доступа для GNSS.
1	QSPI_NSP	RW	0	Настройка доступа для QSPI.
0	USB_NSP	RW	0	Настройка доступа для USB.

### 4.3.3 Описание регистра NSPCTR\_APBPPC4\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC4. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный non-secure доступ.

Формат регистра NSPCTR\_APBPPC4\_NSP приведен в следующей таблице:

**Таблица 4.33. Поля регистра NSPCTR\_APBPPC4\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:5	-	RO	0	Резерв
4	MHU1_NSP	RW	0	Настройка доступа для MHU1.
3	MHU0_NSP	RW	0	Настройка доступа для MHU0.
2	DTIM_NSP	RW	0	Настройка доступа для DTIM.
1	TIM1_NSP	RW	0	Настройка доступа для TIM1.
0	TIM0_NSP	RW	0	Настройка доступа для TIM0.

### 4.3.4 Описание регистра NSPCTR\_APBPPC5\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC5. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_APBPPC5\_NSP приведен в следующей таблице:

**Таблица 4.34. Поля регистра NSPCTR\_APBPPC5\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв

Диапазон	Название	Тип доступа	Исходное состояние	Описание
0	LPTIM_NSP	RW	0	Настройка доступа для LPTIM.

### 4.3.5 Описание регистра NSPCTR\_APBPPC0\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC0. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_APBPPC0\_NSP приведен в следующей таблице:

**Таблица 4.35. Поля регистра NSPCTR\_APBPPC0\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:14	-	RO	0	Резерв
13	GPIO3_NSP	RW	0	Настройка доступа для GPIO3.
12	GPIO2_NSP	RW	0	Настройка доступа для GPIO2.
11	GPIO1_NSP	RW	0	Настройка доступа для GPIO1.
10	GPIO0_NSP	RW	0	Настройка доступа для GPIO0.
9	I2S_NSP	RW	0	Настройка доступа для I2S.
8	I2C1_NSP	RW	0	Настройка доступа для I2C1.
7	I2C0_NSP	RW	0	Настройка доступа для I2C0.
6	SPI2_NSP	RW	0	Настройка доступа для SPI2.
5	SPI1_NSP	RW	0	Настройка доступа для SPI1.
4	SPI0_NSP	RW	0	Настройка доступа для SPI0.
3	UART3_NSP	RW	0	Настройка доступа для UART3.
2	UART2_NSP	RW	0	Настройка доступа для UART2.
1	UART1_NSP	RW	0	Настройка доступа для UART1.
0	UART0_NSP	RW	0	Настройка доступа для UART0.

### 4.3.6 Описание регистра NSPCTR\_APBPPC1\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC1. Каждое поле соответствует одному

из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_APBPPC1\_NSP приведен в следующей таблице:

**Таблица 4.36. Поля регистра NSPCTR\_APBPPC1\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	VTU1_NSP	RW	0	Настройка доступа для VTU1.
2	VTU0_NSP	RW	0	Настройка доступа для VTU0.
1	PWM_NSP	RW	0	Настройка доступа для PWM.
0	CAN_NSP	RW	0	Настройка доступа для CAN.

#### 4.3.7 Описание регистра NSPCTR\_APBPPC2\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC2. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_APBPPC2\_NSP приведен в следующей таблице:

**Таблица 4.37. Поля регистра NSPCTR\_APBPPC2\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:4	-	RO	0	Резерв
3	JTM_NSP	RW	0	Настройка доступа для JTM.
2	RWC_NSP	RW	0	Настройка доступа для RWC.
1:0	-	RO	0	Резерв

#### 4.3.8 Описание регистра NSPCTR\_APBPPC3\_NSP

Регистр задает требуемый уровень привилегированности Non-secure обращений к устройствам, подключенным к контроллеру APBPPC3. Каждое поле соответствует одному из устройств и кодируется следующим образом:

0x0: разрешен только привилегированный Non-secure доступ;

0x1: разрешен как привилегированный, так и непривилегированный Non-secure доступ.

Формат регистра NSPCTR\_APBPPC3\_NSP приведен в следующей таблице:

**Таблица 4.38. Поля регистра NSPCTR\_APBPPC3\_NSP**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:3	-	RO	0	Резерв
2	PWRCTR_NSP	RW	0	Настройка доступа для PWRCTR.
1	CLKCTR_NSP	RW	0	Настройка доступа для CLKCTR.
0	IOCTR_NSP	RW	0	Настройка доступа для IOCTR.

## 5. ICACHE: КЭШ ИНСТРУКЦИЙ

### 5.1 Введение

Каждая подсистема CPU содержит кэш инструкций, подключенный к интерфейсу C-АНВ ядра Cortex-M33. Кэширование выполняется только для запросов чтения инструкций и данных из диапазона адресов 0x0000\_0000 – 0x1FFF\_FFFF.

Основные особенности кэша:

- двухпутевой множественно-ассоциативный кэш.
- размер кэш-строки – 16 байт.
- размер памяти кэш – 16 Кбайт.
- конфигурационный интерфейс локальный для каждого процессора.
- поддерживает некэшируемые транзакции.

Каждый кэш имеет конфигурационный интерфейс, доступный только для подключенного к нему процессора и расположенный по адресу 0x5001\_0000. Этот конфигурационный интерфейс не доступен для других master-устройств в системе.

Кэш поддерживает разделение содержимого на Secure и Non-secure данные путем сохранения атрибута безопасности кэшируемых данных. Перед изменением конфигурации SAU необходимо отключить кэш и выполнить инвалидацию.

## 5.2 Описание функционирования

В данном разделе приведены рекомендации по программированию кэша L1 процессора. Данные рекомендации не являются исчерпывающим руководством по разработке программного драйвера.

### 5.2.1 Инициализация

После включения питания и сброса микросхемы кэш запускается в отключенном состоянии и автоматически выполняет процедуру инвалидации. Все приходящие в это время транзакции пропускаются без кэширования.

При включении кэша путем установки бита CACHEEN во время выполнения инвалидации фактическое включение произойдет только после завершения инвалидации.

В конце процедуры инвалидации генерируется прерывание IC\_STATUS.

### 5.2.2 Отключение

Отключение кэш возможно в любое время путем очистки бита CACHEEN.

Фактическое отключение кэш происходит после завершения всех активных транзакций. Для определения момента фактического выключения программа может опрашивать бит CDC\_STATUS либо активировать и дожидаться прерывания CDC.

### 5.2.3 Инвалидация

Инвалидация кэш выполняется с помощью установки бита FINV.

Во время выполнения инвалидации все запросы, поступающие к кэш, обрабатываются как некэшируемые. В конце процедуры инвалидации генерируется прерывание IC\_STATUS.

### 5.2.4 Когерентность

Кэш инструкций не обеспечивает когерентность внешней по отношению к кэш памяти и соответствующей строки внутри кэш.

Для изменения содержимого внешней кэшируемой памяти необходимо:

1. Отключить кэш инструкций и дождаться установки бита CDC\_STATUS или соответствующего прерывания.
2. Выполнить инвалидацию кэш и дождаться установки соответствующего бита статуса или прерывания.
3. Изменить содержимое памяти.

4. Включить кэш и дождаться установки соответствующего бита статуса или прерывания.

При изменении конфигурации SAU или MPC необходимо выполнить аналогичные шаги для предотвращения кэширования одной и той же области памяти с разными атрибутами безопасности.

## 5.3 Регистры

### 5.3.1 Перечень регистров ICACHE

Таблица 5.1. Регистры ICACHE

Условное Обозначение	Описание	Исходное состояние	Смещение
ICACHE_HWPARAMS	Регистр аппаратных параметров блока	0x0000_0000	0x000
ICACHE_CTRL	Регистр управления	0x0000_0004	0x004
-	Резерв	0x0000_0000	0x008 – 0x0FC
ICACHE_IRQSTAT	Регистр статуса прерываний	0x0000_0000	0x100
ICACHE_IRQCLR	Регистр очистки прерываний	0x0000_0000	0x104
ICACHE_IRQEN	Регистр разрешения прерываний	0x0000_0000	0x108
ICACHE_DBGFILLERR	Регистр	0x0000_0000	0x10C
-	Резерв	0x0000_0000	0x200 – 0x2FC
ICACHE_SH	Регистр статистики попаданий в кэш	0x0000_0000	0x300
ICACHE_SM	Регистр статистики промахов в кэш	0x0000_0000	0x304
ICACHE_SUC	Регистр статистики некешируемых запросов	0x0000_0000	0x308
-	Резерв	0x0000_0000	0x30C – 0xFCC
ICACHE_PIDR4	Product ID 4.	0x0000_0004	0xFD0
ICACHE_PIDR5	Резерв.	0x0000_0000	0xFD4
ICACHE_PIDR6	Резерв.	0x0000_0000	0xFD8
ICACHE_PIDR7	Резерв.	0x0000_0000	0xFDC
ICACHE_PIDR0	Product ID 0.	0x0000_0057	0xFE0
ICACHE_PIDR1	Product ID 1.	0x0000_00B8	0xFE4
ICACHE_PIDR2	Product ID 2.	0x0000_001B	0xFE8
ICACHE_PIDR3	Product ID 3.	0x0000_0000	0xFEC
ICACHE_CIDR0	Component ID 0.	0x0000_000D	0xFF0
ICACHE_CIDR1	Component ID 1.	0x0000_00F0	0xFF4
ICACHE_CIDR2	Component ID 2.	0x0000_0005	0xFF8
ICACHE_CIDR3	Component ID 3.	0x0000_00B1	0xFFC

### 5.3.2 Регистр аппаратных параметров ICACHE\_HWPARAMS

Регистр ICACHE\_HWPARAMS доступен только для чтения и описывает аппаратную конфигурацию кэша инструкций.

Таблица 5.2. Регистр аппаратных параметров ICACHE\_HWPARAMS

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	COFFSET	RO	0x0	Смещение адреса кешируемой области
15:12	COFFSIZE	RO	0x3	Размер кешируемой области памяти: 0x3: 512 Мбайт
11:7	-	RO	0	Резерв
6	INVMAT	RO	0x0	Инвализация строки кэша при записи выключена
5	DMA	RO	0x0	Блок DMA отсутствует
4	STATS	RO	0x1	Наличие функции сбора статистики
3:0	CSIZE	RO	0xE	Размер кэша: 0xE: 16 Кбайт



### 5.3.3 Регистр управления ICACHE\_CTRL

Регистр ICACHE\_CTRL служит для управления кэшем инструкций, в том числе включение или выключение кэша, запуск инвалидации, настройка кэшируемости транзакций на основе сигнала **HHINT[2]** процессора.

Регистр поддерживает запись только 32-х разрядными словами. Все остальные транзакции записи игнорируются.

**Таблица 5.3. Регистр управления ICACHE\_CTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	-	RO	0x0	Резерв
6	POINV_SMP	RO	0x1	Инвалидация после включения питания активна
5	HALLOC	RW	0x0	Управление кэшированием инструкций обработчиков прерываний: 0x0: все инструкции обработчиков прерываний не кэшируются при промахе. Если запрашиваемые данные уже есть в кэше, то запрос обрабатывается как кэшируемый. HALLOC = 0 делает запросы к коду обработчиков прерываний более детерминированными. Также это предотвращает снижение эффективности кэша при большом количестве прерываний. 0x1: запросы кода обработчиков прерываний обрабатываются как любые другие запросы инструкций
4	STATC	WO	0x0	Запись значения 1 сбрасывает все счетчики статистики
3	STATEN	RW	0x0	Включение статистики: 0x0: счетчики статистики выключены; 0x1: счетчики статистики включены.
2	FINV	WO	0x0	Запись значения 1 запускает процесс инвалидации всего кэша.
1	-	RO	0x0	Резерв
0	CACHEEN	RW	0x0	Включение/выключение кэша: 0x0 – кэш выключен, все транзакции проходят насквозь; 0x1 – кэш включен

### 5.3.4 Регистры прерываний ICACHE\_IRQSTAT, ICACHE\_IRQCLR, ICACHE\_IRQEN

Регистры прерываний позволяют программе определять причину прерывания от кэша инструкций, а также сбрасывать, включать и отключать прерывания:

- ICACHE\_IRQSTAT доступен только для чтения (RO). Регистр содержит статус всех прерываний до наложения маски.
- ICACHE\_IRQCLR доступен только для записи (WO). Регистр позволяет сбрасывать статус активных прерываний путем записи значения '1' в соответствующий бит.

- ICACHE\_IRQEN доступен для чтения и записи (RW). Регистр позволяет включать/выключать влияние статуса прерываний на сигнал прерывания кэша.

Все регистры прерываний имеют одинаковые поля, показанные в следующей таблице.

**Таблица 5.4. Регистры прерываний ICACHE\_IRQSTAT, ICACHE\_IRQCLR, ICACHE\_IRQEN**

Номер бита	Название	Исходное состояние	Назначение
31:6	-	0x0	Резерв
5	SS	0x0	Statistics Saturated IRQ. Обозначает насыщение внутренних счетчиков статистики.
4	SV	0x0	Security Violation IRQ. Прерывание по нарушению безопасности.
3	CFE	0x0	Cache Fill Error IRQ. Обозначает, что во время заполнения строки кэша произошла ошибка шины.
2	CEC	0x0	Cache Enable Complete IRQ. Обозначает, что запрос включения кэша выполнен.
1	CDC	0x0	Cache Disable Complete IRQ. Обозначает, что запрос выключения кэша выполнен.
0	IC	0x0	Invalidate Complete IRQ. Прерывание обозначает завершение процесса инвалидации.

**Примечание:** При включении прерываний возможно получение прерывания CDC, если соответствующее событие произошло перед записью регистра ICACHE\_IRQEN.

### 5.3.5 Регистр отладки ошибки заполнения ICACHE\_DBGFILLERR

Регистр ICACHE\_DBGFILLERR позволяет программе получить адрес, вызвавший ошибку шины при последнем заполнении строки.

**Таблица 5.5. Регистр управления ICACHE\_DBGFILLERR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	ERRADDR	RO	0x0	Адрес последней ошибки заполнения.

**Примечание:** Адрес может измениться между возникновением прерывания и чтением регистра процессором. Следовательно, этот адрес может быть использован только как вспомогательный при отладке. Два младших бита регистра всегда равны нулю.

### 5.3.6 Регистр статистики попаданий ICACHE\_SH

Регистр содержит счетчик транзакций чтения, вызвавших попадание в кэш.

**Таблица 5.6. Регистр статистики попаданий ICACHE\_SH**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	CSH	RO	0x0	Счетчик попаданий в кэш.

### 5.3.7 Регистр статистики промахов ICACHE\_SM

Регистр содержит счетчик транзакций чтения, вызвавших промах в кэш.

**Таблица 5.7. Регистр статистики промахов ICACHE\_SM**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	CSM	RO	0x0	Счетчик промахов в кэш.

### 5.3.8 Регистр статистики некэшируемых запросов ICACHE\_SUC

Регистр содержит счетчик некэшируемых транзакций чтения.

**Таблица 5.8. Регистр статистики промахов ICACHE\_SUC**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	CSUC	RO	0x0	Счетчик некэшируемых транзакций.

## 6. FCACHE: КЭШ FLASH-ПАМЯТИ

### 6.1 Введение

Кэш позволяет уменьшить количество запросов чтения из Flash-памяти, тем самым уменьшая потребление. Также кэш уменьшает общую латентность загрузки данных.

Блок обладает следующими характеристиками:

- двухпутевой множественно-ассоциативный кэш;
- размер памяти кэш – 8 Кбайт;
- атоматическая инвалидация при включении либо ручная инвалидация;
- функция предзагрузки (отключаемая);
- счетчики статистики попаданий и промахов в кэш;
- 32-разрядный АНВ slave-интерфейс для доступа master-устройств;
- 32-разрядный APB интерфейс для доступа к регистрам;
- 128-разрядный АНВ master-интерфейс к контроллеру Flash памяти.

### 6.2 Описание функционирования

#### 6.2.1 Ограничения

#### 6.2.2 Кэш выключен

Когда FCACHE выключен, все запросы чтения и записи по АНВ slave-интерфейсу проходят насквозь без изменений и без дополнительных тактов задержки в АНВ master-интерфейс.

#### 6.2.3 Кэш включен

Когда FCACHE включен, возможны следующие варианты работы:

##### 1. Bypass

Когда FCACHE включен, он работает как кэш инструкций, т.е. все запросы записи, запросы от отладчика и некэшируемые запросы чтения проходят насквозь к контроллеру Flash памяти. При этом добавляется один такт задержки. Также задержка может быть увеличена, если в данный момент выполняется операция предзагрузки.

##### 2. Cache hit (попадание в кэш)

Если происходит попадание в кэш (hit), т.е. данные для кэшируемой транзакции чтения найдены в памяти FCACHE, то транзакция к Flash памяти на master-интерфейсе не генерируется. Запрошенные данные возвращаются на АНВ slave-интерфейсе без дополнительных тактов ожидания.

### 3. Linefill (заполнение строки кэш)

Промех в кэш происходит, если запрашиваемые данные не найдены в памяти FCACHE. В этом случае на АНВ master-интерфейсе формируется транзакция чтения для заполнения строки кэш. FCACHE сохраняет полученные от Flash памяти данные в памяти кэш, а также отправляет их в ответ по АНВ slave-интерфейсу. На передачу требуется один такт задержки. Также задержка может быть увеличена, если в данный момент выполняется операция предзагрузки.

### 4. Prefetch (предзагрузка)

Функция предзагрузки включается битом CCR.SET\_PREFETCH. В этом случае после выполнения операции linefill выполняется предзагрузка при следующих условиях:

- операция linefill была вызвана транзакцией чтения инструкций (HPROT[0] = 0);
- данные из linefill-буфера уже записаны в SRAM память кэш;
- АНВ master-интерфейс не занят.

В качестве адреса предзагрузки используется адрес linefill операции + 0x10.

**Примечание.** Влияние предзагрузки на производительность зависит от приложения, также может возрасти потребление Flash памяти. Использование предзагрузки помогает, если код инструкций еще не был кэширован. Во многих случаях производительность системы может ухудшиться, поскольку начавшаяся операция предзагрузки не может быть отменена. Например, если происходит два последовательных промаха в кэш, то сначала выполняется prefetch по первому промаху, и лишь после этого linefill по второму промаху.

Функция предзагрузки может быть включена, только когда кэш находится в выключенном состоянии.

Если транзакция чтения на АНВ master-интерфейсе завершается с ошибкой (HRESP=1), то FCACHE не обновляет linefill-буфер и не записывает данные в SRAM. Если чтение было инициировано на АНВ slave-интерфейсе, то FCACHE возвращает ошибку инициатору. Если же чтение было вызвано предзагрузкой, то ошибка игнорируется.

## 6.2.4 Включение/выключение кэша

Если содержимое SRAM памяти FCACHE невалидно, например, после включения питания, то для включения кэша необходимо:

- убедиться, что кэш выключен (SR.CS = 0);
- убедиться, что включена автоматическая инвалидация (CCR.SET\_MAN\_INV = 0);

- установить бит CCR.EN.

Если же на момент включения кэша содержимое SRAM памяти валидно, то для избежания инвалидации перед установкой бита CCR.EN нужно установить бит CCR.SET\_MAN\_INV.

Для выключения кэша достаточно сбросить бит CCR.EN.

Включение/выключение кэша происходит на границах АHB пакетов. Если бит CCR.EN установлен/сброшен во время передачи АHB пакета, то включение/выключение кэша происходит только после завершения передачи пакета.

### 6.2.5 Инвалидация кэша

Ручная инвалидация кэша запускается установкой бита CCR.INV\_REQ. При этом бит CCR.SET\_MAN\_INV должен быть установлен. Бит CCR.INV\_REQ не может быть сброшен программно, он сбрасывается автоматически после завершения инвалидации либо при возникновении ошибки MAN\_INV\_ERR.

Ручная инвалидация должна выполняться только при выключенном кэше. В противном случае устанавливается сигнал прерывания MAN\_INV\_ERR, если это разрешено соответствующей маской, и кэш автоматически выключается.

Если же FCACHE включается во время выполнения ручной инвалидации, то фактическое включение задерживается до момента завершения инвалидации.

### 6.2.6 Счетчики статистики

Каждое событие попадания или промаха в кэш инкрементирует соответствующий 32-разрядный счетчик. Счетчики останавливаются при достижении значения 0xFFFF\_FFFF. Значение счетчиков доступно в регистрах CSHR и CSMR. Счетчики могут быть сброшены записью в соответствующий регистр.

## 6.3 Регистры

Доступ к регистрам производится через 32-разрядный APB интерфейс. Поддерживаются только 32-разрядные привилегированные (PROT[0]=1) операции записи.

Непривилегированный доступ на запись игнорируется. Чтение возможно как привилегированными, так и непривилегированными транзакциями.

### 6.3.1 Перечень регистров FCACHE

Перечень регистров FCACHE приведен в следующей таблице.

**Таблица 6.1 Регистры FCACHE**

Условное обозначение	Описание	Исходное состояние	Смещение
FCACHE_CCR	Регистр конфигурации и управления	0x00000040	0x00000000
FCACHE_SR	Регистр статуса	0x00000000	0x00000004
FCACHE_IRQMASK	Регистр маски запросов прерывания	0x00000000	0x00000008
FCACHE_IRQSTAT	Регистр статуса запросов прерывания	0x00000000	0x0000000C
FCACHE_HWPARAMS	Регистр аппаратных параметров	0x00003996	0x00000010
FCACHE_CSHR	Регистр статистики кэш-попаданий	0x00000000	0x00000014
FCACHE_CSMR	Регистр статистики кэш-промахов	0x00000000	0x00000018

### 6.3.2 Регистр конфигурации и управления, FCACHE\_CCR

Формат регистра FCACHE\_CCR приведен в следующей таблице.

**Таблица 6.2 Поля регистра FCACHE\_CCR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	-	RO	0x0	Резерв
6	STATISTIC_EN	RW	0x1	Включить сбор статистики: 0: счетчики статистики выключены; 1: счетчики статистики работают.
5	SET_PREFETCH	RW	0x0	Функция предзагрузки кэша: 0: предзагрузка отключена; 1: предзагрузка включена. Бит может быть установлен только при выключенном кэше.
4	SET_MAN_INV	RW	0x0	Инвалидация кэша: 0: автоматическая инвалидация при включении кэша; 1: режим ручной инвалидации кэша. Бит может быть установлен только при выключенном кэше.
3:2	-	RW	0x0	Резерв. Не изменять.
1	INV_REQ	RW	0x0	Запрос на инвалидацию в ручном режиме. Работает только тогда, когда бит SET_MAN_INV установлен. Автоматически очищается по завершении инвалидации или при ошибке инвалидации. Бит не может быть сброшен программно. Запрос на инвалидацию может быть установлен только в том случае, если кэш отключен, в противном случае это приводит к появлению ошибки MAN_INV_ERR.
0	EN	RW	0x0	Включение кэша: 0: кэш выключен; 1: кэш включен.

### 6.3.3 Регистр статуса, FCACHE\_SR

Формат регистра FCACHE\_SR приведен в следующей таблице.

**Таблица 6.3 Поля регистра FCACHE\_SR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0x0	Резерв
2	INV_STAT	RO	0x0	Статус инвалидации. Значение '1' означает, что кэш в процессе инвалидации.
1:0	CS	RO	0x0	Состояние кэша: 0: отключен; 1: в процессе включения; 2: включен; 3: в процессе отключения.

### 6.3.4 Регистр маски запросов прерывания, FCACHE\_IRQMASK

Формат регистра FCACHE\_IRQMASK приведен в следующей таблице.

**Таблица 6.4 Поля регистра FCACHE\_IRQMASK**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0x0	Резерв
1	MAN_INV_ERR	RW	0x0	Маскирование запроса прерывания при ошибке ручной инвалидации (бит IRQSTAT.MAN_INV_ERR установлен). 0: прерывание включено; 1: прерывание замаскировано.
0	-	RO	0x0	Резерв

### 6.3.5 Регистр статуса запросов прерывания, FCACHE\_IRQSTAT

Регистр статуса запросов прерывания.

Формат регистра FCACHE\_IRQSTAT приведен в следующей таблице.

**Таблица 6.5 Поля регистра FCACHE\_IRQSTAT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0x0	Резерв
1	MAN_INV_ERR	RW	0x0	Статус ошибки при ручной инвалидации. Устанавливается, когда есть запрос на инвалидацию в ручном режиме, но кэш не выключен. Бит устанавливается независимо от значения регистра IRQMASK. Запись 1 сбрасывает данный бит.
0	-	RO	0x0	Резерв



### 6.3.6 Регистр аппаратных параметров, FCACHE\_HWPARAMS

Регистр аппаратных параметров содержит значения параметров, заданные при реализации блока.

Формат регистра FCACHE\_HWPARAMS приведен в следующей таблице.

**Таблица 6.6 Поля регистра FCACHE\_HWPARAMS**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:14	-	RO	0x0	Резерв
13	GEN_STAT_LOGIC	RO	0x1	Значение параметра GEN_STAT_LOGIC.
12	RESET_ALL_REGS	RO	0x1	Значение параметра RESET_ALL_REGS.
11:10	CACHE_WAY	RO	0x2	Количество ассоциативных путей.
9:5	CW	RO	0xC	Размер памяти одного ассоциативного пути: 8: 256 В; 9: 512 В; 11: 2 КВ; 12: 4 КВ.
4:0	AW	RO	0x16	Размер адресуемого по интерфейсу АНВ пространства: 10: 64 КВ; 11: 128 КВ; 12: 256 КВ; 13: 512 КВ; 14: 1024 КВ; 15: 2048 КВ; 16: 4096 КВ.

### 6.3.7 Регистр статистики кэш-попаданий, FCACHE\_CSHR

Регистр содержит значение счетчика попаданий в кэш. Счетчик останавливается при достижении значения 0xFFFF\_FFFF. Запись в регистр сбрасывает счетчик.

Формат регистра FCACHE\_CSHR приведен в следующей таблице.

**Таблица 6.7 Поля регистра FCACHE\_CSHR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	CSHR	RW	0x0	Подсчитывает количество попаданий в кэш. Учитываются только кэшируемые транзакции. Запись в регистр очищает содержимое.

### 6.3.8 Регистр статистики кэш-промахов, FCACHE\_CSMR

Регистр содержит значение счетчика промахов в кэш. Счетчик останавливается при достижении значения 0xFFFF\_FFFF. Запись в регистр сбрасывает счетчик.

Формат регистра FCACHE\_CSMR приведен в следующей таблице.

**Таблица 6.8 Поля регистра FCACHE\_CSMR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	CSMR	RW	0x0	Подсчитывает количество промахов в кэш. Учитываются только кэшируемые транзакции. Запись в регистр очищает содержимое.

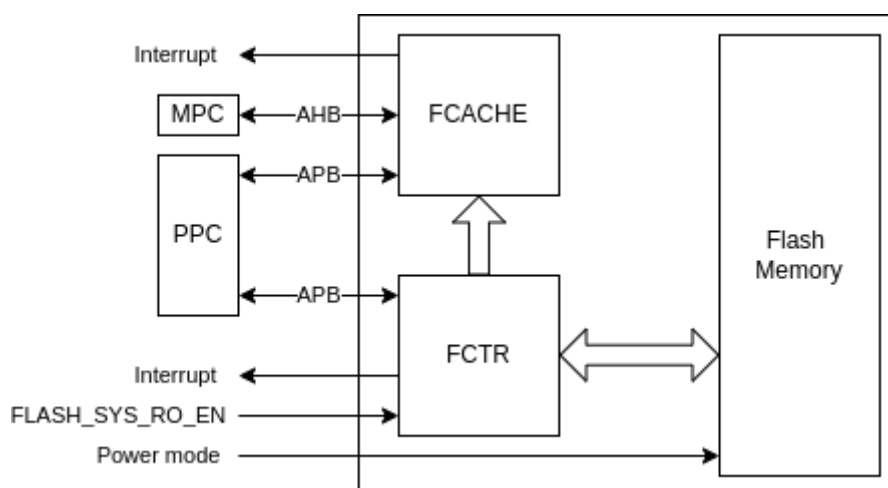
## 7. FLASH: ПОДСИСТЕМА FLASH ПАМЯТИ

### 7.1 Введение

Микросхема имеет встроенную Flash память со следующими характеристиками:

- структура памяти:
  - основной раздел размером 640 Кбайт (80 страниц по 8 Кбайт)
  - системный раздел размером 32 Кбайт (4 страницы по 8 Кбайт)
- 128-разрядное слово данных
- операции постраничного стирания (ERASE) и полного стирания раздела (MAS\_ERASE)
- режимы пониженного потребления Sleep и PowerDown (см. «Системная организация микросхемы» раздел «Питание Flash памяти»)

На Рисунок 7.1 показана структура подсистемы Flash памяти.



**Рисунок 7.1 Структурная схема подсистемы Flash памяти**

Контроллер FCTR предоставляет программный доступ к Flash памяти микросхемы. Основные функции FCTR:

- выполнение операций записи и стирания Flash памяти
- настраиваемое ограничение доступа (чтение/запись) к разделам Flash памяти

Для ускорения чтения из Flash памяти может использоваться кэш (см. «FCACHE: кэш Flash памяти»). После сброса кэширование не выполняется.

В Таблица 7.1 показана адресация Flash памяти. Системная адресация соответствует прямому доступу к памяти через FCACHE (возможны Secure и Non-secure обращения). Доступ к регистрам FCTR возможен только Secure транзакциями.

**Таблица 7.1. Адресация Flash памяти**

Раздел	Размер, Кбайт	Secure адрес в системе	Non-secure адрес в системе	Адресация через FCTR
Основной (Main)	640	0x1000_0000 – 0x1009_FFFF	0x1000_0000 – 0x1009_FFFF	0x00_0000 – 0x09_FFFF
Системный (System)	32	0x1020_0000 – 0x1020_7FFF	-	0x20_0000 – 0x20_7FFF

## 7.2 Описание функционирования

### 7.2.1 Чтение памяти

Чтение Flash памяти может выполняться напрямую через FCACHE либо косвенно через регистры FCTR.

Чтение памяти производится 128-разрядными словами. На интерфейс АНВ данные выдаются 32-разрядными словами в соответствии с запрошенным адресом.

Задержка при доступе к памяти может занимать от одного до нескольких тактов SYSCLK и задается полем READ\_LATENCY регистра FCTR\_PSP\_CTRL. Задержка зависит от частоты тактового сигнала SYSCLK и напряжения питания ядра (задается полем DCDC\_VSEL, см. «Системная организация микросхемы» раздел «Регулятор DC-DC»). Для корректного чтения памяти задержка должна определяться в соответствии с следующей таблицей.

**Таблица 7.2. Задержка чтения Flash памяти**

READ_LATENCY	Частота SYSCLK (МГц)	
	0.9 В (DCDC_VSEL = 0)	1.0/1.1 В (DCDC_VSEL = 1/2)
0	$0 < F_{SYSCLK} \leq 25$	$0 < F_{SYSCLK} \leq 40$
1	-	$40 < F_{SYSCLK} \leq 50$

При увеличении частоты SYSCLK или напряжения питания ядра параметр READ\_LATENCY должен быть обновлен до изменения частоты/напряжения. При уменьшении частоты SYSCLK или напряжения питания ядра параметр READ\_LATENCY должен быть обновлен после изменения частоты/напряжения.

После сброса READ\_LATENCY = 0, что соответствует одному такту задержки.

Чтение через регистры FCTR предназначено только для отладочных целей и выполняется в соответствии с алгоритмом на Рисунок 7.2:

1. С помощью регистра статуса FCTR\_STATUS убедиться, что нет незавершенных команд.
2. Записать адрес считываемого слова в регистр FCTR\_ADDR.
3. Записать команду READ в регистр FCTR\_CTRL.
4. Дождаться завершения команды:

- а. путем опроса регистра FCTR\_IRQ\_STATUS\_CLR до установки бита CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ, либо
  - б. дождаться прерывания CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ.
5. Сбросить статус с помощью регистра FCTR\_IRQ\_STATUS\_CLR.
  6. Считать запрошенные данные из регистров FCTR\_DATA0 – FCTR\_DATA3.

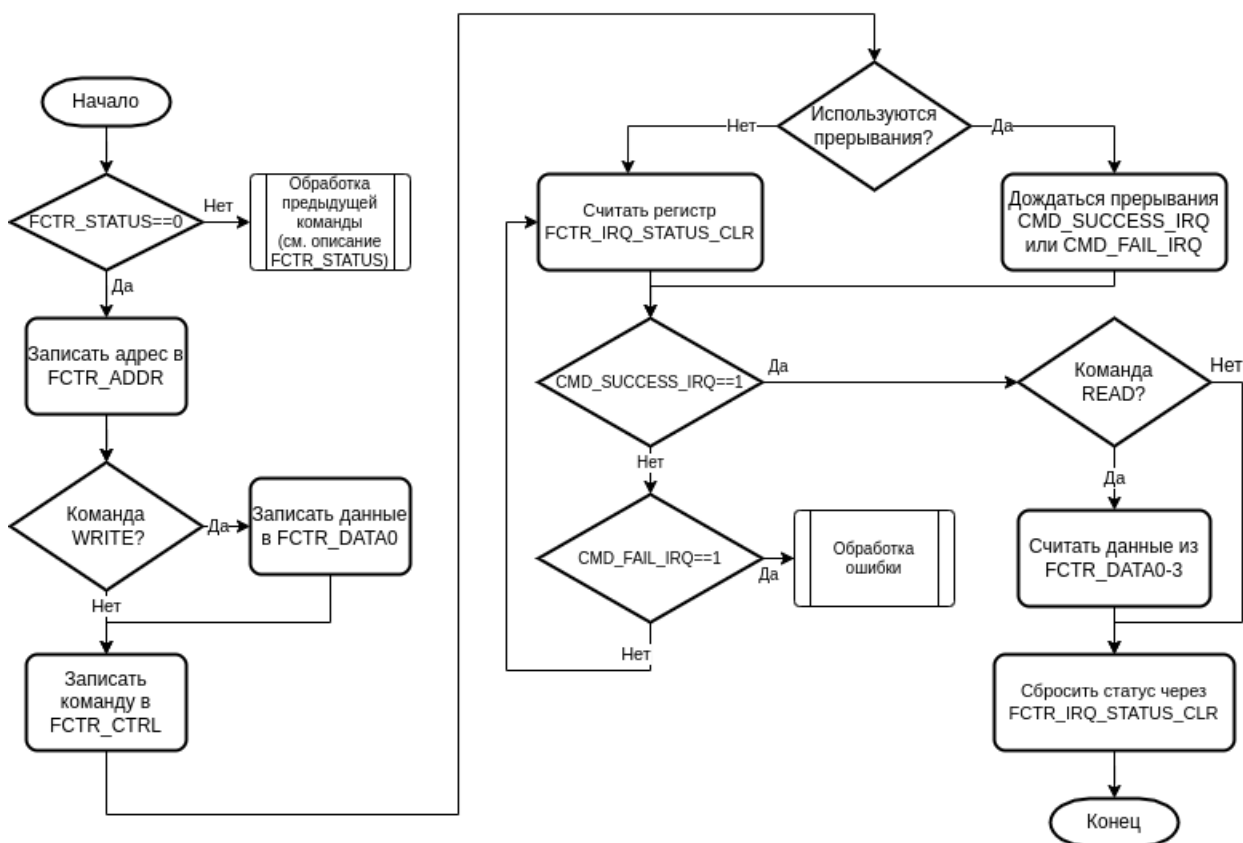


Рисунок 7.2. Алгоритм чтения/записи/стирания Flash памяти

## 7.2.2 Программирование (запись) и стирание памяти

### 7.2.2.1 Ограничения

Выполнение операций записи и стирания занимает длительное время по сравнению с периодом тактового сигнала SYSCLK. Системный сброс, изменение тактовой частоты, напряжения питания ядра, параметров в регистрах FCTR\_PSP\* во время выполнения записи или стирания может привести к порче данных в адресуемых ячейках памяти.

Транзакция чтения во время выполнения записи или стирания приостанавливается до завершения операции записи/стирания.

Для корректного выполнения записи и стирания частота SYSCLK должна быть не менее 100 кГц.

Последовательная запись по одному адресу возможна, если биты изменяются из 1 в 0. Изменение бита из 0 в 1 требует выполнения операции стирания.

После стирания каждое 128-разрядное слово может быть записано не более 4 раз до следующего стирания.

#### 7.2.2.2 Настройка временных параметров

Регистры FCTR\_TM1, FCTR\_TM2 после сброса содержат абсолютные значения временных параметров, необходимых для выполнения записи и стирания памяти. Изменять эти регистры запрещено.

При изменении тактовой частоты SYSCLK требуется подстраивать параметры CYCLES\_1US и CYCLES\_20NS в регистре FCTR\_CTRL по формулам из Таблица 7.16. После сброса значения этих параметров соответствуют частоте 16 МГц.

#### 7.2.2.3 Процедура записи

Запись памяти выполняется 32-разрядными словами с помощью регистров FCTR в следующей последовательности.

1. С помощью регистра статуса FCTR\_STATUS убедиться, что нет незавершенных команд.
2. Записать адрес программируемого слова в регистр FCTR\_ADDR.
3. Записать программируемые данные в регистр FCTR\_DATA0.
4. Записать команду WRITE в регистр FCTR\_CTRL.
5. Дождаться завершения команды:
  - а. путем опроса регистра FCTR\_IRQ\_STATUS\_CLR до установки бита CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ, либо
  - б. дождаться прерывания CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ.
6. Сбросить статус с помощью регистра FCTR\_IRQ\_STATUS\_CLR.

Алгоритм записи памяти изображен на Рисунок 7.2.

#### 7.2.2.4 Процедура стирания

Операция стирания может выполняться над одной страницей памяти (команда ERASE) либо над всеми страницами раздела (команда MASS ERASE).

Для определения номера стираемой страницы младшие 13 бит адреса не используются. Для определения стираемого раздела в случае MASS\_ERASE используется 21-ый бит адреса: 0 – основной раздел, 1 – системный раздел.

Последовательность действий для выполнения стирания:

1. С помощью регистра статуса FCTR\_STATUS убедиться, что нет незавершенных команд.
2. Записать адрес стираемой страницы или раздела в регистр FCTR\_ADDR.

3. Записать в регистр FCTR\_CTRL команду ERASE для стирания страницы или MASS\_ERASE для стирания раздела.
4. Дождаться завершения команды:
  - а. путем опроса регистра FCTR\_IRQ\_STATUS\_CLR до установки бита CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ, либо
  - б. дождаться прерывания CMD\_SUCCESS\_IRQ или CMD\_FAIL\_IRQ.
5. Сбросить статус с помощью регистра FCTR\_IRQ\_STATUS\_CLR.

Алгоритм стирания памяти изображен на Рисунок 7.2.

## 7.3 Регистры

### 7.3.1 Перечень регистров FCTR

Таблица 7.3. Регистры FCTR

Условное обозначение	Описание	Исходное состояние	Смещение
IRQ_ENABLE_SET	Регистр управления прерываний. Генерация прерываний для различных источников.	0x00000000	0x000
IRQ_ENABLE_CLR	Регистр управления прерываний. Выключение генераций прерываний для различных источников.	0x00000000	0x004
IRQ_STATUS_SET	Регистр состояния прерываний для различных источников	0x00000000	0x008
IRQ_STATUS_CLR	Очистка регистра состояния прерываний для различных источников.	0x00000000	0x00C
IRQ_MASKED_STATUS	Регистр состояния прерывания, который показывает статус запросов на прерывание.	0x00000000	0x010
CTRL	Управляющий регистр для инициализации к накристальной памяти.	0x00000000	0x014
STATUS	Регистр состояния, который показывает состояние обращения к накристальной памяти.	0x00000000	0x018
ADDR	Регистр адресов для доступа к накристальной памяти.	0x00000000	0x01C
DATA0	Регистр данных для доступа к накристальной памяти на чтение и запись. Биты [31:0].	0x00000000	0x020
DATA1	Регистр данных для доступа к накристальной памяти на чтение. Биты [63:32].	0x00000000	0x024
DATA2	Регистр данных для доступа к накристальной памяти на чтение. Биты [95:64].	0x00000000	0x028
DATA3	Регистр данных для доступа к накристальной памяти на чтение. Биты [127:96].	0x00000000	0x02C
PIDR4	Байт периферийного регистра идентификации [4]. Возвращает байт [4] периферийного идентификатора. Регистр PIDR4 является частью набора периферийных идентификационных регистров.	0x00000004	0xFD0
PIDR0	Байт периферийного регистра идентификации [0]. Возвращает байт [0] периферийного идентификатора. Регистр PIDR0 является частью набора периферийных идентификационных регистров.	0x00000032	xFE0
PIDR1	Байт периферийного регистра идентификации [1]. Возвращает байт [1] периферийного идентификатора. Регистр PIDR1 является частью набора	0x000000B8	xFE4



Условное обозначение	Описание	Исходное состояние	Смещение
	периферийных идентификационных регистров.		
PIDR2	Байт периферийного регистра идентификации [2]. Возвращает байт [2] периферийного идентификатора. Регистр PIDR2 является частью набора периферийных идентификационных регистров.	0x0000000B	0xFE8
PIDR3	Байт периферийного регистра идентификации [3]. Возвращает байт [3] периферийного идентификатора. Регистр PIDR3 является частью набора периферийных идентификационных регистров.	0x00000000	0xFEC
CIDR0	Регистр идентификации компонентов 0. Возвращает байт [0] идентификатора компонента. Регистр CIDR 0 является частью набора регистров идентификации компонентов.	0x0000000D	0xFF0
CIDR1	Регистр идентификации компонентов 1. Возвращает байт [1] идентификатора компонента. Регистр CIDR 1 является частью набора регистров идентификации компонентов.	0x000000F0	0xFF4
CIDR2	Регистр идентификации компонентов 2. Возвращает байт [2] идентификатора компонента. Регистр CIDR 2 является частью набора регистров идентификации компонентов.	0x00000005	0xFF8
CIDR3	Регистр идентификации компонентов 3. Возвращает байт [3] идентификатора компонента. Регистр CIDR 3 является частью набора регистров идентификации компонентов.	0x000000B1	0xFFC
PSP_CTRL	Регистр управления блоком PSP	0x000F0000	0x1000
PSP_TM1	Регистр управления временными интервалами	0x64050802	0x1004
PSP_TM2	Регистр управления временными интервалами	0x0A0F1313	0x1008
PSP_STATUS	Регистры состояния блока PSP	0x00000000	0x100C
PSP_REPAIR_INFO	Регистр информации по восстановлению	*	0x1010

### 7.3.2 Регистр IRQ\_ENABLE\_SET

Регистр управления прерываний. Генерация прерываний для различных источников.

Таблица 7.4. Регистры IRQ\_ENABLE\_SET

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
4	READ_OVERFLOW_IRQ_EN_SET	RW	0	Позволяет биту READ_OVERFLOW_IRQ генерировать прерывания в строке прерывания. Запись 1 устанавливает включение прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
3	CMD_REJECT_IRQ_EN_SET	RW	0	Позволяет биту CMD_REJECT_IRQ генерировать прерывания в строке прерывания. Запись 1 устанавливает включение прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
2	CMD_FAIL_IRQ_EN_SET	RW	0	Позволяет биту CMD_FAIL_IRQ генерировать прерывания в строке прерывания. Запись 1 устанавливает включение прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
1	CMD_SUCCESS_IRQ_EN_SET	RW	0	Позволяет биту CMD_SUCCESS_IRQ генерировать прерывания в строке прерывания. Запись 1 устанавливает включение прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
0	CMD_ACCEPT_IRQ_EN_SET	RW	0	Позволяет биту CMD_ACCEPT_IRQ генерировать прерывания в строке прерывания. Запись 1 устанавливает включение прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.

### 7.3.3 Регистр IRQ\_ENABLE\_CLR

Регистр управления прерываний. Выключение генераций прерываний для различных источников.

Таблица 7.5. Регистры IRQ\_ENABLE\_CLR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв
4	READ_OVERFLOW_IRQ_EN_CLR	RW	0	Отключает бит READ_OVERFLOW_IRQ. Запись 1 очищает функцию включения прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
3	CMD_REJECT_IRQ_EN_CLR	RW	0	Отключает бит CMD_REJECT_IRQ. Запись 1 очищает функцию включения прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
2	CMD_FAIL_IRQ_EN_CLR	RW	0	Отключает бит CMD_FAIL_IRQ. Запись 1 очищает функцию включения прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
1	CMD_SUCCESS_IRQ_EN_CLR	RW	0	Отключает бит CMD_SUCCESS_IRQ. Запись 1 очищает функцию включения прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.
0	CMD_ACCEPT_IRQ_EN_CLR	RW	0	Отключает бит CMD_ACCEPT_IRQ. Запись 1 очищает функцию включения прерывания. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита.

### 7.3.4 Регистр IRQ\_STATUS\_SET

Регистр состояния прерываний для различных источников

Таблица 7.6. Регистры IRQ\_STATUS\_SET

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Reserved
4	READ_OVERFLOW_IRQ_STS_SET	RW	0	Задаёт состояние бита READ_OVERFLOW_IRQ. Запись 1 устанавливает прерывание. Запись 0 не имеет никакого эффекта. Может использоваться для принудительного задания установленного состояния этого бита прерывания в целях отладки. Чтение этого бита показывает текущее состояние бита прерывания независимо от настройки включения прерывания.
3	CMD_REJECT_IRQ_STS_SET	RW	0	Задаёт состояние бита CMD_REJECT_IRQ. Запись 1 устанавливает прерывание. Запись 0 не имеет никакого эффекта. Может использоваться для принудительного задания установленного состояния этого бита прерывания в целях отладки. Чтение этого бита показывает текущее состояние бита прерывания независимо от настройки включения прерывания.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
2	CMD_FAIL_IRQ_STS_SET	RW	0	Задаёт состояние бита CMD_FAIL_IRQ. Запись 1 устанавливает прерывание. Запись 0 не имеет никакого эффекта. Может использоваться для принудительного задания установленного состояния этого бита прерывания в целях отладки. Чтение этого бита показывает текущее состояние бита прерывания независимо от настройки включения прерывания. Примечание. Запись 1 также устанавливает STATUS.CMD_FAIL.
1	CMD_SUCCESS_IRQ_STS_SET	RW	0	Задаёт состояние бита CMD_SUCCESS_IRQ. Запись 1 устанавливает прерывание. Запись 0 не имеет никакого эффекта. Может использоваться для принудительного задания установленного состояния этого бита прерывания в целях отладки. Чтение этого бита показывает текущее состояние бита прерывания независимо от настройки включения прерывания. Примечание. Запись 1 также устанавливает STATUS.CMD_SUCCESS.
0	CMD_ACCEPT_IRQ_STS_SET	RW	0	Задаёт состояние бита CMD_ACCEPT_IRQ. Запись 1 устанавливает прерывание. Запись 0 не имеет никакого эффекта. Может использоваться для принудительного задания установленного состояния этого бита прерывания в целях отладки. Чтение этого бита показывает текущее состояние бита прерывания независимо от настройки включения прерывания.

### 7.3.5 Регистр IRQ\_STATUS\_CLR

Очистка регистра состояния прерываний для различных источников.

Таблица 7.7. Регистры IRQ\_STATUS\_CLR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв
4	READ_OVERFLOW_IRQ_STS_CLR	RW	0	Очищает состояние бита READ_OVERFLOW_IRQ. Запись 1 очищает прерывание. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита прерывания.
3	CMD_REJECT_IRQ_STS_CLR	RW	0	Очищает состояние бита CMD_REJECT_IRQ. Запись 1 очищает прерывание. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита прерывания.
2	CMD_FAIL_IRQ_STS_CLR	RW	0	Очищает состояние бита CMD_FAIL_IRQ. Запись 1 очищает прерывание. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита прерывания.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
1	CMD_SUCCESS_IRQ_STS_CLR	RW	0	Очищает состояние бита CMD_SUCCESS_IRQ. Запись 1 очищает прерывание. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита прерывания.
0	CMD_ACCEPT_IRQ_STS_CLR	RW	0	Очищает состояние бита CMD_ACCEPT_IRQ. Запись 1 очищает прерывание. Запись 0 не имеет никакого эффекта. Чтение этого бита показывает текущее состояние бита прерывания.

### 7.3.6 Регистр IRQ\_MASKED\_STATUS

Регистр состояния прерывания, который показывает статус запросов на прерывание.

Таблица 7.8. Регистры IRQ\_MASKED\_STATUS

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв
4	READ_OVERFLOW_IRQ	RO	0	Когда этот бит 1, это означает, что READ_OVERFLOW_IRQ является причиной прерывания.
3	CMD_REJECT_IRQ	RO	0	Когда этот бит 1, это означает, что CMD_REJECT_IRQ является причиной прерывания.
2	CMD_FAIL_IRQ	RO	0	Когда этот бит 1, это означает, что CMD_FAIL_IRQ является причиной прерывания.
1	CMD_SUCCESS_IRQ	RO	0	Когда этот бит 1, это означает, что CMD_SUCCESS_IRQ является причиной прерывания.
0	CMD_ACCEPT_IRQ	RO	0	Когда этот бит 1, это означает, что CMD_ACCEPT_IRQ является причиной прерывания.

### 7.3.7 Регистр CTRL

Управляющий регистр для инициализации к накристалльной памяти.

Таблица 7.9. Регистры CTRL

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:5	-	RO	0	Резерв
4	ABORT	RW	0	Прерывание текущей команды к накристалльной памяти через интерфейс APB. При записи ABORT поле CMD игнорируется. ABORT имеет эффект только при наличии STATUS.CMD_ACCEPT. Чтение ABORT показывает состояние сигнала прерывания интерфейса GFB.
3	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
2:0	CMD	RW	000	<p>Иницирует команду для обращения к накрестальной памяти. Адрес берется из регистра ADDR, данные из регистра DATA 0 (для команд записи и записи строк). Программное обеспечение должно гарантировать, что состояние адресованного слова очищено. Эти команды таковы:</p> <p>001: READ  010: WRITE  011: ROW WRITE  100: ERASE  111: MASS ERASE  000: Резерв  101: Резерв  110: Резерв</p> <p>Запись в резервные поля не имеет никакого эффекта. Чтение CMD показывает состояние в настоящее время. Когда CMD принимается, поле CMD принимает значение 0b000</p>

### 7.3.8 Регистр STATUS

Регистр состояния, который показывает состояние обращения к накрестальной памяти.

Таблица 7.10. Регистры STATUS

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:6	-	RO	0	Резерв
5	ARBITRATION_LOCKED	RO	0	Установка этого бита означает, что другой интерфейс держит арбитраж в заблокированном состоянии, и никакие команды APB не обслуживаются до тех пор, пока заблокированное состояние не будет удалено. Это может быть использовано программным обеспечением для определения причины задержки запроса APB.
4	CMD_FINISH	RO	0	Установка этого бита означает, что ранее принятая команда завершена, но результат не может быть обновлен в регистре состояния, так как биты прерывания все еще установлены. Когда биты прерывания результата очищены, состояние может быть обновлено, и этот бит автоматически очищается.
3	CMD_FAIL	RO	0	Установка этого бита означает, что ранее принятая команда завершилась с ошибкой. Либо этот бит, либо бит CMD_SUCCESS устанавливаются для завершенной команды. Обновляется при установке параметра CMD_SUCCESS_IRQ или CMD_FAIL_IRQ. Действителен до тех пор, пока не будет очищен бит CMD_SUCCESS_IRQ или CMD_FAIL_IRQ.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
2	CMD_SUCCESS	RO	0	Установка этого бита означает, что ранее принятая команда завершилась успешно. Либо этот бит, либо бит CMD_FAIL устанавливаются для завершенной команды. Обновляется при установке параметра CMD_SUCCESS_IRQ или CMD_FAIL_IRQ. Действителен до тех пор, пока не будет очищен бит CMD_SUCCESS_IRQ или CMD_FAIL_IRQ.
1	CMD_ACCEPT	RO	0	Бит равным 1 и означает, что команда пересылается в накристальную память. Принимает значение 1, когда бит CMD_PENDING будет очищен. Очищается, когда команда завершена и биты состояния CMD_SUCCESS_IRQ или CMD_FAIL_IRQ очищаются для предыдущей передачи. Если результаты состояния предыдущей команды не очищаются, команда все еще может быть завершена для доступа к накристальной памяти, но результаты не могут обновиться до тех пор, пока не будут очищены предыдущие результаты. Если после завершения текущей команды принимается уже ожидающая команда, то этот бит сохранит значение 1.
0	CMD_PENDING	RO	0	Когда регистр CTRL записан, команда переходит в очередь арбитража и ждет доступа к накристальной памяти. Этот бит устанавливается, когда команда инициирована, но все еще находится в очереди, и очищается, когда команда принимается арбитром. Этот бит также устанавливается, когда прерывание записывается в регистр CTRL, и очищается, когда прерванная команда завершается.

### 7.3.9 Регистр ADDR

Регистр адресов для доступа к накристальной памяти.

Таблица 7.11. Регистры ADDR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:22	-	RO	0	Резерв
21:0	ADDR	RW	0	22-битный адрес для текущего запроса в накристальную память. Позволяет использовать диапазон адресов 2 МБ для основной области и диапазон адресов 2 МБ для расширенной области. ADDR [21] выбирает между диапазонами памяти: 0: Основная область 1: Расширенная область При выборе расширенной области для команд массового стирания очищаются как основные, так и расширенные области. В противном случае очищается только основная область. Содержимое расширенной области остается неизменным после завершения массового стирания.

### 7.3.10 Регистр DATA0

Регистр данных для доступа к накристальной памяти имеет биты чтения и записи [31:0].

**Таблица 7.12. Регистры DATA0**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA0	RW	0	Регистр данных для доступа к накристальной памяти имеет биты чтения и записи [31:0].

### 7.3.11 Регистр DATA1

Регистр данных для доступа к накристальной памяти имеет биты чтения [63:32].

**Таблица 7.13. Регистры DATA1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA1	RO	0	Регистр данных для доступа к накристальной памяти имеет биты чтения [63:32].

### 7.3.12 Регистр DATA2

Регистр данных для доступа к накристальной памяти имеет биты чтения [95:64].

**Таблица 7.14. Регистры DATA2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA2	RO	0	Регистр данных для доступа к накристальной памяти имеет биты чтения [95:64].

### 7.3.13 Регистр DATA3

Регистр данных для доступа к накристальной памяти имеет биты чтения [127:96].

**Таблица 7.15. Регистры DATA3**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA3	RO	0	Регистр данных для доступа к накристальной памяти имеет биты чтения [127:96].

### 7.3.14 Описание регистра PSP\_CTRL

Регистры управления

Формат регистра PSP\_CTRL приведен в следующей таблице:

**Таблица 7.16. Поля регистра PSP\_CTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:25	-	RO	0	Резерв



Номер бита	Название	Тип доступа	Исходное состояние	Назначение
24	CYCLES_20NS	RW	0	Настройка количества тактов частоты SYSCLK для отсчитывания интервала длительностью 20 нс. CYCLES_20NS = <количество тактов> - 1. Количество тактов округляется до большего целого.
23:22	-	RO	0	Резерв
22:16	CYCLES_1US	RW	0x0F	Настройка количества тактов частоты SYSCLK для отсчитывания интервала длительностью 1 мкс. CYCLES_1US = <количество тактов> - 1. Количество тактов округляется до большего целого.
15	-	RO	0	Резерв
14	SYS_LOCK_WR	Write-once	0	Запись '1' блокирует запись в системный раздел памяти. Запись '0' игнорируется. Блокировка отключается только при «холодном» сбросе.
13	SYS_LOCK_RD	Write-once	0	Запись '1' блокирует чтение из системного раздела памяти. Запись '0' игнорируется. Блокировка отключается только при «холодном» сбросе.
12	MAIN_LOCK_WR	Write-once	0	Запись '1' блокирует запись в основной раздел памяти. Запись '0' игнорируется. Блокировка отключается только при «холодном» сбросе.
11	MAIN_LOCK_RD	Write-once	0	Запись '1' блокирует чтение из основного раздела памяти. Запись '0' игнорируется. Блокировка отключается только при «холодном» сбросе.
10:3	-	RO	0	Резерв
2:0	READ_LATENCY	RW	0	Задержка при чтении Flash памяти, вычисляемая как (READ_LATENCY + 1) тактов частоты SYSCLK. Задается в соответствии с рабочей частотой SYSCLK и напряжением питания ядра (см. пункт «Чтение памяти»).

### 7.3.15 Описание регистра PSP\_TM1

Формат регистра PSP\_TM1 приведен в следующей таблице:

**Таблица 7.17. Поля регистра PSP\_TM1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	-	RO	0	Резерв
30:24	TNVH1	RW	100	Длительность интервала T <sub>NVH1</sub> , мкс
23:20	-	RO	0	Резерв
19:16	TNVH	RW	5	Длительность интервала T <sub>NVH</sub> , мкс
15:12	-	RO	0	Резерв
11:8	TNVS	RW	8	Длительность интервала T <sub>NVS</sub> , мкс
7:4	-	RO	0	Резерв
3:0	TPGS	RW	2	Длительность интервала T <sub>PGS</sub> , мкс

### 7.3.16 Описание регистра PSP\_TM2

Формат регистра PSP\_TM2 приведен в следующей таблице:

**Таблица 7.18. Поля регистра PSP\_TM2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:28	-	RO	0	Резерв
27:24	TRCV	RW	10	Длительность интервала $T_{RCV}$ , мкс
23:21	-	RO	0	Резерв
20:16	TPROG	RW	15	Длительность интервала $T_{PROG}$ , мкс
15:13	-	RO	0	Резерв
12:8	TERASE	RW	19	Длительность интервала $T_{ERASE}$ для команды ERASE, мс
7:5	-	RO	0	Резерв
4:0	TMERASE	RW	19	Длительность интервала $T_{ERASE}$ для команды MASS_ERASE, мс

### 7.3.17 Описание регистра PSP\_STATUS

Формат регистра PSP\_STATUS приведен в следующей таблице:

**Таблица 7.19. Поля регистра PSP\_STATUS**

Диапазон	Название	Описание	Сброс
31:15	-	Резерв	0
14	ERR_SYS_WR	Бит устанавливается при попытке записи/стирания системного раздела памяти при $SYS\_LOCK\_WR = 1$ либо при $FLASH\_SYS\_RO\_EN = 1$	0
13	ERR_SYS_RD	Бит устанавливается при попытке чтения системного раздела памяти при $SYS\_LOCK\_RD = 1$	0
12	ERR_MAIN_WR	Бит устанавливается при попытке записи/стирания основного раздела памяти при $MAIN\_LOCK\_WR = 1$	0
11	ERR_MAIN_RD	Бит устанавливается при попытке чтения основного раздела памяти при $MAIN\_LOCK\_RD = 1$	0
10	-	Резерв	0
9	ERR_SYS_ADDR	Бит устанавливается если адрес операции чтения/записи/стирания системного раздела выходит за допустимый диапазон памяти	0
8	ERR_MAIN_ADDR	Бит устанавливается если адрес операции чтения/записи/стирания основного раздела выходит за допустимый диапазон памяти	0
7:0	-	Резерв	0

### 7.3.18 Описание регистра PSP\_REPAIR\_INFO

Формат регистра PSP\_REPAIR\_INFO приведен в следующей таблице.

**Таблица 7.20. Поля регистра PSP\_REPAIR\_INFO**

Диапазон	Название	Описание	Сброс
31	VALID	Флаг валидности информации в данном регистре	0
30:24	-	Резерв	0
23:17	PAGE1	Номер заменяемой страницы	0
16	PAGE1_ENB	Флаг использования резервной страницы 1	0
15:8	-	Резерв	0
7:1	PAGE0	Номер заменяемой страницы	0
0	PAGE0_ENB	Флаг использования резервной страницы 0	0

## 8. JTM: СЕНСОРЫ ТЕМПЕРАТУРЫ И НАПРЯЖЕНИЯ

### 8.1 Введение

Блок JTM предназначен для измерения температуры кристалла и напряжения в контрольных точках с помощью АЦП и интегрированного датчика температуры. JTM может быть использован для применения температурной компенсации входящих в состав микросхемы осцилляторов.

Основные возможности JTM:

- Дифференциальный  $\Sigma\Delta$  АЦП с разрешением 12 бит
- Длительность преобразования АЦП ~4.2 мс
- Измерение напряжения внутренних цепей питания микросхемы
- Измерение температуры кристалла в диапазоне  $-40\text{ }^{\circ}\text{C} - +125\text{ }^{\circ}\text{C}$  с шагом  $0.25\text{ }^{\circ}\text{C}$
- Аппаратное вычисление значения температуры по результату АЦП

### 8.2 Описание функционирования

#### 8.2.1 Запуск измерения

Для запуска измерения температуры или напряжения необходимо:

1. Убедиться путем чтения бита READY регистра JTM\_STAT, что в данный момент не выполняется другое измерение.
2. При измерении температуры задать калибровочные параметры в регистрах JTM\_CFG0, JTM\_CFG1.
3. При необходимости разрешить прерывание по завершению измерения в регистре JTM\_INTEN.
4. Выбрать канал (параметр CHSEL) и режим измерения (параметр MODE) в регистре JTM\_CTR.
5. Запустить измерение записью 1 в бит SOC. Запись может быть совмещена с предыдущим пунктом.
6. Дождаться завершения измерения, обозначаемого установкой бита READY или соответствующего прерывания.

Результат измерения температуры в градусах Цельсия содержится в поле TEMP регистра JTM\_STAT.

При измерении напряжения результат аналого-цифрового преобразования содержится в поле CODE. Напряжение  $V$  вычисляется по формуле:

$$V = V_{FS} \frac{CODE}{4096},$$

где  $V_{FS}$  – максимальное измеряемое напряжение для заданного режима:  $V_{FS} = 1.0$  В при  $MODE = 0x1$ ,  $0.5$  В при  $MODE = 0x2$ ,  $2.0$  В при  $MODE = 0x3$ .

## 8.3 Регистры

### 8.3.1 Перечень регистров JTM

Таблица 8.1. Регистры JTM

Условное обозначение	Название регистра	Исходное состояние	Смещение
JTM_CFG0	Конфигурационный регистр 0	0x00	0x000
JTM_CFG1	Конфигурационный регистр 1	0x00	0x004
JTM_CTR	Регистр управления	0x00	0x008
JTM_STAT	Регистр статуса	0x01	0x00C
JTM_INTEN	Регистр разрешения прерываний	0x00	0x010
JTM_INTSTAT	Регистр статуса прерываний	0x00	0x014
JTM_INTCLR	Регистр сброса прерываний	0x00	0x018

### 8.3.2 Регистр JTM\_CFG0

Таблица 8.2. Регистр JTM\_CFG0

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:27	-	RO	0x0	Резерв
26:16	TCAL	RW	0x0 <sup>1</sup>	Калибровочный параметр.
15:11	-	RO	0x0	Резерв
10:0	WCAL	RW	0x0 <sup>1</sup>	Калибровочный параметр.
Примечания:				
1. Значения калибровочных параметров после сброса автоматически обновляются из памяти ОТР.				

### 8.3.3 Регистр JTM\_CFG1

Таблица 8.3. Регистр JTM\_CFG1

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:25	-	RO	0x0	Резерв
24	CALS	RW	0x0 <sup>1</sup>	Калибровочный параметр
23	-	RO	0x0	Резерв
22:16	WTCALCONF	RW	0x0 <sup>1</sup>	Калибровочный параметр.
15:10	-	RO	0x0	Резерв
9:0	WTCONF	RW	0x0 <sup>1</sup>	Калибровочный параметр.
Примечания:				
1. Значения калибровочных параметров после сброса автоматически обновляются из памяти ОТР.				

### 8.3.4 Регистр JTM\_CTR

Таблица 8.4. Регистр JTM\_CTR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0x0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
10:8	CHSEL	RW	0x0	Выбор канала: 0x0: канал 1, температура; 0x1-0x3: резерв; 0x4: канал 5, внутреннее напряжение питания 1.3 В. Используется только для диагностики; 0x5-0x6: резерв; 0x7: канал 8, напряжение питания ядра с вывода VDDC
7:6	-	RO	0x0	Резерв
5:4	MODE	RW	0x0	Режим преобразования: 0x0: $\Delta V_{be}$ измерение температуры; 0x1: измерение напряжения, $V_{FS} = 1.0$ В; 0x2: измерение напряжения, $V_{FS} = 0.5$ В; 0x3: измерение напряжения, $V_{FS} = 2.0$ В
3:1	-	RO	0x0	Резерв
0	SOC	RW	0x0	Запись 1 запускает преобразование. Поле сбрасывается при завершении преобразования

### 8.3.5 Регистр JTM\_STAT

Таблица 8.5. Регистр JTM\_STAT

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:20	CODE	RO	0x0	Результат аналого-цифрового преобразования по выбранному каналу
19:8	TEMP	RO	0x0	Результат измерения температуры в единицах 0,25 °C
7:1	-	RO	0x0	Резерв
0	READY	RO	0x1	Статус преобразования: 0x0: преобразование в процессе; 0x1: преобразование завершено либо еще не запускалось

### 8.3.6 Регистр JTM\_INTEN

Таблица 8.6. Регистр JTM\_INTEN

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0x0	Резерв
0	READY	RW	0x0	Разрешение прерывания по установке флага READY

### 8.3.7 Регистр JTM\_INTSTAT

Таблица 8.7. Регистр JTM\_INTSTAT

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0x0	Резерв
0	READY	RO	0x0	Статус прерывания по установке флага READY

### 8.3.8 Регистр JTM\_INTCLR

Таблица 8.8. Регистр JTM\_INTCLR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0x0	Резерв
0	READY	WO	0x0	Запись 1 сбрасывает прерывания по установке флага READY. Запись 0 игнорируется



## 9. ИОСТР: МНОГОФУНКЦИОНАЛЬНЫЕ ПОРТЫ ВВОДА/ВЫВОДА

### 9.1 Введение

Микросхема имеет четыре многофункциональных порта ввода вывода PA, PB, PC, PD. Каждый порт имеет 16 выводов.

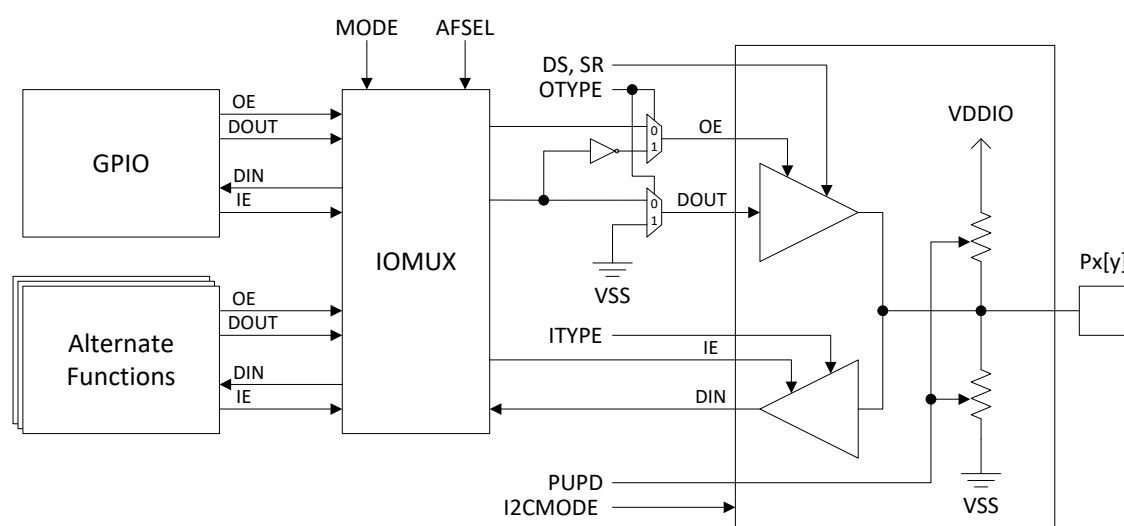
Основные возможности многофункциональных портов:

- Каждый вывод может быть использован как GPIO либо подключен к одной из альтернативных функций
- Работа выходных буферов в режимах push-pull или open drain
- Конфигурируемая подтяжка pull-up/pull-down
- Конфигурируемая скорость работы выходных буферов
- Специальный режим для интерфейса I2C

### 9.2 Описание функционирования

#### 9.2.1 Структура выводов

На следующем рисунке показана структура и схема подключения одного из многофункциональных выводов  $Px[y]$  ( $x = A, B, C, D$ ;  $y = 0 - 15$ ). Защитные диоды ESD на рисунке не отмечены.



**Рисунок 9.1. Структура выводов многофункциональных портов**

Настройка многофункциональных выводов выполняется с помощью блока ИОСТР.

Блок IOMUX служит для выбора функции вывода.

### 9.2.2 Режим работы

Режим работы выводов порта P<sub>x</sub> (x = A, B, C, D) задается регистром IOCTR\_P<sub>x</sub>\_MODE. Возможные варианты:

- Hi-Z состояние – входной и выходной буферы выключены.
- Режим GPIO (основная функция вывода) – управление направлением работы и передачей данных выполняется блоками GPIO0-3. При этом вывод у порта PA соответствует блоку GPIO0, порта PB – блоку GPIO1 и т.д.
- Режим альтернативных функций – управление выводом передается одной из функций, выбранной с помощью регистров IOCTR\_P<sub>x</sub>\_AFL, IOCTR\_P<sub>x</sub>\_AFH. Возможные функции каждого вывода приведены в разделе 35 «ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ».

После «холодного» сброса микросхемы режим выводов PA10 – PA14 соответствует альтернативной функции #0 – интерфейс JTAG. Режим остальных выводов – Hi-Z.

В режиме основной или альтернативной функции подключаемые через IOMUX блоки управляют сигналами разрешения работы буферов (IE, OE) и сигналом выходных данных (DOUT). На вход выбранной функции подключается сигнал DIN от входного буфера. На входы неактивных функций подается фиксированное значение 0 или 1 в зависимости от типа интерфейса.

Настройки PUPD, DS, SR, ITYPE, I2CMODE не зависят от выбранной функции. Для выводов PA10 – PA15 при работе с функцией AFSEL = 0 параметр OTYPE не применим. Для других функций выводов PA10 – PA15, а также для всех остальных выводов настройка OTYPR не зависит от выбранной функции.

### 9.2.3 Управление подтяжкой

Каждый вывод имеет встроенные резисторы для подтяжки к земле или питанию. Подтяжка выводов включается регистром IOCTR\_P<sub>x</sub>\_PUPD.

### 9.2.4 Тип буферов

Выходной буфер каждого вывода может работать в режиме push-pull или open-drain в зависимости от настройки регистра IOCTR\_P<sub>x</sub>\_OTYPE. Режим с открытым стоком не применим для выводов PA10 – PA15 при работе с функцией #0.

Тип входного буфера задается регистром IOCTR\_P<sub>x</sub>\_ITYPE и может быть КМОП-входом либо входом с триггером Шмитта.

### 9.2.5 Настройка мощности буферов

Выходные буферы позволяют настраивать максимальный выходной ток, а также скорость нарастания сигнала с помощью регистров IOCTR\_Px\_DS и IOCTR\_Px\_SR соответственно.

### 9.2.6 Режим I2C

Выходные буферы выводов PA6, PA7, PB9, PB10 могут быть переключены в режим I2C с помощью регистров IOCTR\_Px\_I2CMODE. Данная функция предназначена для работы в режимах интерфейса I2C Standard Mode, Fast Mode, Fast Mode Plus.

При I2CMODE = 1 активируется встроенный glitch-фильтр, не пропускающий импульсы длительностью менее 50 нс. Также, выходной каскад переключается в режим с открытым стоком.

### 9.2.7 Режимы пониженного потребления

Блок IOCTR находится в домене питания PD\_AON, поэтому при переходе в режим STANDBY состояние регистров IOCTR сохраняется.

Если выводы PA10 - PA15 подключены к альтернативной функции #0, то в режиме STANDBY их функция сохраняется. В противном случае буферы IO указанных выводов выключаются.

Буферы IO для всех остальных выводов, кроме PA10 - PA15, выключаются при переходе в режим STANDBY.

В режиме SHUTDOWN настройки IOCTR сбрасываются, а буферы IO всех выводов переходят в состояние Hi-Z.

## 9.3 Регистры

### 9.3.1 Перечень регистров IOCTR

Таблица 9.1. Регистры IOCTR

Условное обозначение	Название регистра	Исходное состояние	Тип доступа	Смещение
IOCTR_Px_MODE	Регистр настройки режимов выводов порта x	<sup>1</sup>	RW	OFFS <sup>2</sup> + 0x000
IOCTR_Px_AFL	Регистр выбора альтернативной функции выводов 0–7 порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x004
IOCTR_Px_AFH	Регистр выбора альтернативной функции выводов 8–15 порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x008
IOCTR_Px_PUPD	Регистр настройки подтяжки выводов порта x	<sup>1</sup>	RW	OFFS <sup>2</sup> + 0x00C
IOCTR_Px_DS	Регистр настройки максимального выходного тока выводов порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x010
IOCTR_Px_SR	Регистр настройки скорости нарастания сигнала на выходных буферах порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x014
IOCTR_Px_OTYPE	Регистр настройки типа выходных буферов порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x018
IOCTR_Px_ITYPE	Регистр настройки типа входных буферов порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x01C
IOCTR_Px_I2CMODE <sup>3</sup>	Регистр включения режима I2C выводов порта x	0x0000 0000	RW	OFFS <sup>2</sup> + 0x020
Примечания: 1 – исходное состояние для каждого порта указано в описании соответствующего регистра. 2 – OFFS – смещение для порта: <ul style="list-style-type: none"> <li>• 0x000 (PA)</li> <li>• 0x100 (PB)</li> <li>• 0x200 (PC)</li> <li>• 0x300 (PD)</li> </ul> 3 – Регистр Px_I2CMODE доступен только для портов A, B. Для остальных портов соответствующий адрес зарезервирован.				

### 9.3.2 Регистр IOCTR\_Px\_MODE

Исходное состояние:

- PA\_MODE: 0x2AA0 0000;
- PB\_MODE, PC\_MODE, PD\_MODE: 0x0000 0000

Таблица 9.2. Регистр IOCTR\_Px\_MODE

Номер бита	Условное обозначение	Назначение
31:30	MODE15	MODEy задает режим работы вывода y порта x: <ul style="list-style-type: none"> <li>• 00: входной и выходной буферы выключены (Hi-Z состояние);</li> <li>• 01: ввод/вывод общего назначения (GPIOx[y]);</li> <li>• 10: альтернативная функция, определяемая регистрами IOCTR_Px_AFL/IOCTR_Px_AFH;</li> </ul>
29:28	MODE14	
27:26	MODE13	
25:24	MODE12	
23:22	MODE11	
21:20	MODE10	

Номер бита	Условное обозначение	Назначение
19:18	MODE9	<ul style="list-style-type: none"> <li>11: резерв</li> </ul>
17:16	MODE8	
15:14	MODE7	
13:12	MODE6	
11:10	MODE5	
9:8	MODE4	
7:6	MODE3	
5:4	MODE2	
3:2	MODE1	
1:0	MODE0	

### 9.3.3 Регистр IOCTR\_Px\_AFL

Таблица 9.3. Регистр IOCTR\_Px\_AFL

Номер бита	Условное обозначение	Назначение
3, 7, 11, 15, 19, 23, 27, 31	-	Резерв
30:28	AFSEL7	AFSELY задает номер альтернативной функции вывода у порта x (см. раздел 35 «ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ»): <ul style="list-style-type: none"> <li>0: AF0;</li> <li>...</li> <li>7: AF7</li> </ul>
26:24	AFSEL6	
22:20	AFSEL5	
18:16	AFSEL4	
14:12	AFSEL3	
10:8	AFSEL2	
6:4	AFSEL1	
2:0	AFSEL0	

### 9.3.4 Регистр IOCTR\_Px\_AFH

Таблица 9.4. Регистр IOCTR\_Px\_AFH

Номер бита	Условное обозначение	Назначение
3, 7, 11, 15, 19, 23, 27, 31	-	Резерв
30:28	AFSEL15	AFSELY задает номер альтернативной функции вывода у порта x (см. раздел 35 «ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ»): <ul style="list-style-type: none"> <li>0: AF0;</li> <li>...</li> <li>7: AF7</li> </ul>
26:24	AFSEL14	
22:20	AFSEL13	
18:16	AFSEL12	
14:12	AFSEL11	
10:8	AFSEL10	
6:4	AFSEL9	
2:0	AFSEL8	

### 9.3.5 Регистр IOCTR\_Px\_PUPD

Исходное состояние:

- PA\_PUPD: 0x37C0 0000;
- PB\_PUPD, PC\_PUPD, PD\_PUPD: 0x0000 0000

Таблица 9.5. Регистр IOCTR\_Px\_PUPDR

Номер бита	Условное обозначение	Назначение
31:30	PUPD15	PUPD <sub>y</sub> задает режим внутренней подтяжки вывода <b>y</b> порта <b>x</b> : <ul style="list-style-type: none"> <li>• 00, 10: нет подтяжки;</li> <li>• 01: подтяжка к земле (pull-down);</li> <li>• 11: подтяжка к питанию (pull-up)</li> </ul>
29:28	PUPD14	
27:26	PUPD13	
25:24	PUPD12	
23:22	PUPD11	
21:20	PUPD10	
19:18	PUPD9	
17:16	PUPD8	
15:14	PUPD7	
13:12	PUPD6	
11:10	PUPD5	
9:8	PUPD4	
7:6	PUPD3	
5:4	PUPD2	
3:2	PUPD1	
1:0	PUPD0	

### 9.3.6 Регистр IOCTR\_Px\_DS

Таблица 9.6. Регистр IOCTR\_Px\_DS

Номер бита	Условное обозначение	Назначение
31:30	DS15	DS <sub>y</sub> задает максимальный выходной ток вывода <b>y</b> порта <b>x</b> . Для портов PA6, PA7, PB9, PB10: <ul style="list-style-type: none"> <li>• 0: 4 мА;</li> <li>• 1: 20 мА.</li> </ul> Для остальных портов: <ul style="list-style-type: none"> <li>• 00: 2 мА;</li> <li>• 01: 4 мА;</li> <li>• 10: 8 мА;</li> <li>• 11: 12 мА</li> </ul>
29:28	DS14	
27:26	DS13	
25:24	DS12	
23:22	DS11	
21:20	DS10	
19:18	DS9	
17:16	DS8	
15:14	DS7	
13:12	DS6	
11:10	DS5	
9:8	DS4	
7:6	DS3	
5:4	DS2	
3:2	DS1	
1:0	DS0	

### 9.3.7 Регистр IOCTR\_Px\_SR

Таблица 9.7. Регистр IOCTR\_Px\_SR

Номер бита	Условное обозначение	Назначение
31:16	-	Резерв
15:0	SR15:SR0	SR <sub>y</sub> задает скорость нарастания сигнала на выводе <b>y</b> порта <b>x</b> для всех портов кроме PA6, PA7, PB9, PB10: <ul style="list-style-type: none"> <li>• 0: fast;</li> <li>• 1: slow.</li> </ul> Для портов PA6, PA7, PB9, PB10 регистр не используется

### 9.3.8 Регистр IOCTR\_Px\_OTYPE

Таблица 9.8. Регистр IOCTR\_Px\_OTYPE

Номер бита	Условное обозначение	Назначение
31:16	-	Резерв
15:0	OT15:OT0	OTy задает тип выходного буфера вывода y порта x: <ul style="list-style-type: none"> <li>0: push-pull;</li> <li>1: open drain</li> </ul>

### 9.3.9 Регистр IOCTR\_Px\_ITYPE

Таблица 9.9. Регистр IOCTR\_Px\_ITYPE

Номер бита	Условное обозначение	Назначение
31:16	-	Резерв
15:0	IT15:IT0	ITy задает тип входного буфера вывода y порта x: <ul style="list-style-type: none"> <li>0: CMOS;</li> <li>1: CMOS Schmitt</li> </ul>

### 9.3.10 Регистры IOCTR\_Px\_I2CMODE

Таблица 9.10. Регистр IOCTR\_PA\_I2CMODE

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:8	-	RO	0	Reserved
7	I2CMODE7	RW	0	Запись в этот бит «1» включает режим I2C для вывода PA7 (I2C0_SCL)
6	I2CMODE6	RW	0	Запись в этот бит «1» включает режим I2C для вывода PA6 (I2C0_SDA)
5:0	-	RO	0	Reserved

Таблица 9.11. Регистр IOCTR\_PB\_I2CMODE

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0	Reserved
10	I2CMODE10	RW	0	Запись в этот бит «1» включает режим I2C для вывода PB10 (I2C1_SCL)
9	I2CMODE9	RW	0	Запись в этот бит «1» включает режим I2C для вывода PB9 (I2C1_SDA)
8:0	-	RO	0	Reserved

## 10. GPIO: БЛОК ВВОДА/ВЫВОДА ОБЩЕГО НАЗНАЧЕНИЯ

### 10.1 Введение

Микросхема содержит четыре блока ввода/вывода общего назначения GPIO0-3, которые подключены в качестве основной функции к портам PA, PB, PC и PD соответственно.

Основные возможности каждого блока:

- 16 выводов с независимой настройкой направления
- программируемая функция прерывания по уровню или фронту независимо для каждого вывода;
- доступ по маске к индивидуальным выводам;
- регистры установки/сброса для многопоточных систем;
- разграничение доступа (secure/non-secure) для каждого вывода;
- отдельные прерывания для secure и non-secure выводов.

### 10.2 Описание функционирования

При доступе к регистрам GPIO проверяется атрибут доверенности транзакции. Этот атрибут используется для разграничения доступа к регистрам GPIO, а также к отдельным выводам блока.

При нарушении доступа устанавливается бит SEC\_INT\_STAT, и, если разрешено регистром GPIOx\_SEC\_INT\_MASK, генерируется прерывание ошибки доступа GPIOx\_ACC. При этом ответ блока GPIO на транзакцию (ошибка или игнорирование) определяется регистром SPCTR\_SECRESPCFG.

Прерывание SEC\_INT\_STAT не устанавливается при нарушении доступа со стороны внешнего отладчика.

Настройка привилегированности доступа выполняется регистрами SPCTR\_APBPPC0\_SP и NSPCTR\_APBPPC0\_NSP для secure и non-secure доступа соответственно.

### 10.3 Регистры GPIOx

Далее приводится описание регистров блоков GPIOx, x = 0, 1, 2, 3.

Доступ к регистрам возможен 8-ми, 16-ти и 32-х разрядными транзакциями.

#### 10.3.1 Перечень регистров GPIOx

Перечень регистров GPIOx приведен в следующей таблице:



Таблица 10.1. Регистры GPIOx

Условное обозначение	Описание	Доступ	Исходное состояние	Смещение
GPIOx_DATA_IN	Регистр входных данных	S, NS	0x00000000	0x000
GPIOx_DATA_OUT	Регистр выходных данных	S, NS	0x00000000	0x004
GPIOx_OUT_EN_SET	Регистр установки сигналов включения выходных буферов	S, NS	0x00000000	0x010
GPIOx_OUT_EN_CLR	Регистр сброса сигналов включения выходных буферов	S, NS	0x00000000	0x014
GPIOx_INT_EN_SET	Регистр включения прерываний	S, NS	0x00000000	0x020
GPIOx_INT_EN_CLR	Регистр выключения прерываний	S, NS	0x00000000	0x024
GPIOx_INT_TYPE_SET	Регистр установки битов типа прерываний	S, NS	0x00000000	0x028
GPIOx_INT_TYPE_CLR	Регистр сброса битов типа прерываний	S, NS	0x00000000	0x02C
GPIOx_INT_POL_SET	Регистр установки бита полярности прерывания	S, NS	0x00000000	0x030
GPIOx_INT_POL_CLR	Регистр сброса бита полярности прерывания	S, NS	0x00000000	0x034
GPIOx_INT_STATUS	Регистр статуса и сброса прерываний	S, NS	0x00000000	0x038
GPIOx_SEC_INT_STAT	Регистр статуса прерывания по нарушению безопасности	S	0x00000000	0x040
GPIOx_SEC_INT_CLR	Регистр сброса прерывания по нарушению безопасности	S	0x00000000	0x044
GPIOx_SEC_INT_MASK	Регистр маски прерывания по нарушению безопасности	S	0x00000000	0x048
GPIOx_SEC_INT_INFO1	Регистр информации 1 о нарушении безопасности	S	0x00000000	0x04C
GPIOx_SEC_INT_INFO2	Регистр информации 2 о нарушении безопасности	S	0x00000000	0x050
GPIOx_SEC_INT_SET	Регистр установки прерывания при нарушении безопасности	S	0x00000000	0x054
GPIOx_PORT_NONSEC_MASK	Регистр битовой маски порта	S	0x00000000	0x058
GPIOx_MASK_LOW_BYTE	Регистры доступа по маске к младшим выводам порта	S, NS	0x00000000	0x400
GPIOx_MASK_HIGH_BYTE	Регистры доступа по маске к старшим выводам порта	S, NS	0x00000000	0x800
GPIOx_PID4	Регистр PID4	S, NS	0x00000004	0xFD0
GPIOx_PID5	Регистр PID5	S, NS	0x00000000	0xFD4
GPIOx_PID6	Регистр PID6	S, NS	0x00000000	0xFD8
GPIOx_PID7	Регистр PID7	S, NS	0x00000000	0xFDC
GPIOx_PID0	Регистр PID0	S, NS	0x00000062	0xFE0
GPIOx_PID1	Регистр PID1	S, NS	0x000000B8	0xFE4
GPIOx_PID2	Регистр PID2	S, NS	0x0000000B	0xFE8
GPIOx_PID3	Регистр PID3	S, NS	0x00000000	0xFEC
GPIOx_CID0	Регистр CID0	S, NS	0x0000000D	0xFF0
GPIOx_CID1	Регистр CID1	S, NS	0x000000F0	0xFF4
GPIOx_CID2	Регистр CID2	S, NS	0x00000005	0xFF8
GPIOx_CID3	Регистр CID3	S, NS	0x000000B1	0xFFC

Примечание: тип доступа «S» обозначает только Secure доступ, «S, NS» - Secure или Non-secure доступ.

### 10.3.2 Регистр GPIOx\_DATA\_IN

Формат регистра GPIOx\_DATA\_IN приведен в следующей таблице.

**Таблица 10.2. Поля регистра GPIOx\_DATA\_IN**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	DATA	RW	0x0	При чтении возвращает значение на входных портах. Считанное значение имеет два такта задержки за счет логики синхронизации. При записи значение DATA помещается в выходной регистр DATA_OUT.

### 10.3.3 Регистр GPIOx\_DATA\_OUT

Формат регистра GPIOx\_DATA\_OUT приведен в следующей таблице.

**Таблица 10.3. Поля регистра GPIOx\_DATA\_OUT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	DATA	RW	0x0	Запись устанавливает значение регистра выходных данных. При чтении возвращается текущее значение регистра.

### 10.3.4 Регистр GPIOx\_OUT\_EN\_SET

Формат регистра GPIOx\_OUT\_EN\_SET приведен в следующей таблице.

**Таблица 10.4. Поля регистра GPIOx\_OUT\_EN\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	OUT_EN_SET	RW	0x0	Запись 1 включает выходной буфер соответствующего вывода. Запись 0 игнорируется. Чтение возвращает направление работы вывода: 0x0: вход; 0x1: выход

### 10.3.5 Регистр GPIOx\_OUT\_EN\_CLR

Формат регистра GPIOx\_OUT\_EN\_CLR приведен в следующей таблице.

**Таблица 10.5. Поля регистра GPIOx\_OUT\_EN\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:0	OUT_EN_CLR	RW	0x0	Запись 1 выключает выходной буфер соответствующего вывода. Запись 0 игнорируется. Чтение возвращает направление работы вывода: 0x0: вход; 0x1: выход

### 10.3.6 Регистр GPIOx\_INT\_EN\_SET

Формат регистра GPIOx\_INT\_EN\_SET приведен в следующей таблице.

**Таблица 10.6. Поля регистра GPIOx\_INT\_EN\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	INT_EN_SET	RW	0x0	Запись 1 разрешает прерывание на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает статус разрешения прерывания: 0x0: прерывание выключено; 0x1: прерывание включено

### 10.3.7 Регистр GPIOx\_INT\_EN\_CLR

Формат регистра GPIOx\_INT\_EN\_CLR приведен в следующей таблице.

**Таблица 10.7. Поля регистра GPIOx\_INT\_EN\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	INT_EN_CLR	RW	0x0	Запись 1 запрещает прерывание на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает статус разрешения прерывания: 0x0: прерывание выключено; 0x1: прерывание включено

### 10.3.8 Регистр GPIOx\_INT\_TYPE\_SET

Формат регистра GPIOx\_INT\_TYPE\_SET приведен в следующей таблице.

**Таблица 10.8. Поля регистра GPIOx\_INT\_TYPE\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:0	INT_TYPE_SET	RW	0x0	Запись 1 устанавливает бит типа прерывания на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает тип прерывания: 0x0: прерывание по уровню сигнала; 0x1: прерывание по фронту сигнала

### 10.3.9 Регистр GPIOx\_INT\_TYPE\_CLR

Формат регистра GPIOx\_INT\_TYPE\_CLR приведен в следующей таблице.

**Таблица 10.9. Поля регистра GPIOx\_INT\_TYPE\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	INT_TYPE_CLR	RW	0x0	Запись 1 сбрасывает бит типа прерывания на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает тип прерывания: 0x0: прерывание по уровню сигнала; 0x1: прерывание по фронту сигнала

### 10.3.10 Регистр GPIOx\_INT\_POL\_SET

Формат регистра GPIOx\_INT\_POL\_SET приведен в следующей таблице.

**Таблица 10.10. Поля регистра GPIOx\_INT\_POL\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	INT_POL_SET	RW	0x0	Запись 1 устанавливает бит полярности прерывания на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает полярность прерывания: 0x0: прерывание по низкому уровню или спадающему фронту сигнала; 0x1: прерывание по высокому уровню или нарастающему фронту сигнала

### 10.3.11 Регистр GPIOx\_INT\_POL\_CLR

Формат регистра GPIOx\_INT\_POL\_CLR приведен в следующей таблице.

**Таблица 10.11. Поля регистра GPIOx\_INT\_POL\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:0	INT_POL_CLR	RW	0x0	Запись 1 сбрасывает бит полярности прерывания на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает полярность прерывания: 0x0: прерывание по низкому уровню или спадающему фронту сигнала; 0x1: прерывание по высокому уровню или нарастающему фронту сигнала

### 10.3.12 Регистр GPIOx\_INT\_STATUS

Формат регистра GPIOx\_INT\_STATUS приведен в следующей таблице.

**Таблица 10.12. Поля регистра GPIOx\_INT\_STATUS**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	STATUS_CLEAR	RW	0x0	Запись 1 сбрасывает прерывание на соответствующем выводе. Запись 0 игнорируется. Чтение возвращает статус прерывания.

### 10.3.13 Регистр GPIOx\_SEC\_INT\_STAT

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_STAT приведен в следующей таблице.

**Таблица 10.13. Поля регистра GPIOx\_SEC\_INT\_STAT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	SEC_INT_STAT	RO	0x0	Статус прерывания по нарушению безопасности.

### 10.3.14 Регистр GPIOx\_SEC\_INT\_CLR

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_CLR приведен в следующей таблице.

**Таблица 10.14. Поля регистра GPIOx\_SEC\_INT\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	SEC_INT_CLR	WO	0x0	Запись 1 сбрасывает статус прерывания по нарушению безопасности. Запись 0 игнорируется.

### 10.3.15 Регистр GPIOx\_SEC\_INT\_MASK

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_MASK приведен в следующей таблице.

**Таблица 10.15. Поля регистра GPIOx\_SEC\_INT\_MASK**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0	Резерв
1	PORT_NONSEC_MASK	RW	0x0	Разрешение прерывания при ошибке доступа по битовой маске (PORT_NONSEC_MASK): 0x0: Нарушение битовой маски порта приводит к установке регистров SEC_INT_STAT и SEC_INT_INFO; 0x1: При нарушении битовой маски порта регистры SEC_INT_STAT и SEC_INT_INFO не устанавливаются
0	SEC_ACC_MASK	RW	0x0	Разрешение установки сигнала прерывания GPIOx_ACC при ошибке доступа к регистрам: 0x0: Прерывание GPIOx_ACC устанавливается, если установлен бит SEC_INT_STAT; 0x1: Прерывание GPIOx_ACC не устанавливается, если установлен бит SEC_INT_STAT

### 10.3.16 Регистр GPIOx\_SEC\_INT\_INFO1

При нарушении безопасности в данный регистр заносится информация о транзакции, вызвавшей это нарушение. Новые значения могут быть записаны только после сброса прерывания.

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_INFO1 приведен в следующей таблице.

**Таблица 10.16. Поля регистра GPIOx\_SEC\_INT\_INFO1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	RW	RO	0x0	Тип транзакции: 0x0: чтение; 0x1: запись
16	NONSEC	RO	0x0	Атрибут безопасности транзакции: 0x0: secure; 0x1: non-secure
15:12	BYTE_STROBE	RO	0x0	Стробы записи
11:0	ADDR	RO	0x0	Адрес обращения

### 10.3.17 Регистр GPIOx\_SEC\_INT\_INFO2

При нарушении безопасности в данный регистр заносится информация о транзакции, вызвавшей это нарушение. Новые значения могут быть записаны только после сброса прерывания.

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_INFO2 приведен в следующей таблице.

**Таблица 10.17. Поля регистра GPIOx\_SEC\_INT\_INFO2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	WDATA	RO	0x0	Записываемые данные

### 10.3.18 Регистр GPIOx\_SEC\_INT\_SET

Регистр для установки прерывания SEC\_INT\_STAT. Предназначен для отладки.

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_SEC\_INT\_SET приведен в следующей таблице.

**Таблица 10.18. Поля регистра GPIOx\_SEC\_INT\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	SET	WO	0x0	Запись 1 устанавливает бит SEC_INT_STAT. Запись 0 игнорируется.

### 10.3.19 Регистр GPIOx\_PORT\_NONSEC\_MASK

Регистр устанавливает уровень доступа к отдельным выводам порта.

Регистр доступен только для secure обращений.

Формат регистра GPIOx\_PORT\_NONSEC\_MASK приведен в следующей таблице.

**Таблица 10.19. Поля регистра GPIOx\_PORT\_NONSEC\_MASK**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:0	NONSEC_MASK	RW	0x0	Каждый бит поля устанавливает уровень доступа к соответствующему выводу порта: 0x0: secure; 0x1: non-secure

### 10.3.20 Регистр GPIOx\_MASK\_LOW\_BYTE

Регистры для доступа по маске к младшим выводам порта. Биты [9:2] адреса используются в качестве маски для разрядов [7:0] данных.

Формат регистра GPIOx\_MASK\_LOW\_BYTE приведен в следующей таблице.

**Таблица 10.20. Поля регистра GPIOx\_MASK\_LOW\_BYTE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:8	-	RO	0	Резерв
7:0	LOWER_ENABLE_MASK	RW	0x0	Маска для младших восьми бит порта.

### 10.3.21 Регистр GPIOx\_MASK\_HIGH\_BYTE

Регистры для доступа по маске к старшим выводам порта. Биты [9:2] адреса используются в качестве маски для разрядов [15:8] данных.

Формат регистра GPIOx\_MASK\_HIGH\_BYTE приведен в следующей таблице.

**Таблица 10.21. Поля регистра GPIOx\_MASK\_HIGH\_BYTE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:8	HIGHER_ENABLE_MASK	RW	0x0	Маска для старших восьми бит порта.
7:0	-	RO	0	Резерв



## 11. UART: КОНТРОЛЛЕР UART

В микросхеме реализованы четыре блока UART.

### 11.1 Основные особенности

- Поддержка длины символа в 9 бит (только для UART0)
- Поддержка интерфейса RS485 (только для UART0)
- Опорный выходной сигнал символьной частоты (только для UART0)
- Режим Auto Flow Control по стандарту 16750 (только для UART0)
- Режим прерывания по Transmitter Holding Register Empty (THRE)
- Программируемые параметры:
  - Длина символа (5-8 бит)
  - Опциональный бит четности
  - Количество стоп бит (1, 1.5 или 2)
- Обнаружение и генерация обрыва на линии
- Аппаратный интерфейс запросов к DMA
- Приоритезация прерываний
- Определение ложного старта бита
- Програмируемая поддержка дробного baud rate
- FIFO передачи и приема по 16 байт каждое
- “Shadow” регистры для быстрой смены контекста
- Функционал busy

### 11.2 Регистры

Перечень регистров uart приведен в следующей таблице:

**Таблица 11.1. Перечень программно-доступных регистров uart0**

Смещение	Условное обозначение	Название регистра	Исходное состояние	Тип доступа
0x0	RBR	Регистр буфера приема	0x0	RO
	THR	Регистр передачи данных	0x0	WO
	DLL	Младший регистр делителя частоты	0x0	RW
0x4	DLH	Старший регистр делителя частоты	0x0	RW
	IER	Регистр разрешения прерывания	0x0	RW
0x8	IIR	Регистр идентификации прерывания	0x0	RO
0xc	LCR	Регистр управления линией	0x0	RW
0x10	MCR	Регистр управления модемом	0x0	RW
0x14	LSR	Регистр состояния линии	0x0	RO
0x18	MSR	Регистр состояния модема	0x0	RO
0x1c	SCR	Регистр общего назначения	0x0	RW
0x20	LPDLL	Делитель скорости передач (Low)	0x0	RW
0x24	LPDLH	Делитель скорости передач (High)	0x0	RW
0x30 + n*0x4	STHR0-15	Теневой регистр на передачу	0x0	RW
0x30	SRBR0	Дублирующий регистр буфера приема	0x0	RO
0x34	SRBR1	Дублирующий регистр буфера приема 1	0x0	RO

0x38	SRBR2	Дублирующий регистр буфера приема 2	0x0	RO
0x3c	SRBR3	Дублирующий регистр буфера приема 3	0x0	RO
0x40	SRBR4	Дублирующий регистр буфера приема 4	0x0	RO
0x44	SRBR5	Дублирующий регистр буфера приема 5	0x0	RO
0x48	SRBR6	Дублирующий регистр буфера приема 6	0x0	RO
0x4c	SRBR7	Дублирующий регистр буфера приема 7	0x0	RO
0x50	SRBR8	Дублирующий регистр буфера приема 8	0x0	RO
0x54	SRBR9	Дублирующий регистр буфера приема 9	0x0	RO
0x58	SRBR10	Дублирующий регистр буфера приема 10	0x0	RO
0x5c	SRBR11	Дублирующий регистр буфера приема 11	0x0	RO
0x60	SRBR12	Дублирующий регистр буфера приема 12	0x0	RO
0x64	SRBR13	Дублирующий регистр буфера приема 13	0x0	RO
0x68	SRBR14	Дублирующий регистр буфера приема 14	0x0	RO
0x6c	SRBR15	Дублирующий регистр буфера приема 15	0x0	RO
0x70	FAR	Регистр доступа FIFO	0x0	RW
0x74	TFR	Регистр чтения FIFO передатчика	0x0	RO
0x78	RFW	Регистр записи FIFO приемника	0x0	RW
0x7c	USR	Регистр статуса UART	0x0	RO
0x80	TFL	Регистр уровня FIFO передатчика	0x0	RO
0x84	RFL	Регистр уровня FIFO приемника	0x0	RO
0x88	SRR	Регистр программного сброса	0x0	RW
0x8c	SRTS	Дублирующий регистр запроса на передачу	0x0	RW
0x90	SBCR	Дублирующий регистр остановки управления	0x0	RW
0x94	SDMAM	Дублирующий регистр режима DMA	0x0	RW
0x98	SFE	Дублирующий регистр активации FIFO	0x0	RW
0x9c	SRT	Дублирующий регистр триггера прерывания приемника	0x0	RW
0xa0	STET	Дублирующий регистр триггера прерывания передатчика	0x0	RW
0xa4	HTX	Регистр остановки передачи	0x0	RW
0xa8	DMASA	Регистр программного оповещения DMA	0x0	RW
0xac	TCR	Регистр управления приемопередатчиком	0x0	RW
0xb0	DE_EN	Регистр управления сигналом DE	0x0	RW
0xb4	RE_EN	Регистр управления сигналом RE	0x0	RW
0xb8	DET	Регистр управления таймером DE	0x0	RW
0xbc	TAT	Регистр управления таймером переключения DE и RE	0x0	RW
0xc0	DLF	Регистр дробной части делителя частоты	0x0	RW
0xc4	RAR	Регистр адреса приема	0x0	RW
0xc8	TAR	Регистр адреса передачи	0x0	RW
0xcc	LCR_EXT	Регистр расширенного управления линией	0x0	RW
0xf4	CPR	Регистр параметров конфигурации UART	0x0	RO
0xf8	UCV	Регистр версии UART	0x0	RO
0xfc	CTR	Регистр UART ID	0x0	RO

### 11.2.1 Описание регистра RBR

Регистр буфера приема.

Значение по сбросу: 0x0

Это регистр доступен, только если DLAB бит (LCR[7]) равен 0x0.

Формат регистра RBR приведен в следующей таблице:

**Таблица 11.2. Поля регистра RBR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_RBR	Резерв	0x0
7:0	RBR	<p>Регистр содержит данные полученные по входному последовательному порту (sin в режиме UART или sir_in в ИК режиме).</p> <p>Данные действительны, если DR бит (LSR[0]) равен 0x1.</p> <p>При выключенном FIFO буфере (FCR[0] равен 0x0), чтение данных из регистра RBR должно происходить до получения следующих данных, иначе данные будут перезаписаны, что приведет к ошибке переполнения.</p> <p>При включенном FIFO буфере (FCR[0] равен 0x1), этот регистр предоставляет доступ к последовательному чтению данных из FIFO буфера приемника. Если FIFO буфер приемника заполнен и чтения данных не произойдет, до получения следующих данных, тогда данные в буфере FIFO не будут перезаписаны, а входящие данные будут потеряны, что, также, вызовет ошибку переполнения.</p>	0x0

### 11.2.2 Описание регистра THR

Регистр передачи данных.

Значение по сбросу: 0x0

Это регистр доступен, только если DLAB бит (LCR[7]) равен 0x0.

Формат регистра THR приведен в следующей таблице:

**Таблица 11.3. Поля регистра THR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_RBR	Резерв	0x0

Диапазон	Название	Описание	Сброс
7:0	THR	<p>Регистр содержит данные для отправления по выходному последовательному порту (sout в режиме UART или sir_out_n в ИК режиме).</p> <p>Данные могут быть записаны только, если THRE бит (LSR[5]) равен 0x1.</p> <p>При выключенном FIFO буфере (FCR[0] равен 0x0) и THRE установлен в 0x1, запись символа в регистр THR приводит к установке THRE в 0x0. Запись до установки THRE в 0x1 приведет к перезаписи данных.</p> <p>При включенном FIFO буфере (FCR[0] равен 0x1) и THRE установлен в 0x1, символы могут быть записаны в этот регистр до заполнения FIFO передатчика, данные записанные в случае заполнения FIFO будут утеряны.</p>	0x0

### 11.2.3 Описание регистра DLL

Младший регистр делителя частоты.

Значение по сбросу: 0x0

Это регистр доступен, только если DLAB бит (LCR[7]) равен 0x1.

Формат регистра DLL приведен в следующей таблице:

**Таблица 11.4. Поля регистра DLL**

Диапазон	Название	Описание	Сброс
31:8	RSVD_RBR	Резерв	0x0
7:0	DLL	<p>Младшие 8 бит регистра делителя частоты. Скорость передачи данных определяется формулой <math>\text{БОД} = \text{Частота UART} / (16 * \text{DL})</math>, где <math>\text{DL} = \{\text{DH}, \text{DL}\}</math></p> <p>В случае если DLL и DLH установлены в 0 передача данных по последовательному порту отключена, после установки значения регистра DLL, должно пройти не меньше 8 периодов тактовой частоты UART</p>	0x0

### 11.2.4 Описание регистра DLH

Старший регистр делителя частоты.

Значение по сбросу: 0x0

Это регистр доступен, только если DLAB бит (LCR[7]) равен 0x1.

Формат регистра DLH приведен в следующей таблице:

**Таблица 11.5. Поля регистра DLH**

Диапазон	Название	Описание	Сброс
31:8	RSVD_RBR	Резерв	0x0
7:0	DLL	Старшие 8 бит регистра делителя частоты. Скорость передачи данных определяется формулой $БОД = Частота\ UART / (16 * DL)$ , где $DL = \{DH, DL\}$ . В случае если DLL и DLH установлены в 0 передача данных по последовательному порту отключена, после установки значения регистра DLH, должно пройти не меньше 8 периодов тактовой частоты UART	0x0

### 11.2.5 Описание регистра IER

Регистр разрешения прерывания

Значение по сбросу: 0x0.

Это регистр доступен, только если DLAB бит (LCR[7]) равен 0x0.

Формат регистра IER приведен в следующей таблице:

**Таблица 11.6. Поля регистра IER**

Диапазон	Название	Описание	Сброс
31:8	RSVD_IER_31to8	Резерв	0x0
7	PTIME	Режим программируемого порога для прерывания. Используется для включения/отключения генерации прерывания THRE.  0x0 – отключен 0x1 – включен	0x0
6:5	RSVD_IER_6to5	Резерв	0x0
4	ELCOLR	Управление методом сброса статуса в регистре LSR. Доступен только в случае ошибок: переполнения, четности, кадрирования также в состоянии остановки прерывания. 0x0 – Поле статуса регистра LSR are cleared или по чтению Rx FIFO или по чтению регистра LSR. 0x1 – Поле статуса сбрасываются только по чтению регистра LSR.	0x0
3	EDSSI	Разрешение прерывания по статусу модема (MSR). Четвертое по приоритету прерываний.  0x1 – разрешено 0x0 – запрещено	0x0

Диапазон	Название	Описание	Сброс
2	ELSI	Разрешение прерывания по состоянию линии приема (LSR). Первое по приоритету прерываний.  0x1 – разрешено 0x0 – запрещено	0x0
1	ETBEI	Разрешение прерывания по регистру THRE. Третье по приоритету прерываний  0x1 – разрешено 0x0 – запрещено	0x0
0	ERBFI	Разрешение прерывания по доступности полученных данных (Enable Received Data Available Interrupt) или, при включенном FIFO, прерывания по тайм-ауту* входных данных (Character Timeout Interrupt). Второе по приоритету прерываний.  0x1 – разрешено 0x0 – запрещено  * – блок детектора тайм-аута – указывает на отсутствие движения символьных данных в FIFO приемника в течение заданного периода времени; это используется для генерации символьных прерываний тайм-аута.	0x0

## 11.2.6 Описание регистра FCR

Регистр управления FIFO

Значение по сбросу: 0x0.

Это регистр доступен, только если реализованы FIFO.

Диапазон	Название	Описание	Сброс
31:8	RSVD_FCR_31to8	Резерв	0x0
7:6	RT	Используется для выбора уровня принимающего FIFO, при котором будет сгенерировано прерывание приема данных. Также определяет, когда будет подан сигнал dma_rx_req_n в определенных режимах работы.  0x0 – 1 символ в FIFO 0x1 – FIFO заполнен на 1/4 0x2 – FIFO заполнен на 1/2 0x3 – FIFO на 2 меньше, чем полный	0x0

Диапазон	Название	Описание	Сброс
5:4	TET	Используется для выбора порогового уровня, при котором будут генерироваться прерывания THRE, когда режим активен. Также определяет, когда будет подан сигнал dma_tx_req_n в определенных режимах работы.  0x0 – FIFO пуст 0x1 – 2 символа в FIFO 0x2 – FIFO заполнен на 1/4 0x3 – FIFO заполнен на 1/2	0x0
3	DMAM	Выбор DMA режимов.  0x0 – Режим 0 0x1 – Режим 1	0x0
2	XFIFOR	Сбрасывает управляющую часть FIFO передачи и обрабатывает FIFO, как пустой.  0x1 – Сброс передающего FIFO	0x0
1	RFIFOR	Сбрасывает управляющую часть принимающего FIFO и обрабатывает FIFO, как пустой.  0x1 – Сброс принимающего FIFO	0x0
0	FIFOE	Включает/выключает FIFO передачи (XMIT) и приема (RCVR). При изменении значения этого бита, XMIT и RCVR сбрасываются.  0x0 – FIFO отключен 0x1 – FIFO включен	0x0

### 11.2.7 Описание регистра IIR

Регистр идентификации прерывания

Формат регистра IIR приведен в следующей таблице:

**Таблица 11.7. Поля регистра IIR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_IIR_31to8	Резерв	0x0
7:6	RT	Используется для указания, включены или отключены FIFO.  0x0 – FIFO отключен 0x1 – FIFO включен	0x0
5:4	RSVD_IIR_5to4	Резерв	0x0

Диапазон	Название	Описание	Сброс
3:0	IID	<p>ID прерывания (IID). Отображает прерывание с наивысшим приоритетом</p> <p>0x0 – статус модема</p> <p>0x1 – прерывания не ожидаются</p> <p>0x2 – прерывание THRE</p> <p>0x4 – доступны данные на принимающей линии</p> <p>0x6 – статус линии приемника</p> <p>0x7 – контроллер занят</p> <p>0xc – таймаут ожидания приемника</p>	0x1

### 11.2.8 Описание регистра LCR

Регистр управления линией

Формат регистра LCR приведен в следующей таблице:

**Таблица 11.8. Поля регистра LCR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_LCR_31to8	Резерв	0x0
7	DLAB	<p>Бит доступа к делителю частоты, используется для разрешения чтения и записи регистров DLL/LPDLL и LPDLH. Данный бит должен быть установлен в 0x0 после настройки частоты интерфейса UART, для доступа к другим регистрам. Доступен только для записи в случае если USR[0] == 0x0</p> <p>0x0 – отключен</p> <p>0x1 – включен</p>	0x0
6	BC	<p>Бит обрыва линии. Используется для снятия нагрузки, связанной с выполнением операции чтения и записи в LCR. Это используется для передачи состояния прерывания на принимающее устройство.</p> <p>0x0 – нормальная работа</p> <p>1x1 – на выходе <code>sout</code> устанавливается низкий уровень. В ИК режиме линия <code>sir_out_n</code> продолжительно пульсирует. В режиме кольцевой связи линия <code>sir_out_n</code> в состоянии низкого уровня.</p>	0x0



Диапазон	Название	Описание	Сброс
5	SP	Бит установки паритета. Доступен для записи, когда USR[0] не 0, используется для установки паритета.  0x0 – SP отключен 0x1 - Когда PEN, EPS установлены в 1, бит паритета передается и принимается как логический 0. Если PEN установлен в 1, а EPS в 0, то бит паритета передается и принимается как логическая 1.	0x0
4	EPS	Бит четности паритета  Доступен для записи, когда USR[0] не 0, используется для установки четности паритета.  0x0 – нечетность 0x1 – четность	0x0
3	PEN	Разрешение паритета Доступен для записи, когда USR[0] не 0, используется для разрешения приема, проверки и передачи контрольного бита (бита паритета).  0x0 – отключить паритет 0x1 – включить паритет	0x0
2	STOP	Количество стоп-битов Доступен для записи, когда USR[0] не 0.  0x0 – 1 стоп-бит, 0x1 – 1.5 стоп-бита, если LCR[0:1]=0, иначе 2 стоп-бита.  Приемник анализирует только первый стоп бит.	0x0
1:0	DLS	Количество бит данных в передаваемом символе. Доступен для записи, когда USR[0] не 0.  0x00 – 5 бит 0x01 – 6 бит 0x10 – 7 бит 0x11 – 8 бит	0x0

### 11.2.9 Описание регистра MCR

Регистр управления модемом

Формат регистра MCR приведен в следующей таблице:

**Таблица 11.9. Поля регистра MCR**

Диапазон	Название	Описание	Сброс
31:7	RSVD_MCR_31to7	Резерв	0x0

Диапазон	Название	Описание	Сброс
6	SIRE	Активация ИК режима  0x0 – ИК режим отключен 0x1 – ИК режим включен	0x0
5	AFCE	Включение автоматического контроля управления потоком возможно только при включенных FIFO (FCR[0]=1) (см. раздел 11.3.6).  0x0 – Автоматическое управление потоком выключено. 0x1 – Автоматическое управление потоком включено.	0x0
4	LoopBack	Активация петлевого режима Используется для активации диагностического режима с целью тестирования.  При отключенном ИК режиме, линия sout, будут находиться в высоком состоянии, и подключена на линию sin, до отключения петлевого режима. В этом режиме функционируют все прерывания. Также, все модемные и программируемые выходы будут отключены и закольцованы на модемные входы.  В ИК режиме линия sir_out_n будет находиться в низком уровне и закольцована на линию sir_in.	0x0
3	OUT2	Программируемый выход 2  Значение, записанное в поле, инвертируется и передается на выход (out2_n)	0x0
2	OUT1	Программируемый выход 1  Значение, записанное в поле, инвертируется и передается на выход (out1_n)	0x0

Диапазон	Название	Описание	Сброс
1	RTS	<p>Запрос на отправление данных. Этот бит используется для управления выводом RTS_N. Вывод RTS_N используется для информирования модема окончного устройства о том, что UART готов к обмену данными.</p> <p>При выключенном режиме автоматического управления потоком (MCR[5] = 0), сигнал RTS_N переводится в низкий уровень записью 1 в бит RTS, и переводится в высокий уровень записью 0 в бит RTS. При включенном режиме автоматического управления потоком (MCR[5]=1), и включенных FIFO (FCR[0]=1), выходной сигнал RTS_N может управляться таким же способом, но он также меняет свое значение в зависимости от уровня заполнения FIFO приемника (становится равным 1, если кол-во записей в FIFO приемника превышает пороговое значение заполнения FIFO приемника, и переходит обратно в 0, когда кол-во записей в FIFO становится меньше этого значения).</p>	0x0
0	DTR	<p>Управление линией dtr_n Используется для прямого управления сигналом dtr_n. Записанное значение инвертируется и выводится на dtr_n. Сигнал DTR используется для оповещения модема, о готовности абонента установить соединение.</p> <p>0x0 – dtr_n не активирован (логическая 1) 0x1 – dtr_n активирован (логический 0)</p>	0x0

## 11.2.10 Описание регистра LSR

Регистр состояния линии

Формат регистра LSR приведен в следующей таблице:

**Таблица 11.10. Поля регистра LSR**

Диапазон	Название	Описание	Сброс
31:9	RSVD_LSR_31to9	Резерв	0x0
8	RSVD_ADDR_RCVD	Резерв	0x0
7	RFE	<p>Бит ошибки приемника</p> <p>Определяет наличие хотя бы одной ошибки четности, кадрирования или состояние обрыва (LSR[4]).</p> <p>Сбрасывается при чтении LSR.</p>	0x0

Диапазон	Название	Описание	Сброс
6	TEMT	Бит отсутствия передаваемых данных  Устанавливается в 1 в случае отсутствия данных в FIFO буфере и сдвиговом регистре передатчика.	0x1
5	THRE	Бит отсутствия данных в буфере передатчика  Показывает, что UART готов принять следующий символ для передачи. Устанавливается, когда содержимое регистра THR передается в передающий сдвигающий регистр. Одновременно с этим генерируется прерывание THREI, если оно разрешено. Бит сбрасывается при записи символа в регистр THR.	0x1
4	BI	Обрыв линии  Устанавливается, если вход приема данных находится в состоянии 0 не менее чем время передачи всего символа. В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. При возникновении этой ситуации, в FIFO загружается только один нулевой символ. Прием следующих символов разрешается после того, как вход приема данных перейдет в единицу и будет принят действительный стартовый бит. Бит сбрасывается при чтении содержимого регистра LSR.	0x0
3	FE	Ошибка кадрирования.  Устанавливается, если стоп-бит равен нулю. В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. После этой ошибки UART пересинхронизируется. Бит сбрасывается при чтении содержимого регистра LSR.	0x0
2	PE	Ошибка паритета (контрольного бита)  В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. Бит сбрасывается при чтении содержимого регистра LSR.	0x0

Диапазон	Название	Описание	Сброс
1	OE	Ошибка переполнения  Устанавливается, если содержимое регистра RBR не было прочитано, в сдвигающий регистр принят следующий символ и начат прием очередного символа. При этом новый символ записывается в сдвигающий регистр вместо старого. В режиме FIFO устанавливается, если после перехода порогового уровня FIFO заполнено до конца, во входной сдвигающий регистр полностью принят следующий символ и начат прием очередного символа. При этом в FIFO ничего не передается. Бит сбрасывается при чтении содержимого регистра LSR.	0x0
0	DR	Бит готовности данных Показывает наличие данных в приемнике, которые еще не были прочитаны.	0x0

### 11.2.11 Описание регистра MSR

Регистр состояния модема

При записи в биты 0, 1, 2 или 3 логической 1, для отображения изменения настроек модема, будет сгенерирован прерывание по статусу модема (если он разрешен), если оно разрешено через IER, независимо от того когда произошла запись. Биты (бит 0, 1, 3) могут быть установлены после сброса, даже если их соответствующие модемные сигналы неактивны, поскольку синхронизированная версия модемных сигналов имеет значение сброса 0 и изменяется на значение 1 после сброса. Для предотвращения нежелательных прерываний, чтение MSR можно производить после сброса.

Формат регистра MSR приведен в следующей таблице:

**Таблица 11.11. Поля регистра MSR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_MSR_31to8	Резерв	0x0
7	DCD	Обнаружение носителя данных  Используется для чтения состояния линии dcd_n. dcd_n указывает на то, что носитель был обнаружен модемом или набором данных.	0x0
6	RI	Индикатор доступа  Используется для чтения состояния линии ri_n. ri_n указывает на то, что доступ был получен модемом или набором данных.	0x0

Диапазон	Название	Описание	Сброс
5	DSR	Готовность данных  Используется для чтения состояния линии dsr_n. dsr_n указывает на то, что модем или набор данных готов установить связь с UART.	0x0
4	CTS	Готовность отправки  Используется для чтения состояния линии cts_n. cts_n указывает на то, что модем или набор данных готов к обмену данными с UART.	0x0
3	DDCD	Обнаружение изменения линии dcd_n  Определяет изменялась ли линия dcd_n после последнего чтения регистра MSR.	0x0
2	TERI	Обнаружение изменения линии ri_n  Определяет изменялась ли линия ri_n после последнего чтения регистра MSR.	0x0
1	DDSR	Обнаружение изменения линии dsr_n  Определяет изменялась ли линия dsr_n после последнего чтения регистра MSR.	0x0
0	DCTS	Обнаружение изменения линии cts_n  Определяет изменялась ли линия cts_n после последнего чтения регистра MSR.	0x0

### 11.2.12 Описание регистра SCR

Регистр общего назначения

Формат регистра SCR приведен в следующей таблице:

**Таблица 11.12. Поля регистра SCR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_SCR_31to8	Резерв	0x0
7:0	SCR	Этот регистр предназначен для временного хранения данных. Не имеет определенной цели в управлении контроллером.	0x0

### 11.2.13 Описание регистра SRBR0 - 15

Дублирующий регистр буфера приема

Формат регистра SRBR0 приведен в следующей таблице:

**Таблица 11.13. Поля регистров SRBR0 - 15**

Диапазон	Название	Описание	Сброс
31:8	RSVD_SRBRn	SRBR0 31 to SRBRN_REG_SIZE Reserved bits read read as 0.	0x0
7:0	SRBRn	Этот регистр дублирует регистр RBR. К нему можно обратиться по одному из 32-х адресов.	0x0

### 11.2.14 Описание регистра FAR

Регистр доступа FIFO

Формат регистра FAR приведен в следующей таблице:

**Таблица 11.14. Поля регистра FAR**

Диапазон	Название	Описание	Сброс
31:1	RSVD_FAR_31to1	Резерв	0x0
0	FAR	Регистр доступа FIFO  Позволяет разрешить прямой доступ к FIFO буферам. Регистр используется для включения режима доступа к FIFO для тестирования, так что принимающее FIFO может быть записан мастером, а передающее FIFO может быть прочитан мастером, когда FIFO реализованы и включены. Когда FIFO не реализованы или не включены, это позволяет мастеру записывать RBR и считывать THR.  0x0 – режим доступа FIFO отключен 1x1 – режим доступа FIFO включен	0x0

### 11.2.15 Описание регистра TFR

Регистр чтения FIFO передатчика

Формат регистра TFR приведен в следующей таблице:

**Таблица 11.15. Поля регистра TFR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_TFR_31to8	TFR 31to8 Reserved bits read as 0.	0x0
7:0	TFR	Чтение данных FIFO передатчика  Доступен в режиме доступа к FIFO (FAR[0] = 1).	0x0

## 11.2.16 Описание регистра RFW

Регистр записи FIFO приемника

Формат регистра RFW приведен в следующей таблице:

**Таблица 11.16. Поля регистра RFW**

Диапазон	Название	Описание	Сброс
31:10	RSVD_RFW_31to10	Резерв	0x0
9	RFPE	<p>Запись ошибки кадрирования</p> <p>Если FIFO реализованы и включены, этот бит используется для записи информации об обнаружении ошибок кадрирования принимающего FIFO. Если FIFO не реализованы или не включены, этот бит используется для записи информации об обнаружении ошибок кадрирования в RBR.</p> <p>0x0 – ошибка кадра отключена 1x1 – ошибка кадра включена</p>	0x0
8	RFPE	<p>Запись ошибки четности</p> <p>Если FIFO реализованы и включены, этот бит используется для записи информации об обнаружении ошибок четности принимающего FIFO. Если FIFO не реализованы или не включены, этот бит используется для записи информации об обнаружении ошибок четности в RBR.</p> <p>0x0 – ошибка четности отключена 1x1 – ошибка четности включена</p>	0x0
7:0	RFWD	<p>Запись в FIFO приемника</p> <p>Доступен в режиме доступа к FIFO (FAR[0] = 1).</p>	0x0

## 11.2.17 Описание регистра USR

Регистр статуса UART

Формат регистра USR приведен в следующей таблице:

**Таблица 11.17. Поля регистра USR**

Диапазон	Название	Описание	Сброс
31:5	RSVD_USR_31to5	Резерв	0x0
4	RFF	<p>FIFO приемника полон</p> <p>0x0 – FIFO приемника не полон 1x1 – FIFO приемника полон</p>	0x0



Диапазон	Название	Описание	Сброс
3	RFNE	FIFO приемника не пуст  0x0 – FIFO приемника не пуст 0x1 – FIFO приемника пуст	0x0
2	TFE	FIFO передатчика пуст  0x0 – FIFO передатчика не пуст 0x1 – FIFO передатчика пуст	0x1
1	TFNF	FIFO передатчика не полон  0x0 – FIFO передатчика не полон 0x1 – FIFO передатчика полон	0x1
0	BUSY	UART занят. Этот бит установлен в 1, в следующих случаях:  1. В настоящее время происходит передача данных по последовательному интерфейсу. 2. Регистр THR содержит данные на передачу, при не нулевом значении делителя частоты ( $\{DLH, DLL\} \sim 0$ ) и $LCR.DLAB = 0$ . 3. В настоящее время происходит прием данных по последовательному интерфейсу. 4. Регистр RBR содержит данные на прием.	0x0

### 11.2.18 Описание регистра TFL

Регистр уровня FIFO передатчика

Формат регистра TFL приведен в следующей таблице:

**Таблица 11.18. Поля регистра TFL**

Диапазон	Название	Описание	Сброс
31:6	RSVD_TFL_31toADDR_WIDTH	Резерв	0x0
5:0	tfl	Уровень FIFO передатчика  Отображает количество слов данных в FIFO передатчика.	0x0

### 11.2.19 Описание регистра RFL

Регистр уровня FIFO приемника

**Таблица 11.19. Поля регистра RFL**

Диапазон	Название	Описание	Сброс
31:6	RSVD_RFL_31toADDR_WIDTH	Резерв	0x0

Диапазон	Название	Описание	Сброс
5:0	rfl	Уровень FIFO приемника  Отображает количество слов данных в FIFO приемника.	0x0

## 11.2.20 Описание регистра SRR

Регистр программного сброса

Формат регистра SRR приведен в следующей таблице:

**Таблица 11.20. Поля регистра SRR**

Диапазон	Название	Описание	Сброс
31:3	RSVD_SRR_31to3	Резерв	0x0
2	XFR	Является теньвым регистром FCR[2]. Используется, для снятия нагрузки с программного обеспечения, связанной с сохранением ранее записанных значений FCR только для сброса передающего FIFO. Сбрасывает управляющую часть передающего FIFO и рассматривает FIFO как пустой.	0x0
1	RFR	Является теньвым регистром FCR[1]. Используется, для снятия нагрузки с программного обеспечения, связанной с сохранением ранее записанных значений FCR только для сброса повторного FIFO. Сбрасывает управляющую часть приемного FIFO и рассматривает FIFO как пустой.	0x0
0	UR	Программный сброс UART  Асинхронно сбрасывает UART и синхронно удаляет утверждение сброса. Для реализации с двумя тактовыми сигналами будут сброшены оба домена pclk и sclk.	0x0

## 11.2.21 Описание регистра SRTS

Дублирующий регистр запроса на передачу

Формат регистра SRTS приведен в следующей таблице:

**Таблица 11.21. Поля регистра SRTS**

Диапазон	Название	Описание	Сброс
31:1	RSVD_SRTS_31to1	Резерв	0x0
0	SRTS	Дублирует MCR[1]	0x0

### 11.2.22 Описание регистра SBCR

Дублирующий регистр остановки управления

Формат регистра SBCR приведен в следующей таблице:

**Таблица 11.22. Поля регистра SBCR**

Диапазон	Название	Описание	Сброс
31:1	RSVD_SBCR_31to1	Резерв	0x0
0	SBCB	Дублирует LCR[6]	0x0

### 11.2.23 Описание регистра SDMA

Дублирующий регистр режима DMA

Формат регистра SDMA приведен в следующей таблице:

**Таблица 11.23. Поля регистра SDMA**

Диапазон	Название	Описание	Сброс
31:1	RSVD_SDMA_31to1	Резерв	0x0
0	SDMAM	Дублирует FCR[3]	0x0

### 11.2.24 Описание регистра SFE

Дублирующий регистр активации FIFO

Формат регистра SFE приведен в следующей таблице:

**Таблица 11.24. Поля регистра SFE**

Диапазон	Название	Описание	Сброс
31:1	RSVD_SFE_31to1	Резерв	0x0
0	SFE	Дублирует FCR[0]	0x0

### 11.2.25 Описание регистра SRT

Дублирующий регистр триггера прерывания приемника

Формат регистра SRT приведен в следующей таблице:

**Таблица 11.25. Поля регистра SRT**

Диапазон	Название	Описание	Сброс
31:2	RSVD_SRT_31to2	Резерв	0x0
1:0	SRT	Дублирует FCR[7:6]	0x0

### 11.2.26 Описание регистра STET

Дублирующий регистр триггера прерывания передатчика

Формат регистра STET приведен в следующей таблице:

**Таблица 11.26. Поля регистра STET**

Диапазон	Название	Описание	Сброс
31:2	RSVD_STET_31to2	Резерв	0x0
1:0	STET	Дублирует FCR[5:4]	0x0

### 11.2.27 Описание регистра HTX

Регистр остановки передачи

Формат регистра HTX приведен в следующей таблице:

**Таблица 11.27. Поля регистра HTX**

Диапазон	Название	Описание	Сброс
31:1	RSVD_HTX_31to1	Резерв	0x0
0	HTX	Остановка передачи  Используется для остановки передачи в целях тестирования, таким образом, что буфер FIFO может быть заполнен. В случае, если FIFO режим отключен запись не в этот регистр не повлияет на работу UART.	0x0

### 11.2.28 Описание регистра DMASA

Регистр программного оповещения DMA

Формат регистра DMASA приведен в следующей таблице:

**Таблица 11.28. Поля регистра DMASA**

Диапазон	Название	Описание	Сброс
31:1	RSVD_DMASA_31to1	Резерв	0x0
0	DMASA	Программное оповещения DMA  Используется, если передача должна быть прекращена из-за состояния ошибки. Бит самосбрасывается и не требует ручного сброса.	0x0

## 11.2.29 Описание регистра TCR

Регистр управления приемопередатчиком

Этот регистр используется, для включения/отключения режима RS485, также для управления полярностью сигналов de и re.

Формат регистра TCR приведен в следующей таблице:

**Таблица 11.29. Поля регистра TCR**

Диапазон	Название	Описание	Сброс
31:5	RSVD_TCR_31to5	Резерв	0x0
4:3	XFER_MODE	Режим передачи  0x0 – одновременная передача и прием  0x1 – DE и RE взаимно исключающие сигналы, ожидается, что при программировании, только один из этих сигналов будет в активном состоянии. Во время переключения, будет задействован регистр TAT для определения задержек.  0x2 – DE и RE взаимно исключающие сигналы, изначально находится в состоянии приема (зависит от состояний DE_EN/RE_EN) при появлении данных в передающем буфере, UART ожидает окончания приема, после чего переключается на передачу данных, до тех пока буфер передачи не будет пуст.	0x0
2	DE_POL	Полярность сигнала DE  0x0 – активное высокое состояние 0x1 – активное низкое состояние	0x0
1	RE_POL	Полярность сигнала RE  0x0 – активное высокое состояние 0x1 – активное низкое состояние	0x0
0	RS485_EN	Режим RS485  0x0 – отключен, все поля этого регистра, также регистры DE_EN, RE_EN, TAT являются зарезервированными (не используются). 0x1 – включен все регистры для настройки режима RS485 находятся в рабочем режиме.	0x0

### 11.2.30 Описание регистра DE\_EN

Регистр управления сигналом DE

Формат регистра DE\_EN приведен в следующей таблице:

**Таблица 11.30. Поля регистра DE\_EN**

Диапазон	Название	Описание	Сброс
31:1	RSVD_DE_EN_31to1	Резерв	0x0
0	DE_Enable	Регистр управления сигналом DE 0x0 – сбросить 0x1 – установить	0x0

### 11.2.31 Описание регистра RE\_EN

Регистр управления сигналом RE

Формат регистра RE\_EN приведен в следующей таблице:

**Таблица 11.31. Поля регистра RE\_EN**

Диапазон	Название	Описание	Сброс
31:1	RSVD_RE_EN_31to1	Резерв	0x0
0	RE_Enable	Регистр управления сигналом RE 0x0 – сбросить 0x1 – установить	0x0

### 11.2.32 Описание регистра DET

Регистр управления таймером DE

Формат регистра DET приведен в следующей таблице:

**Таблица 11.32. Поля регистра DET**

Диапазон	Название	Описание	Сброс
31:24	RSVD_DE_DEAT_31to24	Резерв	0x0
23:16	DE_De-assertion_Time	Управления таймером сброса DE Это поле определяет количество периодов, тактового сигнала между окончанием приема данных и сброса сигнала DE.	0x0
15:8	RSVD_DE_AT_15to8	Резерв	0x0
7:0	DE_Assertion_Time	Управления таймером установки DE Это поле определяет количество периодов, тактового сигнала между установкой сигнала DE и началом передачи данных.	0x0

### 11.2.33 Описание регистра TAT

Регистр управления таймером переключения DE и RE

Формат регистра TAT приведен в следующей таблице:

**Таблица 11.33. Поля регистра TAT**

Диапазон	Название	Описание	Сброс
31:16	RE_to_DE	Задержка переключения с RE на DE  Если DET=0, тогда задержка устанавливается в виде запрограммированного значения + 3. Если DET=1, тогда задержка устанавливается в виде запрограммированного значения + 2. Если DET>1, тогда задержка устанавливается в виде запрограммированного значения + 1.	0x0
15:0	DE_to_RE	Задержка переключения с RE на DE  Задержка устанавливается в виде запрограммированного значения + 1.	0x0

### 11.2.34 Описание регистра DLF

Регистр дробной части делителя частоты

Формат регистра DLF приведен в следующей таблице:

**Таблица 11.34. Поля регистра DLF**

Диапазон	Название	Описание	Сброс
31:4	RSVD_DLF	Резерв	0x0
3:0	DLF	Дробная часть делителя частоты  Определяется как $DLF \cdot (2^{DLF\_SIZE})$ DLF_SIZE – настраиваемый параметр, используется для выбора ширины регистра, в котором хранится дробная часть делителя. Значение по сбросу: 0x0	0x0

### 11.2.35 Описание регистра CPR

Регистр параметров конфигурации UART

Формат регистра CPR приведен в следующей таблице:

**Таблица 11.35. Поля регистра CPR**

Диапазон	Название	Описание	Сброс
31:24	RSVD_CPR_31to24	Резерв	0x0

Диапазон	Название	Описание	Сброс
23:16	FIFO_MODE	Encoding of FIFO_MODE configuration parameter value.  0x0 – нет поддержки FIFO 0x1 – глубина FIFO равняется 16 0x2 – глубина FIFO равняется 32 0x4 – глубина FIFO равняется 64 0x8 – глубина FIFO равняется 128 0x10 – глубина FIFO равняется 256 0x20 – глубина FIFO равняется 512 0x40 – глубина FIFO равняется 1024 0x80 – глубина FIFO равняется 2048	0x2
15:14	RSVD_CPR_15to14	Резерв	0x0
13	DMA_EXTRA	Encoding of DMA_EXTRA configuration parameter value.  0x0 – DMA_EXTRA отключен 0x1 – DMA_EXTRA включен	0x1
12	UART_ADD_ENCODED_PARAMS	Encoding of UART_ADD_ENCODED_PARAMS configuration parameter value.  0x0 – UART_ADD_ENCODED_PARAMS отключен 0x1 – UART_ADD_ENCODED_PARAMS включен	0x1
11	SHADOW	Encoding of SHADOW configuration parameter value.  0x0 – SHADOW отключен 0x1 – SHADOW включен	0x1
10	FIFO_STAT	Encoding of FIFO_STAT configuration parameter value.  0x0 – FIFO_STAT отключен 0x1 – FIFO_STAT включен	0x1
9	FIFO_ACCESS	Encoding of FIFO_ACCESS configuration parameter value.  0x0 – FIFO_ACCESS отключен 0x1 – FIFO_ACCESS включен	0x1
8	ADDITIONAL_FEAT	Encoding of ADDITIONAL_FEATURES configuration parameter value.  0x0 – Дополнительные функции отключены 0x1 – Дополнительные функции включены	0x1



Диапазон	Название	Описание	Сброс
7	SIR_LP_MODE	Encoding of SIR_LP_MODE configuration parameter value.  0x0 – режим SIR_LP отключен 0x1 – режим SIR_LP включен	0x0
6	SIR_MODE	Encoding of SIR_MODE configuration parameter value.  0x0 – режим SIR отключен 0x1 – режим SIR включен	0x0
5	THRE_MODE	Encoding of THRE_MODE configuration parameter value.  0x0 – режим THRE отключен 0x1 – режим THRE включен	0x1
4	AFCE_MODE	Encoding of AFCE_MODE configuration parameter value.  0x0 – режим AFCE отключен 0x1 – режим AFCE включен	0x1
3:2	RSVD_CPR_3to2	Резерв	0x0
1:0	APB_DATA_WIDTH	Encoding of APB_DATA_WIDTH configuration parameter value.  0x0 – ширина данных APB 8 бит 0x1 – ширина данных APB 16 бит 0x2 – ширина данных APB 32 бит	0x2

### 11.2.36 Описание регистра UCV

Регистр версии UART

Формат регистра UCV приведен в следующей таблице:

**Таблица 11.36. Поля регистра UCV**

Диапазон	Название	Описание	Сброс
31:0	UART_Component_Version	ASCII значение Для каждого номера версии следует знак *. Например: 32_30_31_2A представляется в виде 2.01*.	0x3430312a

### 11.2.37 Описание регистра CTR

Регистр UART ID

Формат регистра CTR приведен в следующей таблице:

**Таблица 11.37. Поля регистра CTR**

Диапазон	Название	Описание	Сброс
31:0	Peripheral_ID	Периферийное ID	0x44570110

**11.2.38 Описание регистра LPDLL**

Делитель скорости передач (Low)

Значение по сбросу: 0x0.

Регистр действителен только тогда, когда UART настроен для реализации возможностей приема с низким энергопотреблением SIR. Если возможности приема с низким энергопотреблением SIR не реализованы, этот регистр не существует, и чтение из этого адреса регистра возвращает 0.

Формат регистра LPDLL приведен в следующей таблице:

**Таблица 11.38. Поля регистра LPDLL**

Диапазон	Название	Описание	Сброс
31:8	RSVD_LPDLL_31to8	Резерв	0x0
7:0	LPDLL	Регистр составляет младшие 8 бит LPDL. Содержит делитель скорости передачи для UART, который должен обеспечивать скорость передачи 115,2К. Это требуется для обнаружения малой мощности SIR (минимальная ширина импульса) на приемнике.	0x0

**11.2.39 Описание регистра LPDLH**

Делитель скорости передач (High)

Значение по сбросу: 0x0.

Регистр действителен только тогда, когда UART настроен для реализации возможностей приема с низким энергопотреблением SIR. Если возможности приема с низким энергопотреблением SIR не реализованы, этот регистр не существует, и чтение из этого адреса регистра возвращает 0.

Формат регистра LPDLH приведен в следующей таблице:

**Таблица 11.39. Поля регистра LPDLH**

Диапазон	Название	Описание	Сброс
31:8	RSVD_LPDH_31to8	Резерв	0x0

Диапазон	Название	Описание	Сброс
7:0	LPDLH	Регистр составляет старшие 8 бит LPDL. Содержит делитель скорости передачи для UART, который должен обеспечивать скорость передачи 115,2К. Это требуется для обнаружения малой мощности SIR (минимальная ширина импульса) на приемнике.	0x0

### 11.2.40 Описание регистра STHR0 – 15

Теневой регистр на передачу

Значение по сбросу: 0x0.

Этот регистр действителен только тогда, когда UART настроен для реализации дополнительных теневых регистров. Если теневые регистры не реализованы, этот регистр не существует, и чтение из этого адреса регистра возвращает 0.

Формат регистра STHR приведен в следующей таблице:

**Таблица 11.40. Поля регистра STHR0 – 15**

Диапазон	Название	Описание	Сброс
31:y	RSVD_STHRn	Резерв	0x0
x:0	STHRn	Теневой регистр THR обеспечивает пакетный доступ от мастера. Содержит данные, которые должны передаваться на sout в режиме UART или на sir_out_n в инфракрасном режиме. Данные следует записывать в THR только тогда, когда установлен бит THR Empty.	0x0

### 11.2.41 Описание регистра RAR

Приемный регистр адреса

Значение по сбросу: 0x0.

Формат регистра RAR приведен в следующей таблице:

**Таблица 11.41. Поля регистра RAR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_RAR_31to8	Резерв	0x0

Диапазон	Название	Описание	Сброс
7:0	RAR	Регистр согласования адресов в режиме приема. Если во входящем символе установлен 9-й бит, то оставшиеся 8 битов будут проверяться на соответствие этому значению регистра. Если совпадение происходит, то последующие символы с 9-м битом, установленным в 0, будут обрабатываться как байт данных до тех пор, пока не будет получен следующий байт адреса.	0x0

### 11.2.42 Описание регистра TAR

Передающий регистр адреса

Значение по сбросу: 0x0.

Формат регистра TAR приведен в следующей таблице:

**Таблица 11.42. Поля регистра TAR**

Диапазон	Название	Описание	Сброс
31:8	RSVD_TAR_31to8	Резерв	0x0
7:0	TAR	Регистр согласования адресов в режиме передачи. Если бит DLS_E (LCR_EXT [0]) равен 1, то UART отправит 9-битный символ с 9-м битом, установленным в 1, а оставшийся 8-битный адрес будет отправлен из этого регистра.	0x0

### 11.2.43 Описание регистра LCR\_EXT

Регистр расширенного управления линией

Значение по сбросу: 0x0.

Формат регистра LCR\_EXT приведен в следующей таблице:

**Таблица 11.43. Поля регистра LCR\_EXT**

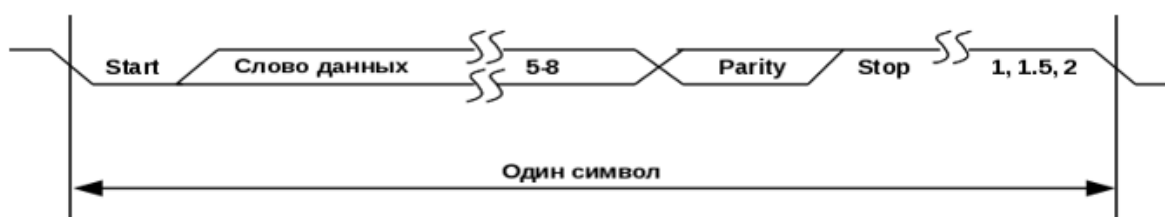
Диапазон	Название	Описание	Сброс
31:4	RSVD_LCR_EXT	Резерв	0x0
3	TRANSMIT_MODE	Бит управления режимом передачи.  1 – THR и STHR имеют ширину 9 бит. Адрес: 9 бит установлен в 1 Данные: 9 бит установлен в 0 0 – THR и STHR имеют ширину 8 бит.	0x0

Диапазон	Название	Описание	Сброс
2	SEND_ADDR	Бит управления отправкой адреса.  1 – 9-битный символ будет передан с 9 битом, установленным в 1, а остальные 8 бит будут соответствовать тому, что запрограммировано в TAR 0 – 9-битный символ будет передан с 9 битом, установленным в 0, а остальные 8 бит будут взяты из TXFIFO, который программируется через регистр THR/STHR шириной 8 бит.	0x0
1	ADDR_MATCH	Бит используется для включения функции совпадения адресов во время приема.  1 – режим совпадения адресов 0 – нормальный режим	0x0
0	DLS_E	Расширение для DLS. Бит используется для включения 9-битных данных при передаче и приеме.	0x0

## 11.3 Функционирование порта

### 11.3.1 Последовательный протокол (RS232)

Так как последовательный обмен между контроллером UART и конечным устройством является асинхронным, то к последовательным данным добавляются дополнительные биты Start и Stop, обозначающие начало и конец передачи. Совокупность слова данных и битов Start и Stop называется символом. Структура символа данных отображена на Рисунок 11.1.



**Рисунок 11.1. RS232. Формат символа данных**

Также к символу может быть добавлен бит контроля четности (Parity). Этот бит появляется после последнего бита данных и перед Stop битом, чтобы предоставить UART возможность выполнять простую проверку ошибок в полученных данных.

Управление параметрами символа последовательного обмена происходит через регистр LCR. Отдельные биты слова данных отправляются после Start бита, начиная с младшего значащего бита (LSB). За ними следует дополнительный бит четности, за которым следуют Stop биты, которые могут быть 1, 1.5, 2.

Все биты передаются в течение одинакового времени (исключение из этого правила является бит половинного Stop, когда используется 1,5 Stop бита). Эта продолжительность называется битовым периодом или битовым временем; один бит времени равняется шестнадцати тактам бод.

Типовой сигнал данных представляет собой просто напряжение, уровень которого изменяется между состояниями логического нуля и единицы. Приемник может правильно преобразовать эти логические состояния в цифровые данные, только если он знает, когда сделать выборку сигнала.

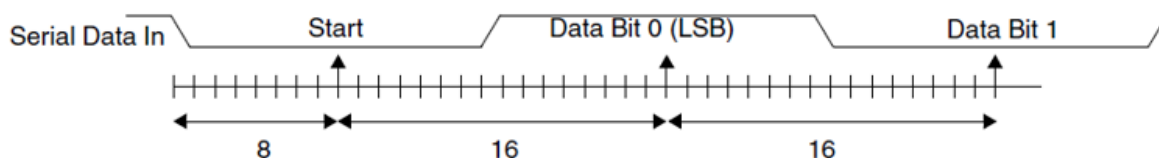
Передатчик формирует битовый поток на основе своего тактового сигнала, а целью приемника является использование своего тактового сигнала для выборки входящих данных в середине каждого битового периода.

Последовательность приемника начинается с заднего фронта стартового бита. Внутренний тактовый генератор приемника полностью независим от внутреннего тактового генератора передатчика – другими словами, эта граница спада сигнала может соответствовать любой точке в тактовом сигнале приемника.

Допустим, что один битовый период соответствует 16 циклам тактового сигнала приемника. В этом случае синхронизация и выборка могут быть выполнены следующим образом:

1. Процесс приема запускается по границе спада стартового бита.
2. Приемник ждет в течение 8 циклов тактового сигнала, чтобы установить момент выборки, который находится близко к середине периода бита.
3. Приемник ждет в течение 16 циклов тактового сигнала, которые приводят его к середине периода первого бита данных.
4. Первый бит данных оцифровывается и сохраняется в регистре приемника, а затем модуль снова ожидает 16 циклов тактового сигнала перед выборкой второго бита данных.
5. Этот процесс повторяется до тех пор, пока все биты данных не будут выбраны и сохранены, а затем нарастающий фронт стопового бита возвращает интерфейс UART в режим ожидания.

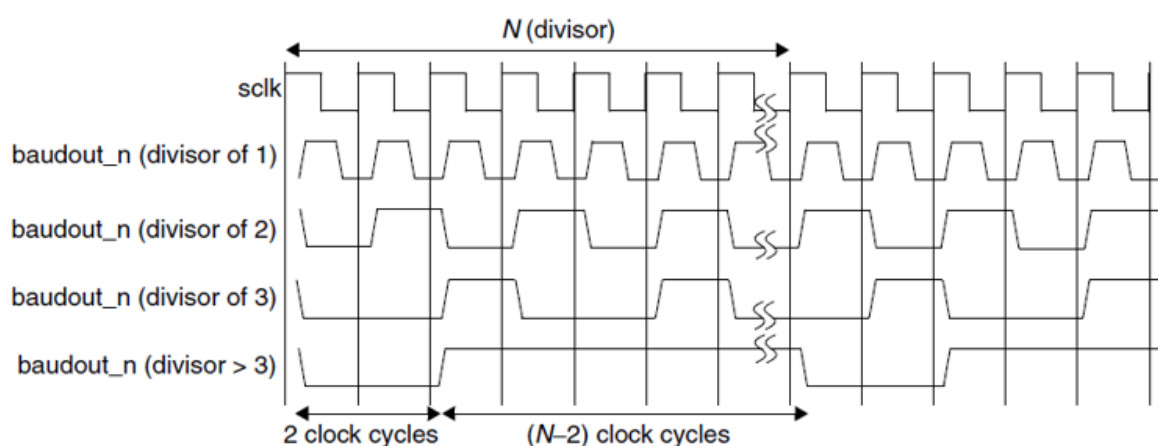
На Рисунок 11.2 показаны точки выборки первых двух битов в последовательном символе.



**Рисунок 11.2. Точки выборки данных**

В рамках стандарта, дополнительный выходной сигнал тактовой частоты в бодах (baudout\_n) предоставляет информацию о синхронизации принимающим устройствам, которым это требуется. Скорость передачи данных UART управляется sclк или рслк в реализации с одним тактом, а также DLH и DLL.

На Рисунок 11.3 показана временная диаграмма выхода baudout для различных значений делителя.



**Рисунок 11.3. Временная диаграмма опорного тактового сигнала в бодах**

### 11.3.2 Последовательный протокол (RS485)

Стандарт RS485 поддерживает последовательную связь по конфигурации с витой парой, такой как RS232. Разница между стандартами RS232 и RS485 заключается в использовании симметричной линии для передачи. Это использование также известно как дифференциальный формат, который отправляет один и тот же сигнал по двум отдельным линиям с фазовой задержкой, а затем сравнивает сигналы в конце, вычитает любой шум и складывает их, чтобы восстановить уровень сигнала. Этот процесс позволяет стандарту RS485 быть жизнеспособным на больших расстояниях.

UART поддерживает последовательный протокол RS485, который обеспечивает передачу последовательных данных с использованием интерфейса RS485. Сигналы включения драйвера (DE) и включения приемника (RE) генерируются для включения поддержки интерфейса RS485. Сигналы DE и RE генерируются аппаратно, и время

подтверждения/отключения для этих сигналов программируется. Активный уровень этих сигналов настраивается.

Конфигурация UART для интерфейса RS485 выполняет следующие действия:

1. Бит 0 регистра TCR включает или отключает режим RS485.
2. Бит 1 и бит 2 TCR используются для выбора полярности сигналов RE и DE.
3. Биты TCR [4:3] выбирают тип передачи в режиме RS485.
4. Регистры DE\_EN и RE\_EN используются для программного управления сигналами DE и RE.
5. Регистр DET используется для программирования времени подтверждения и отмены сигнала DE.
6. Регистр TAT используется для программирования времени выполнения с DE на RE и RE на DE.

#### 11.3.2.1 Время подтверждения и отмены DE

Время подтверждения и отмены сигнала DE контролируется через регистр DET:

- Время утверждения DE (DET[7:0]): время утверждения – это время между активацией сигнала DE и началом Start битом.
- Время отмены подтверждения DE (DET[15:8]): Время отмены подтверждения – это время между концом последнего Stop бита в переданном символе и отключением сигнала DE.

Аппаратное обеспечение гарантирует соблюдение этих значений для подтверждения DE и отмены подтверждения DE до/после активации передачи данных.

На Рисунок 11.4  $t_1$  – представляет время подтверждения DE, а  $t_2$  — время отмены подтверждения DE. DE отменяется только после того, как все символы данных переданы.

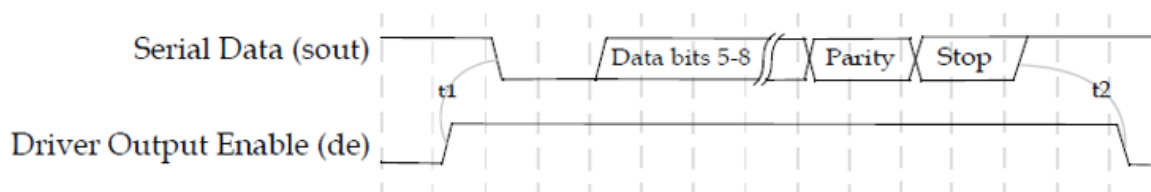


Рисунок 11.4. DE утверждение и отказ от утверждения



### 11.3.2.2 Режимы RS485

UART состоит из следующих режимов RS485, основанных на XFER\_MODE в регистре TCR:

- Полнодуплексный режим — XFER\_MODE установлен в 0.
- Программно-управляемый полудуплексный режим — XFER\_MODE установлен в 1.
- Полудуплексный режим с аппаратным управлением — XFER\_MODE установлен в 2.

#### 11.3.2.2.1 Полнодуплексный режим

Полнодуплексный режим поддерживает одновременную передачу и прием.

Сигнал DE:

- Становится активным, если выполняются оба эти условия:
  - Если для DE\_EN[0] установлено значение 1.
  - Если THR не пуст в режиме без FIFO или FIFO передатчика не пуст в режиме FIFO.
- Становится неактивным, если выполняются оба эти условия:
  - Когда текущая последовательная передача завершена.
  - Либо DE\_EN[0] установлен в 0, либо передатчик FIFO пуст в режиме FIFO или THR пуст в режиме без FIFO.

Сигнал RE:

- Становится активным, если RE\_EN[0] установлен в 1.
- Становится неактивным, если RE\_EN[0] установлен в 0.

Пользователь может выбирать, когда передавать, а когда принимать. И RE и DE могут быть одновременно подтверждены или отменены в любое время. В этом режиме UART не устанавливает время переключения между передачей и приемом или приемом и передачей. Этот режим может использоваться в полнодуплексном режиме работы, где для передачи и приема используется отдельная дифференциальная пара проводов.

#### 11.3.2.2.2 Программно-управляемый полудуплексный режим

Программно-управляемый полудуплексный режим поддерживает одновременную передачу или прием, но не обе одновременно. Переключение между передачей на прием

или приемом на передачу осуществляется путем программирования регистров DE\_EN и RE\_EN.

Сигнал DE:

- Становится активным, если выполняются следующие условия:
  - DE\_EN[0] установлен в 1.
  - THR не пуст в режиме без FIFO или FIFO передатчика не пуст в режиме FIFO.
  - Если какая-либо передача приема продолжается, то сигнал ожидает, пока приемник не завершит работу и истечения счетчика времени выполнения («RE to DE»).
- Становится неактивным, если выполняются следующие условия:
  - Текущая последовательная передача завершена.
  - DE\_EN[0] установлен в 0.
  - Либо FIFO передатчика пуст в режиме FIFO, либо THR пуст в режиме без FIFO.

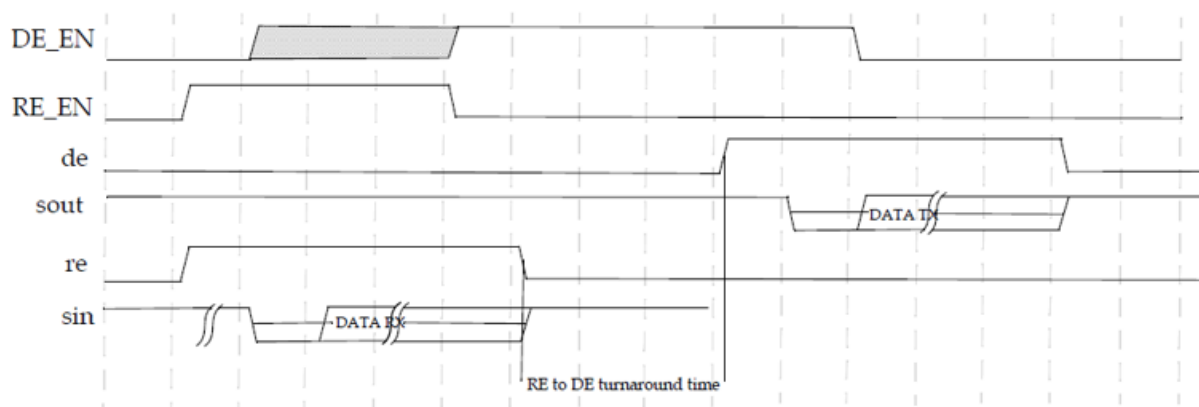
Сигнал RE:

- Становится активным, если выполняются следующие условия:
  - RE\_EN[0] установлен в 1.
  - Если какая-либо передача приема продолжается, то сигнал ожидает окончания передачи и истечения счетчика времени выполнения («DE to RE»).
- Становится неактивным, если выполняются следующие условия:
  - Текущая последовательная передача приема завершена.
  - RE\_EN[0] установлен в 0.

Пользователь должен включить либо DE, либо RE, но не оба одновременно. Поскольку сигналы «RE» и «DE» являются взаимоисключающими, пользователь должен убедиться, что они оба не запрограммированы на то, чтобы быть активными в любой момент времени.

#### **11.3.2.2.2.1 Время выполнения от RE до DE**

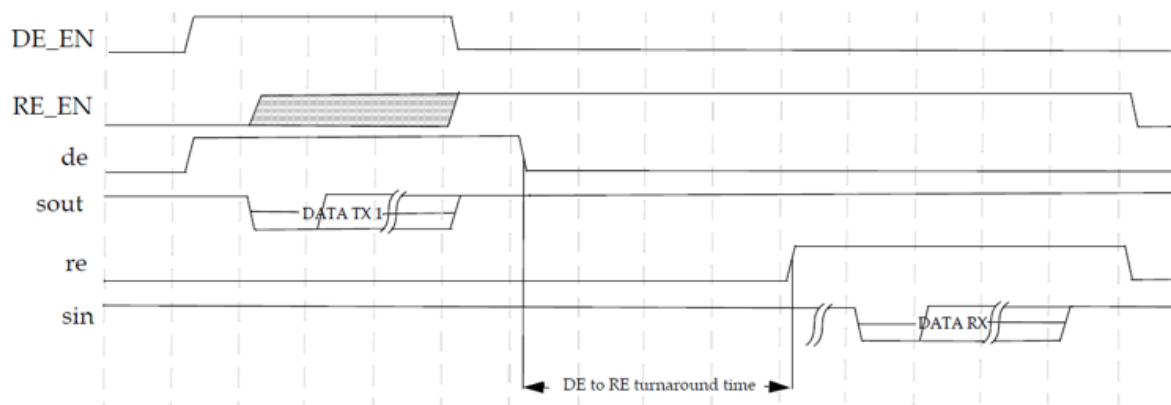
UART дополняет состоянием ожидания перед переключением из режима приема в режим передачи, как показано на Рисунок 11.5.



**Рисунок 11.5. Время выполнения от RE до DE**

#### 11.3.2.2.2 Время выполнения от DE до RE.

UART дополняет состоянием ожидания перед переключением из режима передачи в режим приема, как показано на Рисунок 11.6.



**Рисунок 11.6. Время выполнения от DE до RE**

#### 11.3.2.2.3 Полудуплексный режим с аппаратным управлением

Полудуплексный режим с аппаратным управлением поддерживает передачу или прием одновременно, но не оба одновременно. Если оба бита «DE Enable» и «RE Enable» регистров DE\_EN и RE\_EN включены, переключение между передачей на прием или приемом на передачу происходит автоматически, исходя из состояния Tx-FIFO.

Сигнал DE:

- Становится активным, если выполняются следующие условия:
  - DE\_EN[0] установлен в 1.

- THR не пуст в режиме без FIFO или FIFO передатчика не пуст в режиме FIFO.
- Если передача данных на прием продолжается, то сигнал ждет, пока прием не завершится и истечения счетчика времени выполнения («RE to DE»).
- Становится неактивным, если выполняются следующие условия:
  - Текущая последовательная передача завершена.
  - Либо FIFO передатчика пуст в режиме FIFO, либо THR пуст в режиме без FIFO, либо DE\_EN[0] установлен в 0.

RE сигнал:

- Становится активным, если выполняются следующие условия:
  - RE\_EN[0] установлен в 1.
  - Либо FIFO передатчика пуст в режиме FIFO, либо THR пуст в режиме без FIFO.
  - Если идет передача данных, то сигнал ожидает окончания передачи данных и истечения счетчика времени выполнения («DE to RE»).
- Становится неактивным, если выполняются следующие условия:
  - Текущая последовательная передача приема завершена.
  - Либо FIFO передатчика не пуст в режиме FIFO, либо THR не пуст в режиме без FIFO, либо RE\_EN[0] установлен в 0.

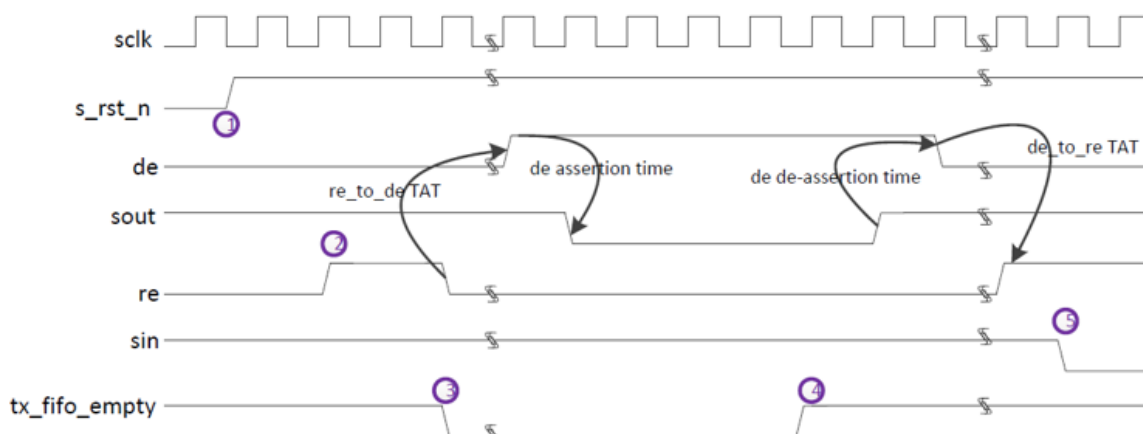
### 11.3.2.3 Примеры сценариев

Рассмотрим сценарий, в котором контролер UART принимает 3 символа и другое устройство UART отправляет эти символы. В то время как 1-й символ принимается контроллером UART, если программное обеспечение выполняет запись в TX FIFO контроллера UART, то в конце первого символа контролер UART переключит режим с приема на передачу. UART отменит утверждение RE и подтвердит сигнал DE. Это приведет к тому, что контроллер UART не будет получать последующие символы.

Следовательно, в полудуплексном режиме с аппаратным переключением пользователь должен убедиться, что данные приема были получены перед записью в Tx-FIFO, чтобы избежать пропуска принимаемых символов.

#### 11.3.2.3.1 Обычный сценарий передачи

Рисунок 11.7 представляет пример сценария для нормальной передачи.



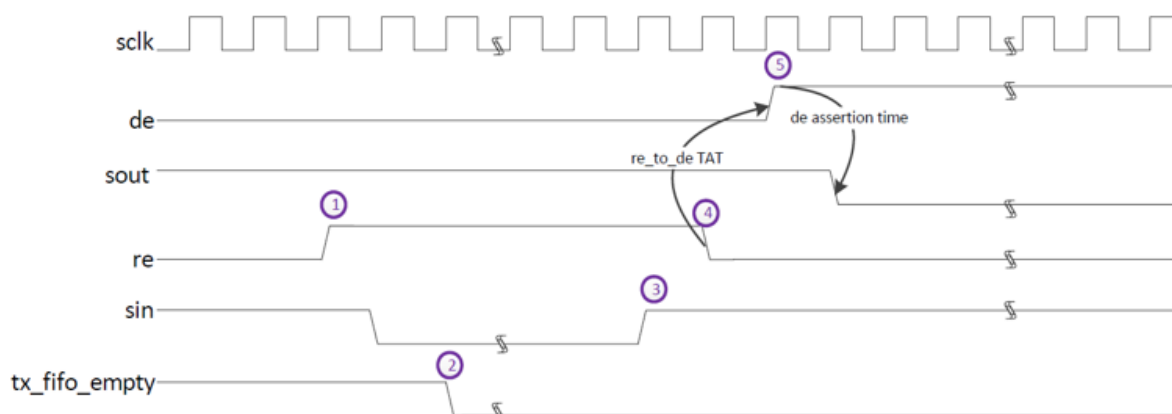
**Рисунок 11.7. Сценарий, где XFER\_MODE = 2**

На Рисунок 11.7 показаны следующие действия в различных точках этого сценария:

1. В этот момент сброс снимается, а сигналы DE и RE приводятся к своим сконфигурированным значениям сброса.
2. В этот момент регистры DE\_EN и RE\_EN устанавливаются 1, и tx\_fifo\_empty (внутренней сигнал UART) равен 1, что указывает на отсутствие данных в TX FIFO.
3. В этот момент программное обеспечение заполняет TX FIFO, а передача приема не происходит. Поэтому сигнал RE становится низким. Тем не менее, контроллер UART ожидает значения TAT перед подачей сигнала DE. После подачи сигнала DE передача символа начинается с учетом «времени отмены сигнала».
4. В этот момент TX FIFO становится пустым. После передачи текущего символа, UART снимает сигнал DE. Контроллер UART ждет, пока появится значение TAT, прежде чем снова подать сигнал RE.
5. В этот момент контроллер UART начинает получать символ.

#### 11.3.2.3.2 Сценарий приема, когда TX FIFO заполняется

В этом сценарии буфер TX FIFO заполняется при получении символа. В этом случае ожидается, что UART дожидется завершения текущего символа, прежде чем сменить роль и начать передачу.



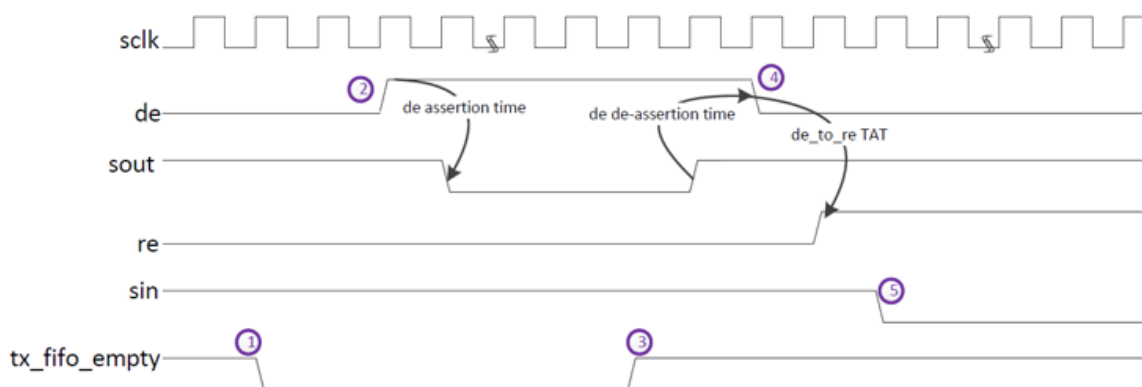
**Рисунок 11.8. Прием в процессе, когда TX FIFO заполнен**

На Рисунок 11.8 показаны следующие действия в различных моментах этого сценария:

1. Программное обеспечение программирует DE\_EN и RE\_EN в 1, тем самым подтверждает сигнал RE. После этого контроллер UART начинает получать символ.
2. Tx\_fifo\_empty понижается. Однако контроллер UART ожидает, пока не будет получен текущий символ, прежде чем утверждать сигнал DE.
3. Входящий символ полностью принят.
4. Сигнал RE становится неактивным после того, как Stop бит полностью принят.
5. После TAT, сигнал DE становится активным, и контроллер UART начинает передачу по истечению времени DET.

### 11.3.2.3.3 TX FIFO заполнен перед включением регистров DE\_EN и RE\_EN

В этом случае TX FIFO заполняется до включения DE\_EN или RE\_EN. Контроллер UART включает DE вместо RE в этом случае, потому что TX FIFO уже содержит данные для передачи.



### Рисунок 11.9. TX FIFO заполняется перед включением DE/RE

На Рисунок 11.9 показаны следующие действия в различных моментах этого сценария:

1. Tx\_fifo\_empty устанавливается в 0.
2. DE\_EN и RE\_EN устанавливается в 1. Поскольку данные уже присутствуют в TX FIFO, контроллер UART подает сигнал DE. UART запускает отправку по истечению времени DET.
3. TX FIFO становится пустым.
4. Сигнал DE отменяется по истечении времени DET. После TAT утверждается сигнал RE.
5. Контроллер UART начинает принимать входящий символ.

### 11.3.3 Протокол IrDA 1.0 SIR

Режим Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) поддерживает двунаправленную передачу данных с удаленными устройствами с использованием инфракрасного излучения в качестве среды передачи. В режиме IrDA 1.0 SIR максимальная скорость передачи данных составляет 115,2 Кбод.

Формат данных аналогичен стандартному формату последовательных данных `sout` и `sin`. Каждый символ данных отправляется последовательно в следующем порядке:

1. Начинается с начального бита.
2. За ним следует 8 бит данных.
3. Заканчивается по крайней мере одним Stop битом.

Таким образом, количество передаваемых битов данных фиксировано. Информация о четности не может быть предоставлена, и в этом режиме используется только один Stop бит. Попытка отрегулировать количество отправляемых битов данных или включить четность с помощью регистра LCR не имеет никакого эффекта.

Конфигурация UART для IrDA 1.0 SIR выполняет следующие действия:

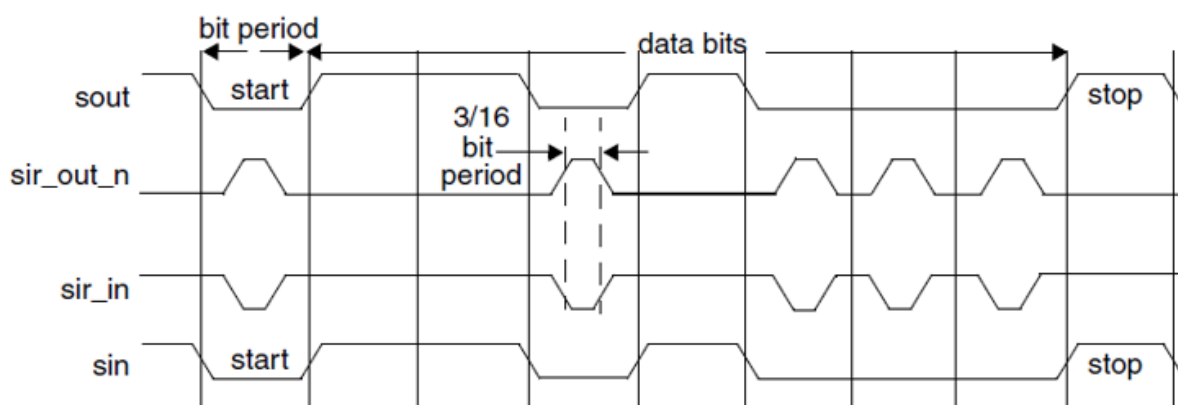
- Бит 6 регистра MCR включает или отключает режим IrDA 1.0 SIR.
- Отключение режима IrDA SIR приводит к тому, что логика не реализуется; режим не может быть активирован.
- Когда режим IrDA SIR включен и активен, последовательные данные передаются и принимаются через порты `sir_out_n` и `sir_in`.

Передача или отсутствие передачи одного инфракрасного импульса указывает на следующее:

- Передача одного инфракрасного импульса указывает на логический 0.
- Отсутствие передачи импульса указывает на логическую 1.

Длительность каждого импульса составляет  $3/16$  обычного битового времени. Таким образом, каждый новый символ начинается с инфракрасного импульса в качестве Start бита. Однако полученные данные инвертируются из передаваемых данных из-за инфракрасных импульсов, возбуждающих питание базы фототранзистора приемника IrDA, что снижает его выход. Этот инвертированный транзисторный выход подается на порт UART `sir_in`, который обеспечивает правильную полярность UART.

На Рисунок 11.10 показана временная диаграмма для формата данных IrDA SIR по сравнению со стандартным последовательным форматом.



**Рисунок 11.10. Формат данных IrDA SIR**

Так как UART может быть сконфигурирован для поддержки режима приема с низким энергопотреблением. Когда UART настроен в этом режиме, можно принимать импульсы SIR длительностью 1,41 микросекунды (минимальная длительность импульса), а также номинальные  $3/16$  нормального времени последовательного бита. Чтобы использовать этот режим приема с низким энергопотреблением, необходимо запрограммировать регистры LPDLL / LPDLH.

Когда включен режим IrDA SIR, UART работает аналогично отключенному режиму, за одним исключением: передача данных может осуществляться только в полудуплексном режиме, когда включен режим IrDA SIR. Это связано с тем, что физический уровень IrDA SIR определяет минимальную задержку в 10 мс между передачей и приемом; эта задержка в 10 мс должна быть сгенерирована программным обеспечением.



### 11.3.4 Поддержка FIFO

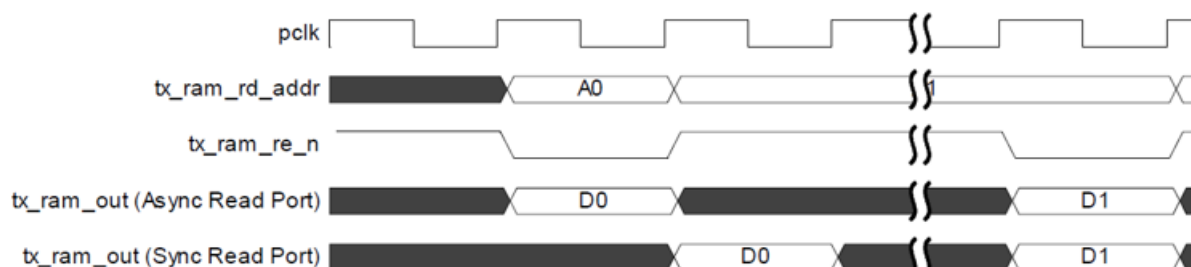
Можно настроить UART для реализации FIFO, которые буферизируют передачу и прием данных. Если поддержка FIFO не выбрана, то FIFO не реализованы, и только один байт данных приема и байт данных передачи может храниться одновременно в регистрах RBR и THR; это подразумевает 16450-совместимый режим работы. Однако в этом режиме работы большинство расширенных функций недоступны.

В режиме FIFO можно выбрать один из следующих вариантов FIFO:

- FIFO RAM, поставляемые заказчиком.
- Внутренняя RAM на основе D-триггеров.

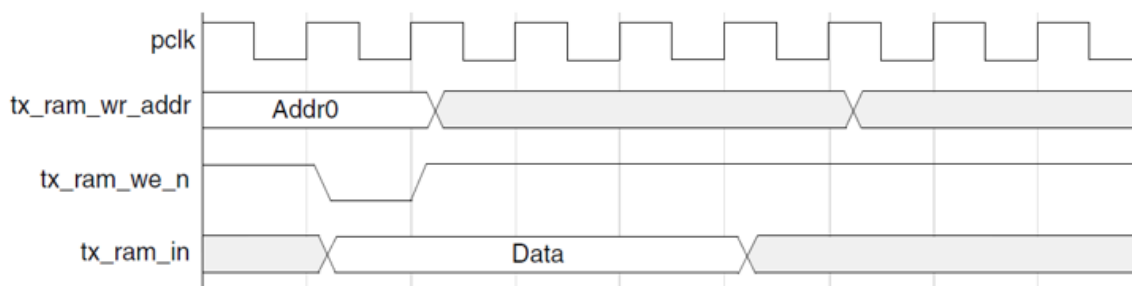
Если сконфигурированная глубина FIFO превышает 256, выбор памяти FIFO ограничен. Кроме того, выбор внутренней памяти ограничивает тип порта чтения памяти для RAM-портов синхронного чтения на основе D-триггера.

Если выбрана поддержка внешней оперативной памяти, можно использовать синхронную или асинхронную оперативную память. Асинхронная оперативная память обеспечивает считывание данных в течение тактового цикла. Синхронная одноступенчатая оперативная память регистрирует данные по текущему адресу и недоступна до следующего тактового цикла. На Рисунок 11.11 показана временная диаграмма, как для асинхронной, так и для синхронной RAM.



**Рисунок 11.11. Время чтения RAM**

Точно также можно использовать синхронную RAM для записи, которая регистрирует данные по текущему адресу. На Рисунок 11.12 показана временная диаграмма для записи в RAM.



**Рисунок 11.12. Время записи RAM**

Когда выбрана поддержка FIFO, для целей тестирования доступен дополнительный программируемый режим доступа FIFO, который позволяет:

- Получить FIFO для записи мастером.
- Передать FIFO для чтения мастером.

Когда режим доступа FIFO не выбран, ни одна из соответствующих логик не реализована, и режим не может быть включен.

Когда выбран режим доступа FIFO, его можно включить с помощью регистра доступа FIFO (FAR [0]). После включения, управляющие части FIFO передачи и приема сбрасываются, и FIFO обрабатываются как пустые.

Данные могут быть записаны в передающие FIFO; однако в этом режиме не происходит никакой последовательной передачи и, таким образом, никакие данные не покидают FIFO. Данные, которые были записаны в FIFO передачи, могут быть считаны обратно с помощью регистра TFR.

Точно так же данные могут быть прочитаны из FIFO приемника. Поскольку в этом режиме работа UART останавливается, данные должны быть записаны в принимающие FIFO, чтобы их можно было прочитать обратно.

Данные записываются в FIFO приемника с использованием регистра RFW. Два старших бита 10-битного регистра используются для записи информации об ошибке кадра и обнаружения ошибок четности в FIFO приемника следующим образом:

- RFW[9] указывает на ошибку кадра.
- RFW[8] указывает на ошибку четности.

Хотя эти биты не могут быть считаны обратно через регистр буфера приема, их можно проверить, прочитав регистр состояния строки и проверив соответствующие биты, когда данные, о которых идет речь, находятся в верхней части FIFO.

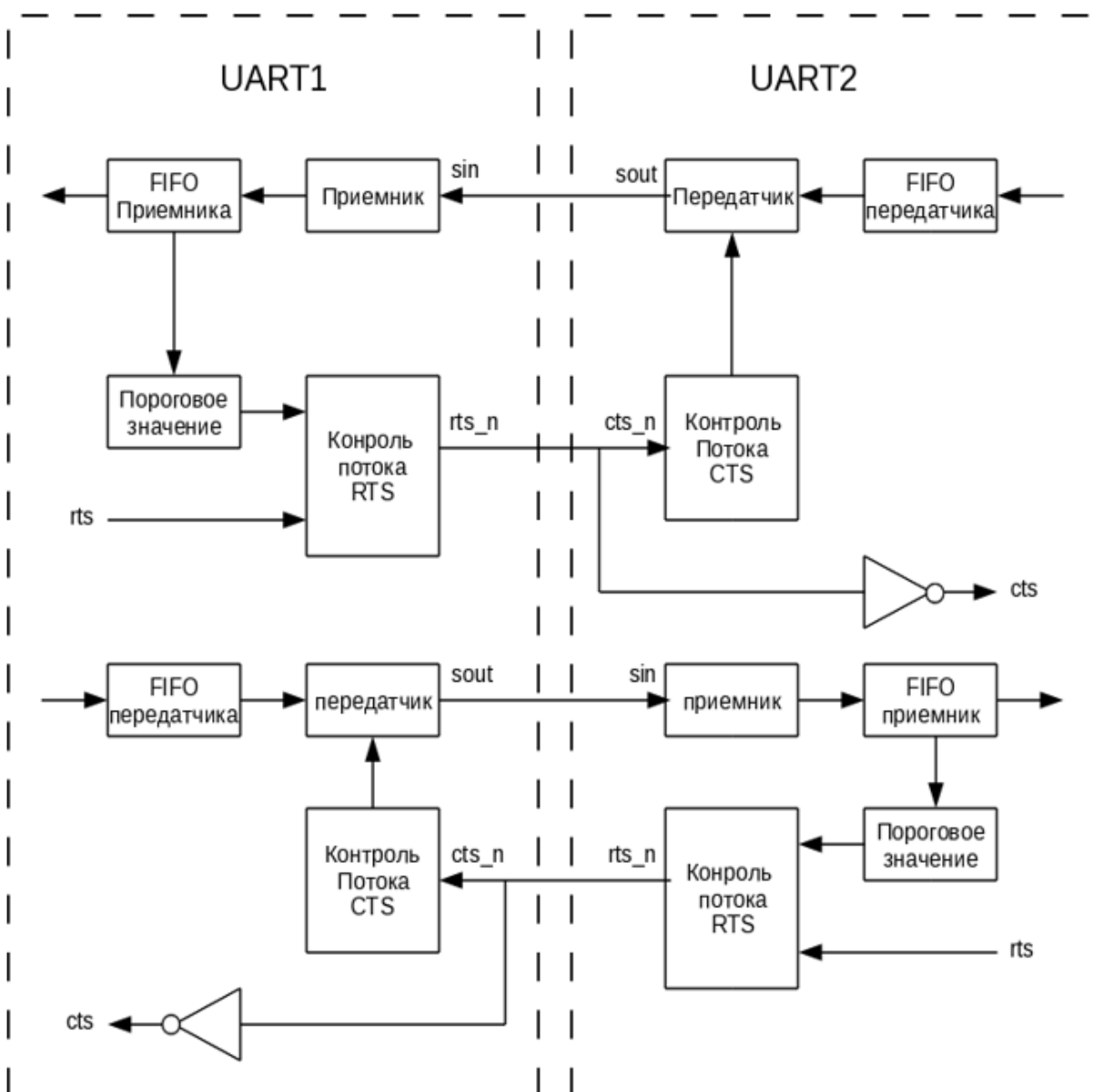
### **11.3.5 Последовательная передача потока данных**

Когда FIFO передачи содержит несколько записей данных, UART передает символы из FIFO последовательно. Однако, если параметр конфигурации CLOCK\_MODE равен 2, задержки синхронизации UART могут вызвать период бездействия между концом текущего бита Stop и началом следующего бита Start; это проявляется, как увеличенная длительность бита Stop.

### **11.3.6 Автоматическое управление потоком**

UART можно настроить так, чтобы был доступен режим управления последовательным потоком данных. Если FIFO не реализованы, этот режим выбрать невозможно. Когда автоматическое управление потоком не выбрано, никакая соответствующая логика не реализуется, и режим не может быть включен. Когда выбран режим автоматического управления потоком, его можно включить с помощью регистра MCR [5].

На Рисунок 11.13 показана блок-схема функции автоматического управления потоком.



**Рисунок 11.13. Блок-схема автоматического управления потоком**

Auto RTS – активируется, когда происходит следующее:

- Автоматическое управление потоком выбирается во время настройки.
- Реализованы FIFO.
- RTS (установлены бит MCR[1] и бит MCR[5]).
- FIFO включены (установлен бит FCR[0]).
- Режим SIR отключен (бит MCR[6] не установлен).

Когда функция Auto RTS включена, выход rts\_n принудительно отключается (высокий уровень), когда уровень FIFO приемника достигает порога, установленного FCR[7:6], но

только если триггер управления потоком RTC отключен. В противном случае, выход `rts_n` принудительно неактивен (высокий уровень), когда FIFO почти заполнен, где "почти заполнен" означает два свободных слота в FIFO. Когда `rts_n` подключен к входу `cts_n` другого устройства UART, другой UART прекращает отправку последовательных данных, пока в приемнике FIFO не освободится место; то есть, пока он не станет полностью пустым.

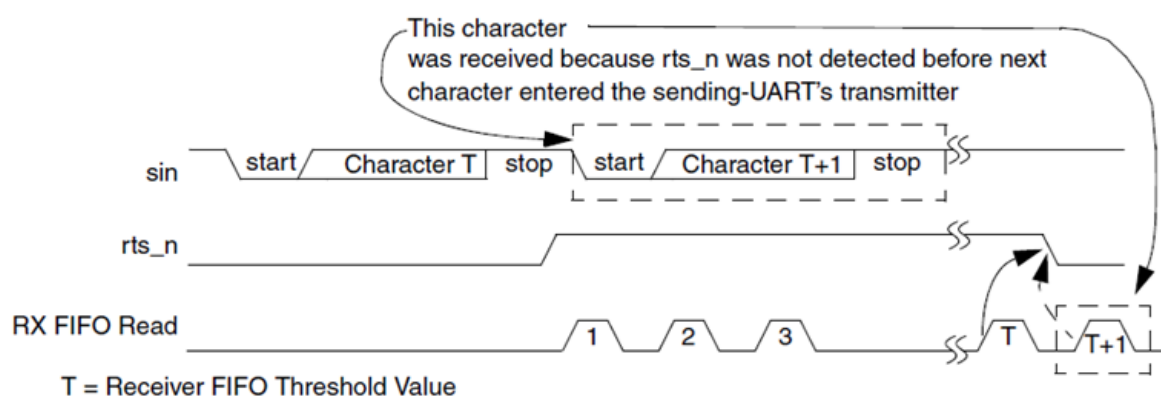
Пороговые значения FIFO приемника:

- 1.
- $1/4$ .
- $1/2$ .
- 2 меньше, чем полный.

Поскольку один дополнительный символ может быть передан в UART после того, как `rts_n` стал неактивным - из-за того, что данные уже поступили в блок передатчика в другом UART - установка порога на «2 меньше, чем полный» позволяет максимально использовать FIFO с зоной безопасности в один символ.

Как только FIFO приемника становится полностью пустым в результате чтения RBR, `rts_n` снова становится активным (низким), сигнализируя другому UART о продолжении отправки данных.

На Рисунок 11.14 показана временная диаграмма работы Auto RTS.



**Рисунок 11.14. Auto RTS**

Auto CTS – активируется, когда происходит следующее:

- Автоматическое управление потоком выбирается во время настройки.
- Реализованы FIFO.
- AFCE (`MCR[5] = 1`).

- FIFO активируются через бит FCR[0] регистра управления FIFO.
- Режим SIR отключен ( $MCR[6] = 0$ ).

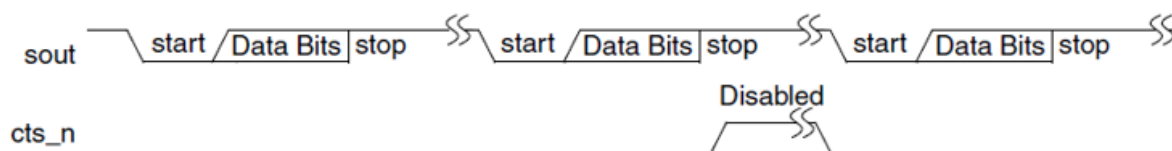
Когда функция Auto CTS включена (активна), передатчик UART отключается всякий раз, когда вход cts\_n становится неактивным (высоким); это предотвращает переполнение FIFO принимающего UART.

Если вход cts\_n не отключен до середины последнего Stop бита, перед отключением передатчика передается другой символ. В то время как передатчик отключен, в FIFO передатчика все еще можно записать.

Поэтому при использовании этого режима происходит следующее:

- Регистр состояния UART может быть прочитан, чтобы проверить, заполнен ли передающий FIFO (USR[1] установлен в 0).
- Текущий уровень FIFO можно прочитать с помощью регистра TFL.
- Программируемый режим прерывания THRE должен быть включен для доступа к статусу «FIFO full» с помощью LSR.

На Рисунок 11.15 показана временная диаграмма работы Auto CTS.



**Рисунок 11.15. Auto CTS**

### 11.3.7 Аппаратное управление потоком данных

Порт UART имеет 2 дополнительных вывода RTS\_N и CTS\_N, через которые осуществляется автоматическое управление потоком между двумя портами UART. Чтобы включить режим автоматического управления потоком необходимо установить в 1 бит MCR[5].

Выходной сигнал RTS\_N переходит в 1 в случае, когда число записей в приемнике FIFO превышает пороговое значение, записанное в битах FCR[6:7]. Когда вывод RTS\_N соединен с выводом CTS\_N другого порта UART, то другой UART будет вынужден приостановить передачу данных, до тех пор, пока FIFO приемника первого UART не станет пустым. После этого, сигнал RTS\_N перейдет в 0 и передача будет продолжена.

### 11.3.8 Прерывания

UART формирует следующие прерывания:

1. Ошибка приемника.
2. Приемник имеет данные для чтения.
3. Таймаут символа (в режиме FIFO).
4. Регистр передатчика пуст / Буфер передатчика ниже порогового значения (в режиме работы по прерыванию THRE).
5. Контроллер занят.

### 11.3.9 Программируемое прерывание THRE

UART можно настроить для режима программируемого прерывания THRE, чтобы повысить производительность системы; если FIFO не реализованы, то этот режим не может быть выбран:

- Когда режим программируемого прерывания THRE не выбран, никакая логика не реализуется, и режим не может быть включен.
- Когда выбран режим программируемого прерывания THRE, его можно включить с помощью регистра IER [7].

### 11.3.10 Включение Clock Gate

UART может быть сконфигурирован так, чтобы иметь выход разрешения синхронизации.

- Когда опция включения clock gate не выбрана, никакая логика не реализуется.
- Когда выбрана опция включения clock gate, сигналы включения - `uart_lp_req_pclk` для реализаций с одним тактовым сигналом или `uart_lp_req_pclk` и `uart_lp_req_sclk` для реализации с двумя тактовыми сигналами - используется для обозначения следующего:
  - Конвейер передачи и приема пуст (нет данных).
  - Никаких действий не произошло.
  - Входные сигналы управления не изменились более чем за один символ – время, необходимое для передачи/приема символа.

Утверждение разрешающих сигналов clock gate указывает на то, что UART неактивен, поэтому тактовые импульсы могут быть гейтированы, чтобы перевести устройство в режим

пониженного энергопотребления (LP). Следовательно, следующее должно быть истинным хотя бы для одного символа времени для утверждения сигнала включения clock gate:

- Нет данных в RBR (в режиме без FIFO) или RX FIFO пуст (в режиме FIFO).
- Нет данных в THR (в режиме без FIFO) или TX FIFO пуст (в режиме FIFO).
- sin/sir\_in и sout/sir\_out\_n неактивен (sin/sir\_in держится на высоком уровне, sout на высоком уровне или sir\_out\_n на низком уровне), что указывает на отсутствие активности.
- Нет изменений во входных сигналах управления.

Подтверждение включения clock gate не происходит в следующих режимах работы:

- Режим обратной связи.
- Режим доступа FIFO.
- При передаче паузы.

### 11.3.11 Поддержка DMA

UART поддерживает DMA с использованием выходных сигналов dma\_tx\_req\_n и dma\_rx\_req\_n, чтобы узнать:

- Когда данные можно читать.
- Когда передающий FIFO пуст.

#### 11.3.11.1 Режимы DMA

UART использует два канала DMA — один для передачи данных и один для приема данных. Существует два режима DMA:

- Режим 0 — 3 бит регистра управления FIFO установлен в 0.
- Режим 1 — 3 бит регистра управления FIFO установлен в 1.

##### 11.3.11.1.1 DMA режим 0

Режим DMA 0 поддерживает единовременную передачу данных DMA.

Сигнал dma\_tx\_req\_n:

- Становится активным-низким при следующих условиях:
  - THR пуст в режиме без FIFO.



- FIFO передатчика пуст в режиме FIFO с отключенным режимом программируемого прерывания THRE.
- FIFO передатчика находится на уровне или ниже запрограммированного порога с включенным режимом программируемого прерывания THRE.
- Становится неактивным:
  - Одиночный символ был записан в THR или FIFO передатчика с отключенным режимом программируемого прерывания THRE.
  - FIFO передатчика превышает пороговое значение с включенным режимом программируемого прерывания THRE.

Сигнал dma\_rx\_req\_n:

- Становится активным на низком уровне, когда один символ доступен в FIFO приемника или регистре буфера приема.
- Становится неактивным, когда регистр буфера приема или FIFO приемника пусты, в зависимости от режима FIFO.

#### 11.3.11.1.2 DMA режим 1

Режим DMA 1 поддерживает мульти-DMA передачу данных, при которой несколько передач осуществляются непрерывно до тех пор, пока не опустеет FIFO приемника или не заполнится FIFO передатчика.

Сигнал dma\_tx\_req\_n:

- Когда FIFO передатчика пуст при отключенном режиме программируемого прерывания THRE.
- Когда FIFO передатчика находится на уровне или ниже запрограммированного порога при включенном режиме программируемого прерывания THRE.

Сигнал dma\_tx\_req\_n подается, когда FIFO передатчика полностью заполнен.

Сигнал dma\_rx\_req\_n:

- Когда FIFO приемника находится на уровне или выше запрограммированного уровня срабатывания.
- Когда истекло время ожидания символа; ERBFI не нужно устанавливать.

Сигнал dma\_rx\_req\_n отменяется, когда FIFO приемника становится пустым.

### 11.3.12 Аппаратный интерфейс запросов PDMA

Контроллеры UART0, UART1, UART2 имеют аппаратный интерфейс запросов PDMA. Запросы формируются следующим образом.

Запрос к PDMA на запись в UART формируется:

1. При выключенном FIFO, когда регистр THR пуст.
2. При включенном FIFO но выключенном прерывании THRE, когда FIFO передатчика пуст.
3. При включенном FIFO и включенном прерывании THRE, когда кол-во слов данных в FIFO передатчика ниже установленного порогового значения, установленного в поле TET регистра FCR.

Запрос к PDMA на чтение из UART формируется:

1. При выключенном FIFO, когда в регистре RBR есть данные.
2. При включенном FIFO, когда кол-во слов данных в FIFO приемника равно или превышает пороговое значение, установленное в поле RFTL регистра FCR.

Размер всех пересылок UART-PDMA контролирует только PDMA.

### 11.3.13 Сигнал сброса

При настройке для асинхронной последовательной синхронизации UART включает два отдельных сигнала сброса, каждый из которых предназначен для своего собственного домена синхронизации:

- Present сбрасывает логику в pclk.
- s\_rst\_n сбрасывает логику с sclk.

Во избежание серьезных сбоев в работе оба тактовых домена UART должны быть сброшены до того, как будет предпринята любая попытка отправить или получить данные по последовательной линии; то есть сбросить только один тактовый домен UART без сброса другого тактового домена является недопустимой.

Каждый сигнал сброса должен быть отключен синхронно с соответствующим тактовым сигналом.

При утверждении сигналов сброса сигнал s\_rst\_n должен быть заявлен до или в то же время, что и presetn; это предотвращает любую неожиданную активность на последовательной

линии, которая может возникнуть в результате сброса регистров программирования без сброса последовательной логики.

Аналогично, при отмене подтверждения сигналов сброса, `s_rst_n` должен быть отменен до отмены `presetn`. Самая безопасная процедура сброса UART следующая:

- Утвердить `s_rst_n` и `presetn`; последовательность утверждения этих двух сигналов и их временные отношения с `sclk` и `pclk` не важны.
- Снять сигнал `s_rst_n` синхронно с `sclk`.
- Снять сигнал `present` синхронно с `pclk`.

Оба сигнала сброса должны быть активны как минимум в течение трех циклов соответствующего тактового сигнала.

## 12. SPI: контроллер SPI

В микросхеме реализовано три блока SPI.

### 12.1 Основные особенности

Контроллер SPI имеет следующие характеристики:

- Поддерживаемые протоколы – Motorola SPI, Texas Instruments Synchronous Serial, National Semiconductor Microwire.
- Аппаратно заданные режимы работы master (SPI0, SPI1) и slave (SPI2)
- Динамическое управление скоростью передачи данных в режиме master
- Размер одного слова от 4 до 32 бит (SPI0, SPI1) или от 4 до 16 бит (SPI2)
- FIFO буфер приемника и передатчика 8 слов
- 1 канал slave-select
- Аппаратный интерфейс запросов к DMA

Всего присутствует три контроллера SPI

### 12.2 Регистры

Перечень регистров приведен в Таблица 12.1.

**Таблица 12.1. Перечень программно-доступных регистров**

Смещение	Условное обозначение	Название регистра	Исходное состояние	Тип доступа
0x0	CTRLR0	Управляющий регистр 0	0x01070000	RW
0x4	CTRLR1	Управляющий регистр 1	0x00000000	RW
0x8	SSIENR	Разрешающий регистр	0x00000000	RW
0xc	MWCR	Управление протоколом Microwire	0x00000000	RW
0x10	SER	Управление слэйвом	0x00000000	RW
0x14	BAUDR	Выбор скоростного режима	0x00000000	RW
0x18	TXFTLR	Порог заполнения, FIFO передатчика	0x00000000	RW
0x1c	RXFTLR	Порог заполнения, FIFO приемника	0x00000000	RW
0x20	TXFLR	Уровень заполнения FIFO передатчика	0x00000000	RO
0x24	RXFLR	Уровень заполнения FIFO приемника	0x00000000	RO
0x28	SR	Регистр статуса	0x00000006	RO
0x2c	IMR	Регистр маски прерываний	0x0000003F	RW

0x30	ISR	Регистр статуса прерываний	0x00000000	RO
0x34	RISR	Регистр прерываний	0x00000000	RO
0x38	TXOICR	Регистр снятия прерывания переполнения FIFO передатчика	0x00000000	RO
0x3c	RXOICR	Регистр снятия прерывания переполнения FIFO приемника	0x00000000	RO
0x40	RXUICR	Регистр снятия прерывания опустошения FIFO приемника	0x00000000	RO
0x44	MSTICR	Регистр снятия прерывания Multi-Master	0x00000000	RO
0x48	ICR	Регистр снятия прерываний	0x00000000	RO
0x4c	DMACR	Регистр управление DMA	0x00000000	RW
0x50	DMATDLR	Уровень передачи DMA	0x00000000	RW
0x54	DMARDLR	Уровень приема DMA	0x00000000	RW
0x58	IDR	Регистр идентификации	0xFFFFFFFF	RO
0x5c	SSI_VERSION_I D	Версия компонента	0x3430312A	RO
0x60	DR0	Регистр данных x	0x00000000	RW
0x64	DR1	Регистр данных x	0x00000000	RW
0x68	DR2	Регистр данных x	0x00000000	RW
0x6c	DR3	Регистр данных x	0x00000000	RW
0x70	DR4	Регистр данных x	0x00000000	RW
0x74	DR5	Регистр данных x	0x00000000	RW
0x78	DR6	Регистр данных x	0x00000000	RW
0x7c	DR7	Регистр данных x	0x00000000	RW
0x80	DR8	Регистр данных x	0x00000000	RW
0x84	DR9	Регистр данных x	0x00000000	RW
0x88	DR10	Регистр данных x	0x00000000	RW
0x8c	DR11	Регистр данных x	0x00000000	RW
0x90	DR12	Регистр данных x	0x00000000	RW
0x94	DR13	Регистр данных x	0x00000000	RW
0x98	DR14	Регистр данных x	0x00000000	RW
0x9c	DR15	Регистр данных x	0x00000000	RW
0xa0	DR16	Регистр данных x	0x00000000	RW
0xa4	DR17	Регистр данных x	0x00000000	RW
0xa8	DR18	Регистр данных x	0x00000000	RW
0xac	DR19	Регистр данных x	0x00000000	RW
0xb0	DR20	Регистр данных x	0x00000000	RW
0xb4	DR21	Регистр данных x	0x00000000	RW
0xb8	DR22	Регистр данных x	0x00000000	RW
0xbc	DR23	Регистр данных x	0x00000000	RW
0xc0	DR24	Регистр данных x	0x00000000	RW
0xc4	DR25	Регистр данных x	0x00000000	RW
0xc8	DR26	Регистр данных x	0x00000000	RW
0xcc	DR27	Регистр данных x	0x00000000	RW
0xd0	DR28	Регистр данных x	0x00000000	RW
0xd4	DR29	Регистр данных x	0x00000000	RW
0xd8	DR30	Регистр данных x	0x00000000	RW
0xdc	DR31	Регистр данных x	0x00000000	RW
0xe0	DR32	Регистр данных x	0x00000000	RW
0xe4	DR33	Регистр данных x	0x00000000	RW
0xe8	DR34	Регистр данных x	0x00000000	RW
0xec	DR35	Регистр данных x	0x00000000	RW
0xfc	RSVD	Резерв	0x00000000	RO

### 12.2.1 Регистр CTRLR0

Регистр управления последовательной передачей данных. Запись в регистр невозможна, если регистр SSIENR находится в активном состоянии.

Формат регистра CTRLR0 приведен в следующей таблице:

**Таблица 12.2. Формат регистра CTRLR0**

Разряды	Название	Описание	Сброс
31:25	RSVD_CTRLR0	Резерв – только чтение	0x0
24	SSTE	Slave Select Toggle Enable  Управление переключением slave select линии. Во время функционирования в режиме SPI с фазой синхронизации (SCPH), установленной в 0, этот регистр управляет поведением линии slave select (SPI_SS) между блоками данных. Если это поле установлено в 1, линия SPI_SS будет переключаться между последовательными блоками данных и SPI_SCK будет удерживаться в состоянии по умолчанию. Если поле установлено в 0, SPI_SS будет оставаться в низком положении, а SPI_SCK будет продолжать работу на протяжении всей передачи данных.	0x1
23	RSVD_CTRLR0_23	CTRLR0_23 резерв – только чтение	0x0
22:21	SPI_FRF	SPI Frame Format Возможные значения: ■ 0x0 (STD_SPI_FRF): Standard SPI Frame Format Тип доступа: только чтение	0x0
20:16	DFS_32	Data Frame Size  Размер блока данных в 32-х битном режиме передачи данных.  Поле применимо только для контроллеров spi0 и spi1 RO для контроллера spi2	0x7
15:12	CFS	Control Frame Size.  Выбор длины управляющего слова для протокола Microwire.	0x0
11	SRL	Shift Register Loop.  Используется только в целях тестирования. В активном состоянии соединяет передающий сдвиговый регистр со сдвиговым регистром приемника.	0x0

Разряды	Название	Описание	Сброс
10	SLV_OE	<p>Slave Output Enable для SPI2</p> <p>Резерв – только чтение для SPI0, SPI1</p> <p>Этот бит отключает выдачу данных на выходе MOSI slave-устройства.</p> <p>Данный функционал может использоваться, когда master передает в широкополосном режиме (на все slave-устройства) и только одно slave-устройство может отвечать данными по линии MISO master-устройства.</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>■ 0x1 (DISABLED): Выход slave-устройства отключен</li> <li>■ 0x0 (ENABLED): Выход slave-устройства включен</li> </ul> <p>Значение после сброса: 0x0</p>	0x0
9:8	TMOD	<p>Transfer Mode.</p> <p>Выбор режима последовательной передачи данных. Это поле не влияет на удвоение передачи, а только определяет валидность принимаемых или передаваемых данных.</p> <p>00 – Передача и прием</p> <p>01 – Только передача</p> <p>10 – Только прием</p> <p>11 – EEPROM чтение</p> <p>В режиме “Только передача”, принимаемые данные не валидны и не поступают в FIFO приемника перезаписываются после следующей передачи данных.</p> <p>В режиме “Только прием” передаваемые данные не валидны. После первой передачи, это же слово будет отправлено снова за стандартное время транзакции.</p> <p>В режиме “Прием и передача”, передаваемые и принимаемые данные валидны. Передача продолжится до последнего слова в FIFO передатчика. Принимаемые данные остаются в FIFO приемника до тех пор, пока не будут прочитаны.</p> <p>В режиме “EEPROM чтение”, принимаемые данные не будут валидны до окончания передачи управляющих инструкций. Когда все инструкции были отправлены, принимаемые данные становятся валидными, передаваемые данные становятся не валидными. Все данные передатчика предполагаются управляющими инструкциями.</p>	0x0

Разряды	Название	Описание	Сброс
7	SCPOL	Serial Clock Polarity.  Выбор полярности неактивного тактового сигнала, который становится неактивным, в случае отсутствия передаваемых данных в режиме master  Используется, только если для формата блока данных (FRF) выбран to Motorola SPI.  Значения: ■ 0x0 (SCLK_LOW): низкое состояние неактивного тактового сигнала ■ 0x1 (SCLK_HIGH): высокое состояние неактивного тактового сигнала Значение после сброса: SSI_DFLT_SCPOL	0x0
6	SCPH	Serial Clock Phase.  Фаза тактового сигнала, задает отношение тактового сигнала и slave select.  0x0 – захват данных происходит по первому фронту тактового сигнала  0x1 – происходит пропуск одного периода тактового сигнала после установки slave select, захват данных происходит по второму фронту тактового сигнала	0x0
5:4	FRF	Frame Format.  Выбор протокола передачи данных  0x0 – MOTOROLA_SPI 0x1 – TEXAS_SSP 0x2 – NS_MICROWIRE 0x3 – Резерв	0x0
3:0	DFS	Data Frame Size  Размер блока данных в 16-х битном режиме передачи данных.  Поле применимо только для контроллеров spi2 RO для контроллера spi0, spi1	0x0

### 12.2.2 Регистр CTRLR1

Регистр управления завершением последовательной передачи. Запись в регистр невозможна, если регистр SSIENR находится в активном состоянии.

Формат регистра CTRLR1 приведен в следующей таблице:

**Таблица 12.3. Формат регистра CTRLR1**

Разряды	Название	Описание	Сброс
31:16	RSVD_CTRLR1	CTRLR1 Резерв – только чтение	0x0



Разряды	Название	Описание	Сброс
15:0	NDF	<p>Number of Data Frames.</p> <p>Выбор количество блоков данных для приема, когда TMOD = 10 или TMOD = 11.</p> <p>Количество блоков данных принятых по последовательному интерфейсу равно NDF + 1 (Максимум 64 Кб)</p>	0x0

### 12.2.3 Регистр SSIENR

Регистр включения/отключения SPI.

Формат регистра SSIENR приведен в следующей таблице:

**Таблица 12.4. Формат регистра SSIENR**

Разряды	Название	Описание	Сброс
31:1	RSVD_SSIENR	SSIENR Резерв – только чтение.	0x0
0	SSI_EN	<p>SSI Enable.</p> <p>Включает и отключает все операции SPI. При отключении любые операции передачи данных прерываются немедленно. FIFO буферы приемника и передатчика очищаются. В активном состоянии, блокируется возможность записи в некоторые регистры управления.</p>	0x0

### 12.2.4 Регистр MWCR

Регистр управления направлением передачи данных в режиме полудуплекс Microwire. Запись в регистр невозможна, если регистр SSIENR находится в активном состоянии.

Формат регистра MWCR приведен в следующей таблице:

**Таблица 12.5. Формат регистра MWCR**

Разряды	Название	Описание	Сброс
31:3	RSVD_MWCR	MWCR Резерв – только чтение.	0x0

Разряды	Название	Описание	Сброс
2	MHS	Microwire Handshaking.  Включает/отключает busy/ready проверку, протокола Microwire. В активном состоянии SPI проверяет готовность целевого slave устройства, после передачи последнего бита данных, перед снятием BUSY статуса в регистре SR.	0x0
1	MDD	Microwire Control.  Определяет направление передачи слова данных в режиме Microwire.  0x0 – SPI передает слово данных 0x1 – SPI принимает слово данных	0x0
0	MWMOD	Microwire Transfer Mode.  Определяет тип передачи Microwire.  0x0 – одиночная Microwire передача 0x1 – последовательная передача Microwire	0x0

### 12.2.5 Регистр SER

Регистр управления slave-select сигналом. Регистр может быть установлен в 1 или 0, если SSI\_EN=0. Если SSI\_EN=1, регистр может быть установлен только в 1 (для задержки установки slave-select сигнала во время заполнения FIFO передатчика).

Формат регистра SER приведен в следующей таблице:

**Таблица 12.6. Формат регистра SER**

Разряды	Название	Описание	Сброс
31:4	RSVD_SER	SER Резерв – только чтение.	0x0

Разряды	Название	Описание	Сброс
3:0	SER	Slave Select Enable Flag.  Каждый бит этого поля отвечает за соответствующий сигнал slave-select(ss_x_n). Нужно отметить, что очищение любого бита в этом поле во время передачи данных не приведет к немедленному сбросу соответствующего slave-select сигнала, запись в это поле должна производиться перед началом передачи данных. Если не выбран режим вещания, то должен быть установлен только один бит этого поля.	0x0

### 12.2.6 Регистр BAUDR

Регистр определения частоты тактового сигнала последовательного интерфейса. Значение в этом регистре является делителем частоты SPI. Запись в регистр невозможна, если регистр SSIENR находится в активном состоянии.

Формат регистра BAUDR приведен в следующей таблице:

**Таблица 12.7. Формат регистра BAUDR**

Разряды	Название	Описание	Сброс
31:16	RSVD_BAUDR	BAUDR Резерв – Только чтение	0x0
15:0	SCKDV	SSI Clock Divider.  Младший бит этого регистра всегда равен 0 и доступен только для чтения – это гарантирует четность значения в этом регистре.  Если значение SCKDV = 0, то тактовый сигнал SPI будет отключен.  Частота тактового сигнала SPI определяется формулой:  $F_{SPI\_SCK} = F_{SYSCLK} / SCKDV$  Где SCDV – любое четное число от 2 до 65534	0x0

### 12.2.7 Регистр TXFTLR

Регистр управления порогом заполнения FIFO буфера передатчика.

Формат регистра TXFTLR приведен в следующей таблице:

**Таблица 12.8. Формат регистра TXFTLR**

Разряды	Название	Описание	Сброс
31:6	RSVD_TXFTLR	TXFTLR Резерв – только чтение.	0x0
5:0	TFT	<p>Transmit FIFO Threshold.</p> <p>Задаёт количество слов данных на котором FIFO буфер передатчика сгенерирует прерывание.</p> <p>При попытке записи значения, превышающего глубину FIFO – значение регистра останется неизменным. Когда количество слов данных в FIFO меньше или равно значению в этом поле, генерируется прерывание (ssi_txe_intr).</p>	0x0

## 12.2.8 Регистр RXFTLR

Регистр управления порогом заполнения FIFO буфера приемника.

Reset Value: 0x0

Формат регистра RXFTLR приведен в следующей таблице:

**Таблица 12.9. Формат регистра RXFTLR**

Разряды	Название	Описание	Сброс
31:6	RSVD_RXFTLR	RXFTLR Резерв – только чтение.	0x0
5:0	RFT	<p>Receive FIFO Threshold.</p> <p>Задаёт количество слов данных на котором FIFO буфер приемника сгенерирует прерывание.</p> <p>При попытке записи значения, превышающего глубину FIFO – значение регистра останется неизменным. Когда количество слов данных в FIFO больше или равно значению в этом поле, генерируется прерывание (ssi_rxe_intr).</p>	0x0

## 12.2.9 Регистр TXFLR

Регистр количества слов данных в FIFO передатчика.

Формат регистра TXFLR приведен в следующей таблице:

**Таблица 12.10. Формат регистра TXFLR**

Разряды	Название	Описание	Сброс
31:7	RSVD_TXFLR	TXFLR Резерв – только чтение.	0x0
6:0	TXTFL	Transmit FIFO Level.  Количество слов данных в FIFO буфере передатчика.	0x0

## 12.2.10 Регистр RXFLR

Регистр количества слов данных в FIFO приемника.

Формат регистра RXFLR приведен в следующей таблице:

**Таблица 12.11. Формат регистра RXFLR**

Разряды	Название	Описание	Сброс
31:7	RSVD_RXFLR	RXFLR Reserved bits - Read Only	0x0
6:0	RXTFL	Receive FIFO Level.  Количество слов данных в FIFO буфере приемника.	0x0

## 12.2.11 Регистр SR

Регистр индикации статуса передачи данных, статуса FIFO и любых ошибок приема/передачи которые могут возникнуть.

Формат регистра SR приведен в следующей таблице:

**Таблица 12.12. Формат регистра SR**

Разряды	Название	Описание	Сброс
31:7	RSVD_SR	SR Резерв – только чтение.	0x0
6	DCOL	Data Collision Error.  Ошибка является указателем того, что последняя передача данных была прервана до завершения. Бит сбрасывается после чтения.	0x0
5	RSVD_TXE	TXE Резерв – только чтение	0x0
4	RFF	Receive FIFO Full.  FIFO приемника полон, бит сбрасывается, если в буфере есть свободные места.	0x0

Разряды	Название	Описание	Сброс
3	RFNE	Receive FIFO Not Empty.  FIFO приемника не пуст, бит устанавливается если в буфере есть хотя бы одно слово данных. Сбрасывается если буфер пуст.	0x0
2	TFE	Transmit FIFO Empty.  FIFO буфер передатчика пуст, сбрасывается если в буфере есть хотя бы одно слово данных.	0x1
1	TFNF	Transmit FIFO Not Full.  FIFO передатчика не полон, бит сбрасывается, если буфер полон.	0x1
0	BUSY	SSI Busy Flag.  0x1 – идет передача SPI 0x0 – SPI в состоянии ожидания или выключен	0x0

### 12.2.12 Регистр IMR

Регистр маскирования прерываний.

Формат регистра IMR приведен в следующей таблице:

**Таблица 12.13. Формат регистра IMR**

Разряды	Название	Описание	Сброс
31:6	RSVD_IMR	IMR Резерв – только чтение.	0x0
5	MSTIM	Multi-Master Contention Interrupt	0x1
4	RXFIM	Receive FIFO Full Interrupt Mask	0x1
3	RXOIM	Receive FIFO Overflow Interrupt Mask	0x1
2	RXUIM	Receive FIFO Underflow Interrupt Mask	0x1
1	TXOIM	Transmit FIFO Overflow Interrupt Mask	0x1
0	TXEIM	Transmit FIFO Empty Interrupt Mask	0x1

### 12.2.13 Регистр ISR

Регистр статуса прерываний после маскирования.

Формат регистра ISR приведен в следующей таблице:

**Таблица 12.14. Формат регистра ISR**

Разряды	Название	Описание	Сброс
31:6	RSVD_ISR	ISR Reserved bits - Read Only	0x0
5	MSTIS	Multi-Master Contention Interrupt Status.	0x0
4	RXFIS	Receive FIFO Full Interrupt Status	0x0
3	RXOIS	Receive FIFO Overflow Interrupt Status	0x0
2	RXUIS	Receive FIFO Underflow Interrupt Status	0x0
1	TXOIS	Transmit FIFO Overflow Interrupt Status	0x0
0	TXEIS	Transmit FIFO Empty Interrupt Status	0x0

## 12.2.14 Регистр RISR

Регистр статуса прерываний до маскирования.

Формат регистра RISR приведен в следующей таблице:

**Таблица 12.15. Формат регистра RISR**

Разряды	Название	Описание	Сброс
31:6	RSVD_RISR	RISR Reserved bits - Read Only	0x0
5	MSTIR	Multi-Master Contention Raw Interrupt Status.	0x0
4	RXFIR	Receive FIFO Full Raw Interrupt Status	0x0
3	RXOIR	Receive FIFO Overflow Raw Interrupt Status	0x0
2	RXUIR	Receive FIFO Underflow Raw Interrupt Status	0x0
1	TXOIR	Transmit FIFO Overflow Raw Interrupt Status	0x0
0	TXEIR	Transmit FIFO Empty Raw Interrupt Status	0x0

## 12.2.15 Регистр TXOICR

Регистр сброса прерывания: “Переполнен FIFO передатчика”.

Формат регистра TXOICR приведен в следующей таблице:

**Таблица 12.16. Формат регистра TXOICR**

Разряды	Название	Описание	Сброс
31:1	RSVD_TXOICR	TXOICR Резерв – только чтение.	0x0

Разряды	Название	Описание	Сброс
0	TXOICR	Clear Transmit FIFO Overflow Interrupt.  Значение поля отображает статус прерывания, Чтение регистра сбросит прерывание ssi_txo_intr. Запись не имеет эффекта.	0x0

## 12.2.16 Регистр RXOICR

Регистр сброса прерывания: “Переполнен FIFO приемника”.

Формат регистра RXOICR приведен в следующей таблице:

**Таблица 12.17. Формат регистра RXOICR**

Разряды	Название	Описание	Сброс
31:1	RSVD_RXOICR	R XOICR Резерв – только чтение.	0x0
0	R XOICR	Clear Receive FIFO Overflow Interrupt.  Значение поля отображает статус прерывания, Чтение регистра сбросит прерывание ssi_rxo_intr. Запись не имеет эффекта.	0x0

## 12.2.17 Регистр RXUICR

Регистр сброса прерывания: “FIFO приемника пуст”

Формат регистра RXUICR приведен в следующей таблице:

**Таблица 12.18. Формат регистра RXUICR**

Разряды	Название	Описание	Сброс
31:1	RSVD_RXUICR	R XUICR Резерв – только чтение.	0x0
0	R XUICR	Clear Receive FIFO Underflow Interrupt.  Значение поля отображает статус прерывания, Чтение регистра сбросит прерывание ssi_rxu_intr. Запись не имеет эффекта.	0x0



### 12.2.18 Регистр MSTICR

Регистр сброса прерывания: “Возможный конфликт master устройств”

Формат регистра MSTICR приведен в следующей таблице:

**Таблица 12.19. Формат регистра MSTICR**

Разряды	Название	Описание	Сброс
31:1	RSVD_MSTICR	MSTICR Резерв – только чтение.	0x0
0	MSTICR	Clear Multi-Master Contention Interrupt.  Значение поля отображает статус прерывания, Чтение регистра сбросит прерывание ssi_mst_intr. Запись не имеет эффекта.	0x0

### 12.2.19 Регистр ICR

Регистр сброса прерываний.

Формат регистра ICR приведен в следующей таблице:

**Таблица 12.20. Формат регистра ICR**

Разряды	Название	Описание	Сброс
31:1	RSVD_ICR	ICR Резерв – только чтение.	0x0
0	ICR	Clear Interrupts.  Значение поля отображает наличие любого из следующих прерываний : ssi_txo_intr ssi_rxu_intr ssi_rxo_intr ssi_mst_intr  Чтение сбрасывает все вышеперечисленные прерывание, запись не имеет эффекта.	0x0

### 12.2.20 Регистр DMACR

Регистр управления аппаратным интерфейсом DMA.

Формат регистра DMACR приведен в следующей таблице:

**Таблица 12.21. Формат регистра DMACR**

Разряды	Название	Описание	Сброс
31:2	RSVD_DMACR	DMACR Резерв только чтение.	0x0
1	TDMAE	Transmit DMA Enable.  Бит включения/отключения DMA канала FIFO буфера передатчика.	0x0
0	RDMAE	Receive DMA Enable.  Бит включения/отключения DMA канала FIFO буфера приемника.	0x0

### 12.2.21 Регистр DMATDLR

Регистр управления запросом к DMA по уровню заполнения FIFO буфера передатчика.

Формат регистра DMATDLR приведен в следующей таблице:

**Таблица 12.22. Формат регистра DMATDLR**

Разряды	Название	Описание	Сброс
31:6	RSVD_DMATDLR	DMATDLR Резерв – только чтение.	0x0
5:0	DMATDL	Transmit Data Level.  Поле управления запросом DMA(dma_tx_req), по уровню заполнения FIFO передатчика. Запрос формируется если количество слов данных в буфере передатчика меньше или равно DMATDL и TDMAE = 1.	0x0

### 12.2.22 Регистр DMARDLR

Регистр управления запросом к DMA по уровню заполнения FIFO буфера приемника.

Формат регистра DMARDLR приведен в следующей таблице:

**Таблица 12.23. Формат регистра DMARDLR**

Разряды	Название	Описание	Сброс
31:6	RSVD_DMARDLR	DMARDLR Резерв – только чтение.	0x0

Разряды	Название	Описание	Сброс
5:0	DMARDL	Receive Data Level.  Поле управления запросом DMA(dma_rx_req), по уровню заполнения FIFO приемника. Запрос формируется если количество слов данных в буфере приемника больше или равно DMARDL + 1 и RDMAE = 1.	0x0

### 12.2.23 Регистр IDR

Идентификационный регистр

Формат регистра IDR приведен в следующей таблице:

**Таблица 12.24. Формат регистра IDR**

Разряды	Название	Описание	Сброс
31:0	IDCODE	Identification code.	0xffffffff

### 12.2.24 Регистр SSI\_VERSION\_ID

Регистр версии SPI

Формат регистра SSI\_VERSION\_ID приведен в следующей таблице:

**Таблица 12.25. Формат регистра SSI\_VERSION\_ID**

Разряды	Название	Описание	Сброс
31:0	SSI_COMP_VERSION	HEX представление, версии SPI. Состоит из ASCII значения для каждого числа.	0x3430312a

### 12.2.25 Регистр DR0 – DR35

Регистр данных SPI.

Чтение из регистра считывает данные из FIFO буфера приемника. Записанные в регистр данные помещаются в FIFO буфер передатчика.

Запись возможно только, если SSI\_EN=1.

Регистр сбрасывается при SSI\_EN = 0.

Регистр резервирует 36 адресов, для поддержки АНВ\_BURST транзакций, запись в любой регистр равноценна помещению слов данных в FIFO передатчика, чтение любого из них равноценно чтению данных из FIFO приемника. Данные FIFO буферов не адресуются.

Формат регистра DR0 приведен в следующей таблице:

**Таблица 12.26. Формат регистра DR0**

Разряды	Название	Описание	Сброс
31:0	DR	Data Register	0x0

## 12.2.26 Регистр RSVD

Резерв

Формат регистра RSVD приведен в следующей таблице:

**Таблица 12.27. Формат регистра RSVD**

Разряды	Название	Описание	Сброс
31:0	RSVD	RSVD Резерв	0x0

## 12.3 Функционирование

### 12.3.1 Режим обмена

Выбор режима обмена происходит посредством битов TMOD регистра CTRLR0. В

зависимости от значения этих битов, возможны следующие режимы обмена:

#### Прием и передача:

Когда TMOD = 0x0, возможны прием и передача данных. Обмен данными происходит в соответствии с выбранным протоколом (поле FRF регистра CTRL0) данные поступают из FIFO передатчика. От slave-устройства данные поступают в FIFO приемника.

#### Только передача:

Когда TMOD = 0x1, принимаемые от slave-устройства данные не поступают в FIFO приемника. Передача данных происходит в обычном режиме по выбранному последовательному протоколу. При использовании этого режима необходимо маскировать прерывания от приемника.

#### Только прием:

Когда  $TMOD = 0x2$ , данные не записываются в FIFO передатчика. Прием данных происходит в обычном режиме по выбранному протоколу. При использовании этого режима необходимо маскировать прерывания от передатчика.

### Чтение EEPROM:

Когда  $TMOD = 0x3$ , передача данных используется для передачи кода операции и/или адреса EEPROM-устройства. Обычно это занимает 3 кадра данных (8 бит - старший адрес, 8 бит - младший адрес, 8 бит - код операции). Во время передачи кода операции и адреса не происходит приема данных (игнорируются данные с линии MISO). Порт продолжает передачу данных до тех пор, пока FIFO передатчика не станет пустым. Поэтому перед началом передачи FIFO передатчика должно иметь только необходимые данные (адрес и код операции), иначе происходит потеря данных, читаемых из EEPROM-устройства. После того как FIFO передатчика станет пустым (будет отправлена вся контрольная информация), станет доступен прием данных с линии MISO. Последовательный обмен будет продолжаться до тех пор, пока число принятых слов данных не превысит на 1 значение бит NDF регистра CTRLR1. Режим недоступен для SSP протокола.

## 12.3.2 Тактирование SPI контроллера

Максимальная частота интерфейса SPI ( $SPIx\_SCK$ ) определяется как половина частоты контроллера SPI ( $SYSCLK$ )

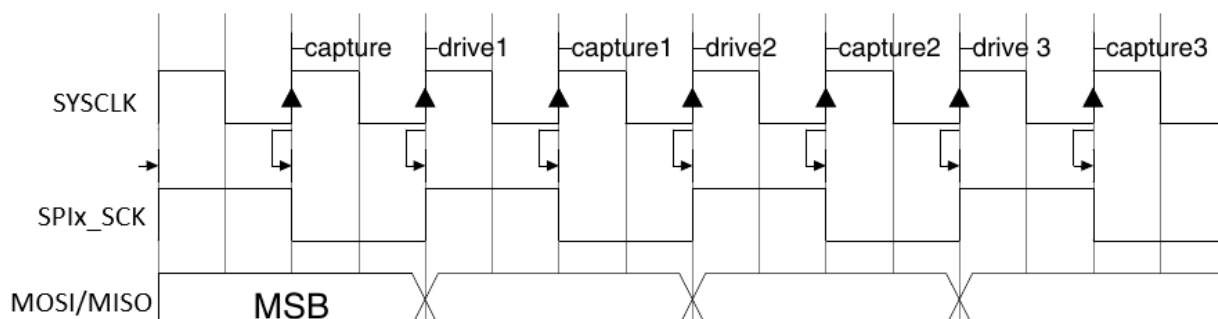


Рисунок 12.1 Максимальное отношение частот  $SPIx\_SCK/SYSCLK$

В общем виде частота интерфейса определяется как  $F_{SPIx\_SCK} = \frac{F_{SYSCLK}}{SCKDV}$

$SCKDV$  – поле регистра BAUDR, может принимать значение от 0 до 65534, если значение поля 0 то выходная частота отсутствует.

## 12.3.3 Прерывания SPI

SPI оснащен комбинированным сигналом прерывания с поддержкой маскирования, прерывание возникает в одном из следующих случаев:

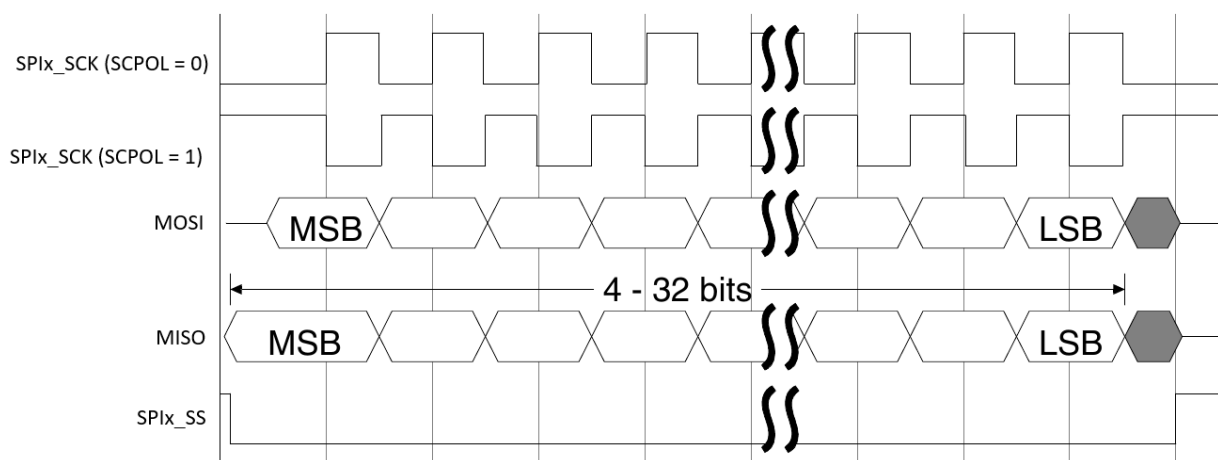
- Transmit FIFO empty Interrupt (`spi_txe_intr`) – уровень FIFO передатчика меньше или равен установленному значению (поле TFT регистра TXFTLR)
- Transmit FIFO overflow Interrupt (`spi_txo_int`) – устанавливается в случае переполнения FIFO передатчика. Снимается чтением TXOICR регистра.
- Receive FIFO Full Interrupt (`ssi_rxf_intr`) – уровень FIFO приемника больше или равен  $RXT + 1$  (регистра RXFLTR), снимается по уменьшению уровня в FIFO приемника.
- Receive FIFO Overflow Interrupt (`ssi_rxo_intr`) – переполнение FIFO приемника, приходящие данные не будут записаны, снимается по чтению RXOICR регистра
- Receive FIFO Underflow Interrupt (`ssi_rxu_intr`) – устанавливается при чтении из пустого FIFO приемника, снимается чтением RXUICR регистра
- Multi-Master Contention Interrupt (`ssi_mst_intr`) – устанавливается в случае обращения другого мастер устройства к контроллеру SPI как к ведомому устройству, что может сигнализировать о конфликтах на SPI шине. Снимается чтением MSTICR регистра.

### 12.3.4 Motorola Serial Peripheral Interface (SPI)

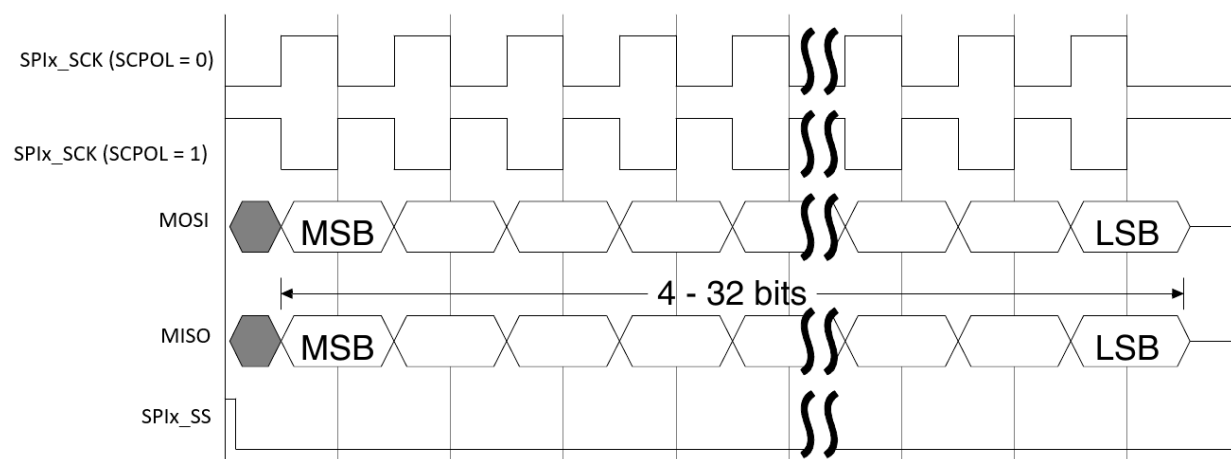
Протокол SPI имеет два определяющих параметра:

SCPOL - исходный уровень сигнала синхронизации. Если SCPOL=0, то линия синхронизации до начала цикла передачи и после его окончания имеет низкий уровень иначе, если SCPOL=1, - высокий.

SCPH - фаза синхронизации. От этого параметра зависит, в какой последовательности выполняется установка и выборка данных (если SCPH=0, то по переднему фронту в цикле синхронизации будет выполняться выборка данных, а затем, по заднему фронту, - установка данных; если же SCPH=1, то установка данных будет выполняться по переднему фронту в цикле синхронизации, а выборка - по заднему).



**Рисунок 12.2 Пример последовательного обмена по протоколу SPI с параметром SCPH = 0**

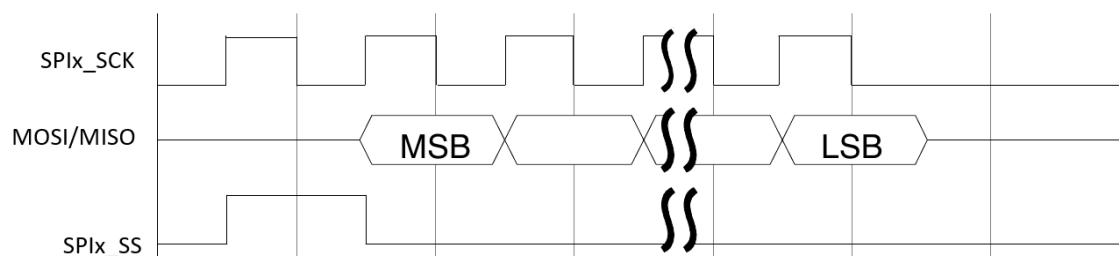


**Рисунок 12.3** Пример последовательного обмена по протоколу SPI с параметром SCPH = 1

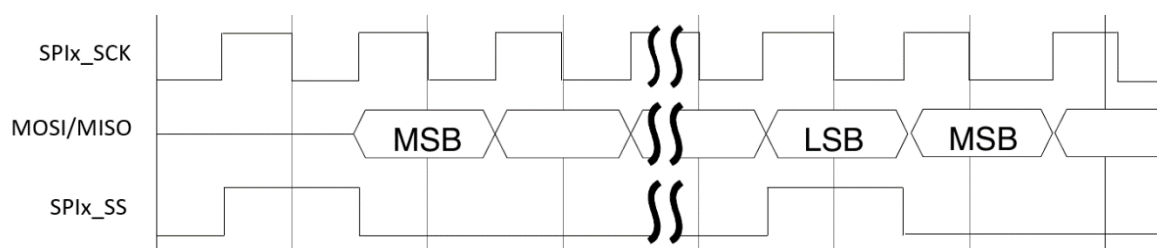
### 12.3.5 Texas Instruments Synchronous Serial Protocol (SSP)

Обмен по протоколу SSP начинается с установки в 1 индикатора линии SS на 1 такт. Передача/прием данных начинается на следующий такт, после перехода сигнала SS в 0. Установка данных всегда происходит по переднему фронту сигнала синхронизации, а выборка – по заднему. Длина кадра данных может быть установлена от 4 до 32 бит.

Возможна передача как одиночного, так и множественных кадров.



**Рисунок 12.4** Пример передачи одного кадра по протоколу SSP



**Рисунок 12.5** Пример передачи множественных кадров по протоколу SSP

### 12.3.6 National Semiconductor Microwire

Обмен данными по протоколу Microwire начинается по заднему фронту сигнала SlaveSelect. Спустя половину периода отправляется первый бит контрольного слова в линию MOSI. Длина контрольного слова может изменяться в диапазоне от 1 до 16 бит. Длина контрольного слова настраивается через биты CFS (CTRLR0[15:12]). Во время передачи контрольного слова линия MISO находится в высокоимпедансном состоянии. Направление передачи слова данных контролируется битом MDD (MWCR[1]). Значение MDD = 0 указывает на то, что порт SSI принимает данные от slave-устройства; Значение MDD = 1, указывает на передачу данных slave-устройству. В случае приема данных, через такт после передачи LSB контрольного слова, slave-устройство выдает бит 0 в линию MISO, и после этого следует 4-16 бит слова данных. Данные устанавливаются по заднему фронту линии синхронизации, а выборка по переднему. В случае передачи данных, данные идут сразу за контрольным словом. Сигнал Slave-Select остается активно-низким на протяжении всей передачи и переходит через полтакта после окончания передачи данных.

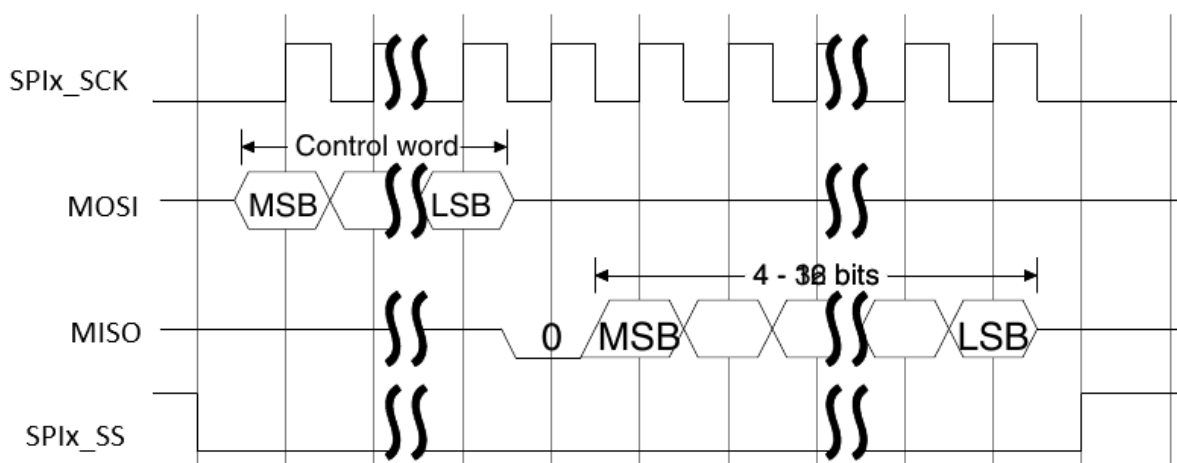


Рисунок 12.6 Пример передачи по протоколу Microwire (MDD = 0)



## 13. I2C: контроллер I2C

В микросхеме реализованы два блока I2C.

### 13.1 Основные особенности

- Двух проводной последовательный интерфейс (SCL, SDA)
- Три скорости работы:
  - Standard (от 0 до 100 Кбит/с)
  - Fast mode ( $\leq 400$  Кбит/с)
  - Fast mode plus ( $\leq 1000$  Кбит/с, только для выводов с поддержкой режима I2C, см. раздел «IOCTR: Многофункциональные порты ввода/вывода»)
- Поддерживает Multi-master режим (синхронизация тактовых частот, процедура арбитража при передаче данных)
- Функционирование в режимах Master или Slave
- Поддерживает 7 и 10-битную адресацию и возможность динамического переключения между ними
- Режим передачи Bulk
- Игнорирование адреса CBUS
- Буферы приема передачи по 8 слов каждое
- Поддерживает режимы работы по прерыванию и по опросу
- Аппаратный интерфейс взаимодействия с DMA контроллерами
- Программируемый параметр SDA hold time (tHD;DAT)
- Функция очистка шины

## 13.1 Описание функционирования

### 13.1.1 Последовательности Start и Stop

Когда шина I<sup>2</sup>C неактивна (находится в состоянии IDLE), оба сигнала SCL и SDA идут высоким уровнем. Когда master-устройство инициализирует передачу, контроллер отправляет последовательность Start: изменение уровня сигнала SDA с 1 на 0, при высоком уровне сигнала SCL. Когда master-устройство завершает передачу, контроллер отправляет последовательность Stop: изменение уровня сигнала SDA с 0 на 1, при высоком уровне сигнала SCL. Во всех остальных случаях значение сигнала SDA остается неизменным при высоком уровне сигнала SCL. Последовательности Start и Stop представлены на Рисунок 13.1.

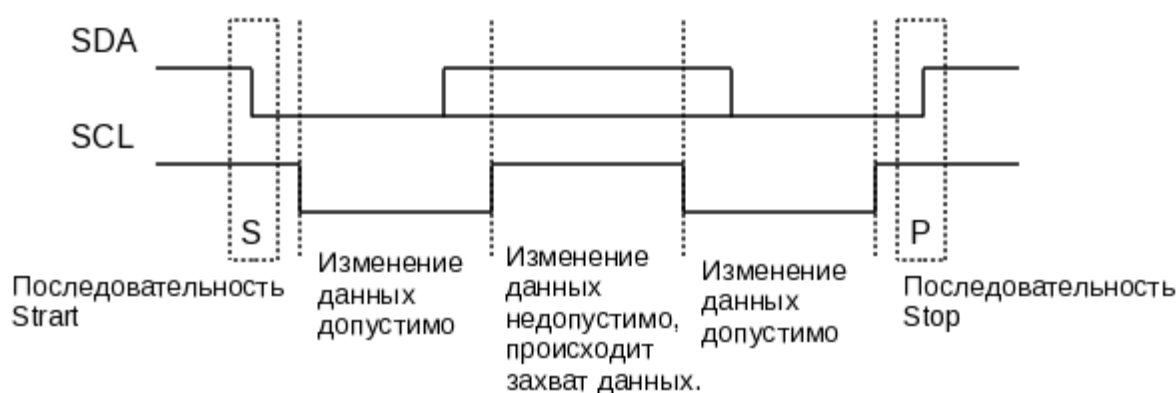
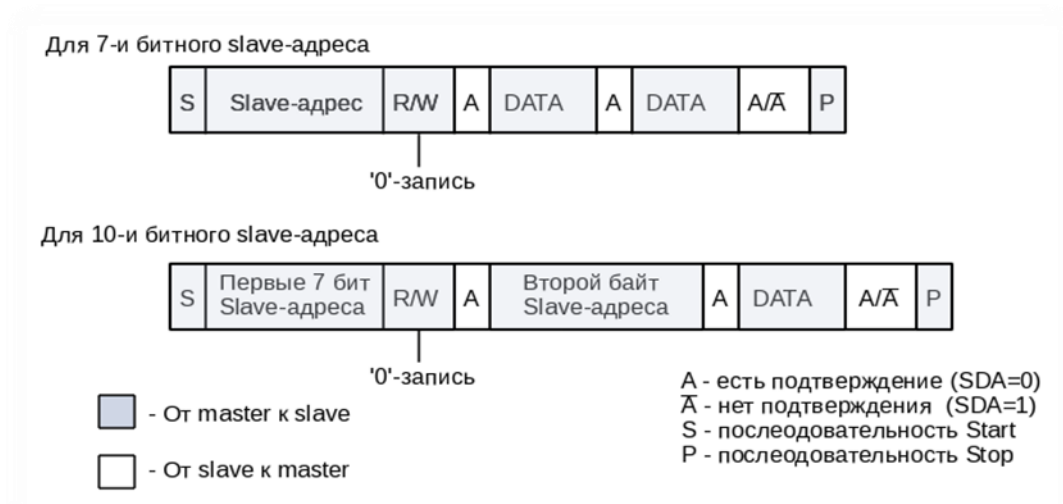


Рисунок 13.1. Последовательности Start и Stop

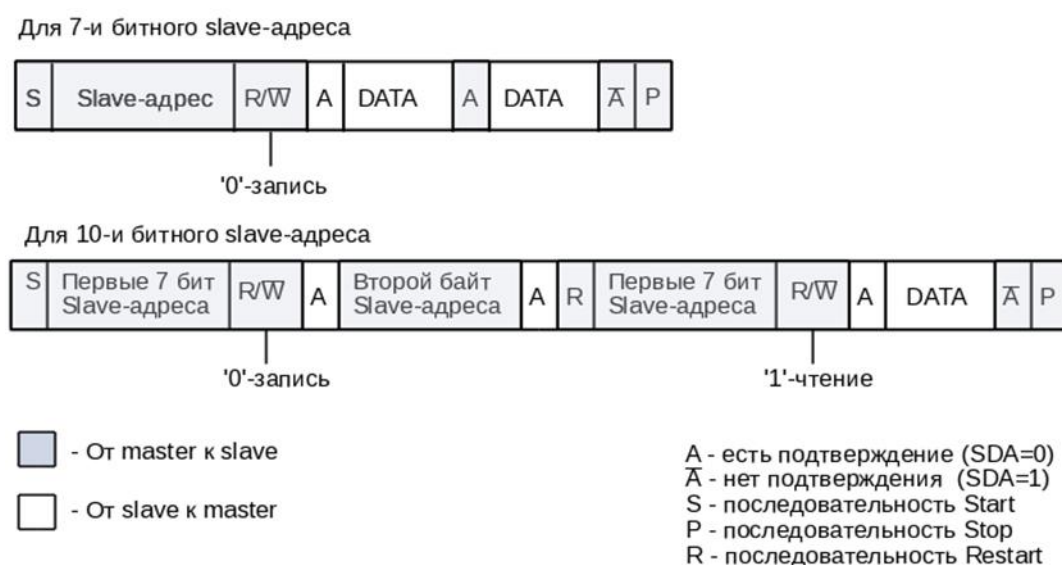
### 13.1.2 Протокол передачи и приема данных

Master-передатчик и slave-приемник. Данные передаются побайтово, число байтов в одной передаче не ограничено. После того, как master-устройство отправляет slave-устройству адрес с битом R/W или байт данных, slave-приемник должен выдать импульс подтверждения ACK в линию SDA. Если slave-приемник не отвечает импульсом ACK, master-устройство прерывает передачу и отправляет последовательность Stop. Структура обмена данными между Master-передатчиком и Slave-приемником представлена на Рисунок 13.2.



**Рисунок 13.2 Обмен данными между Master-передатчиком и Slave-приемником**

Если master-устройство принимает данные, как показано на Рисунок 13.3, тогда оно подтверждает принятые от slave-устройства данные импульсом ACK и ожидает следующий байт. В случае, если master-приемник желает прекратить передачу, то по окончании приема текущего байта он выдает сигнал NACK в линию SDA а затем отправляет последовательность Stop. Структура обмена между Master-приемником и Slave-передатчиком представлена на Рисунок 13.3.



**Рисунок 13.3 Обмен данными между Slave-передатчиком и Master-приемником**

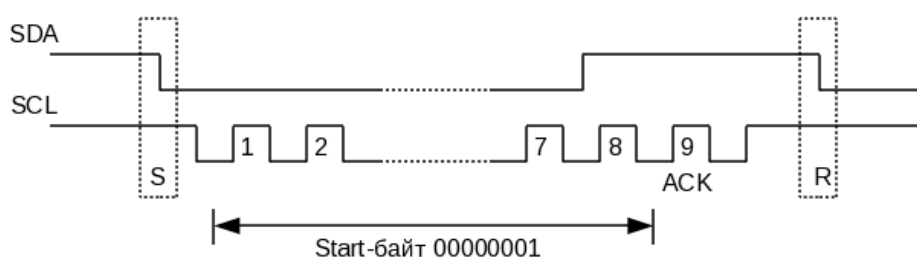
Отправляя последовательность Restart вместо последовательности Stop, контроллер, работающий в режиме master, меняет направление передачи данных без утраты контроля шины.

### 13.1.3 Протокол START BYTE

Протокол START BYTE нужен для взаимодействия контроллера I<sup>2</sup>C с устройствами, не имеющими аппаратного модуля I<sup>2</sup>C. Использование START BYTE протокола имеет смысл только тогда, когда контроллер работает в режиме master.

Протокол заключается в следующем: для инициализации обмена контроллер, отправив последовательность Start, отправляет START-байт «00000001». Это необходимо для того, чтобы устройство, не имеющее аппаратного модуля I<sup>2</sup>C, могло опрашивать шину с более низкой частотой до момента обнаружения одного из нулей START BYTE. Структура протокола START BYTE представлена на Рисунок 13.4.

Полная последовательность действий контроллера:



**Рисунок 13.4. START BYTE**

1. Отправляет последовательность START.
2. Отправляет Start-байт («00000001»).
3. Отправляет импульс ACK.
4. Ни одно slave-устройство не отвечает импульсом ACK.
5. Отправляет последовательность RESTART.

Аппаратный приемник не отвечает импульсом ACK, поскольку Start-байт является зарезервированным адресом.

### 13.1.4 Multi-master арбитраж и синхронизация

Возможно подключение нескольких master-устройств к шине I<sup>2</sup>C. Но если одно master-устройство контролирует шину в конкретный момент времени, то никакое другое master-устройство не сможет получить контроль над шиной, до тех пор пока первое master-устройство не создаст на шине состояние IDLE, отправив последовательность Stop. При этом, возможна ситуация, при которой два или более master-устройств пытаются одновременно получить контроль над шиной в состоянии IDLE, генерируя последовательность Start. В этом случае, они подвергаются арбитражу, структура которого представлена на Рисунок 13.5.

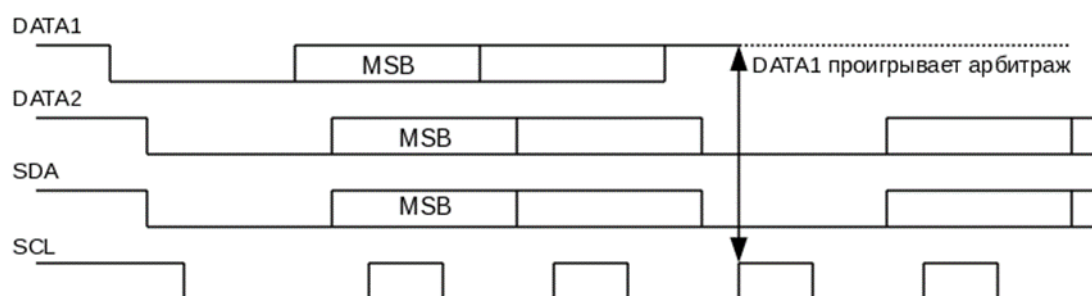


Рисунок 13.5 Арбитраж

Процедура арбитража начинается на линии SDA, когда линия SCL идет высоким уровнем. Master-устройство, которое отправляет 1, когда другое master-устройство отправляет «0», проигрывает арбитраж и прекращает передачу данных. Master-устройство, проигравшее арбитраж, продолжает генерировать тактовый сигнал до конца текущего байта. Если оба master-устройства адресовали передачу одному и тому же slave-устройству, то арбитраж продолжается в фазе передачи данных.

Синхронизация заключается в том, что во время арбитража тактовые сигналы master-устройств складываются по «И» до окончания байта, на котором был завершен арбитраж.

### 13.1.5 Фильтрация

Фильтрующая логика основана на счетчиках, которые отслеживают состояние входных сигналов (SDA и SCL). Запуск счетчика происходит, когда соответствующий ему сигнал меняет значение. Сигнал фиксируется контроллером, когда его длительность становится равна установленному в регистре IC\_\*\_SPLKEN числу тактов фильтрующего счетчика. Принцип работы механизма фильтрации представлен на Рисунок 13.6.

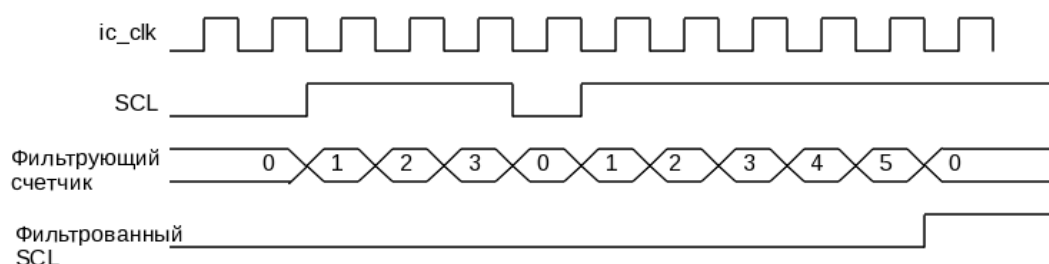


Рисунок 13.6. Фильтрация

### 13.1.6 Расчет частоты SCL и минимальные значения счетчиков делителя частоты

Когда контроллер работает в режиме master, должны выполняться следующие условия:

- значения регистров IC\_SS\_SCL\_LCNT и IC\_FS\_SCL\_LCNT должны быть больше чем IC\_FS\_SPLKEN+7;

- значения регистров IC\_SS\_SCL\_HCNT и IC\_FS\_SCL\_HCNT должны быть больше чем IC\_FS\_SPKLEN+5;
- значение регистра IC\_HS\_SCL\_LCNT должно быть выше, чем IC\_HS\_SPKLEN+7.

Временные параметры обмена представлены на Рисунок 13.7.

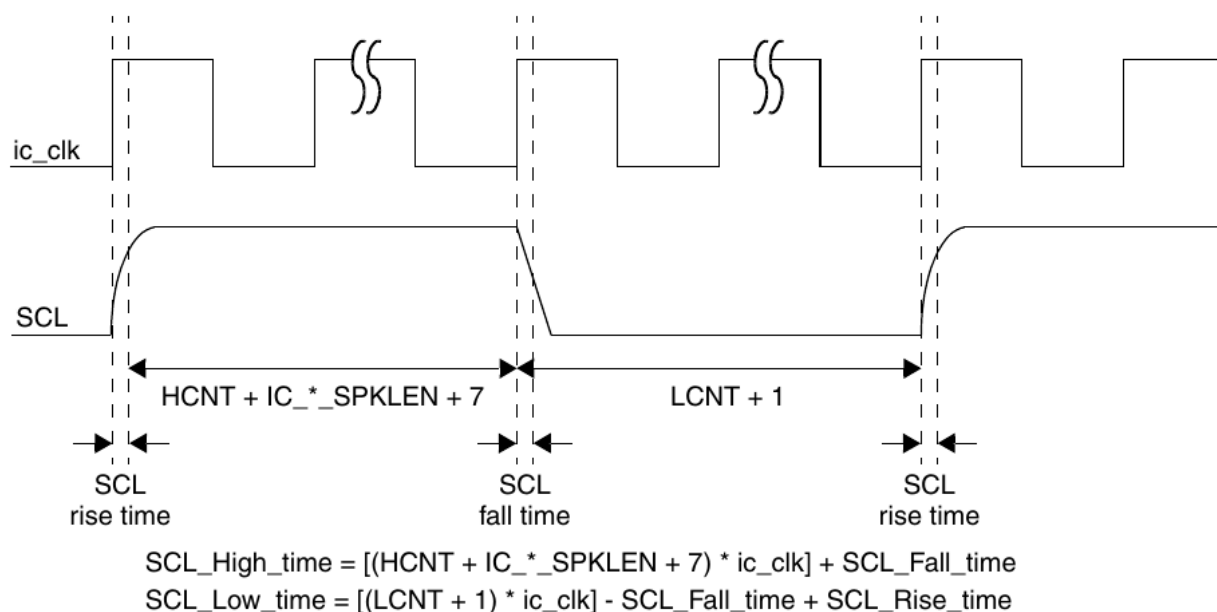


Рисунок 13.7. Временные параметры обмена

## 13.2 Регистры

В Таблица 13.1 приведен перечень программно-доступных регистров контроллера I<sup>2</sup>C.

Таблица 13.1. Перечень программно-доступных регистров контроллера I<sup>2</sup>C

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
IC_CON	Регистр управления	W/R (R для 4-ого бита)	0x7F	0x0
IC_TAR	Регистр адреса абонента	W/R	0x1055	0x4
IC_SAR	Регистр slave-адреса.	W/R	0x55	0x8
IC_HS_MADDR	Регистр кода адреса мастера для high speed режима.	W/R	0x1	0xC
IC_DATA_CMD	Регистр управления передачей.	W/R	0x0	0x10
IC_SS_SCL_HCNT	Старший регистр счетчика предделителя частоты для standard-speed режима.	W/R	0x190	0x14
IC_SS_SCL_LCNT	Младший регистр счетчика предделителя частоты для standard-speed режима.	W/R	0x1d6	0x18

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
IC_FS_SCL_HCNT	Старший регистр счетчика предделителя частоты для fast-speed режима.	W/R	0x3c	0x1C
IC_FS_SCL_LCNT	Младший регистр счетчика предделителя частоты для fast-speed режима.	W/R	0x82	0x20
IC_HS_SCL_HCNT	Старший регистр счетчика предделителя частоты для high-speed режима.	W/R	0x6	0x24
IC_HS_SCL_LCNT	Младший регистр счетчика предделителя частоты для high-speed режима.	W/R	0x10	0x28
IC_INTR_STAT	Регистр статуса прерывания.	R	0x0	0x2C
IC_INTR_MASK	Регистр маски прерывания	W/R	0x8ff	0x30
IC_RAW_INTR_STAT	Регистр статуса необработанного прерывания	R	0x0	0x34
IC_RX_TL	Регистр порога заполнения FIFO приемника.	W/R	0x0	0x38
IC_TX_TL	Регистр порога заполнения FIFO передатчика.	W/R	0x0	0x3C
IC_CLT_INTR	Регистр сброса прерываний.	R	0x0	0x40
IC_CLR_RX_UNDER	Регистр сброса прерывания RX_UNDER	R	0x0	0x44
IC_CLR_RX_OVER	Регистр сброса прерывания RX_OVER	R	0x0	0x48
IC_CLR_TX_OVER	Регистр сброса прерывания TX_OVER	R	0x0	0x4C
IC_CLR_RD_REQ	Регистр сброса прерывания RD_REQ	R	0x0	0x50
IC_CLR_TX_ABRT	Регистр сброса прерывания TX_ABR	R	0x0	0x54
IC_CLR_RX_DONE	Регистр сброса прерывания RX_DONE	R	0x0	0x58
IC_CLR_ACTIVITY	Регистр сброса прерывания ACTIVITY	R	0x0	0x5C
IC_CLR_STOP_DET	Регистр сброса прерывания STOP_DET	R	0x0	0x60
IC_CLR_START_DET	Регистр сброса прерывания START_DET	R	0x0	0x64
IC_CLR_GEN_CALL	Регистр сброса прерывания GEN_CALL	R	0x0	0x68
IC_ENABLE	Регистр включения шины.	W/R	0x0	0x6C
IC_STATUS	Регистр статуса шины.	R	0x6	0x70

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
IC_TXFLR	Регистр уровня FIFO передатчика	R	0x0	0x74
IC_RXFLR	Регистр уровня FIFO приемника	R	0x0	0x78
IC_SDA_HOLD	Регистр времени удержания SDA.	W/R	0x1	0x7C
IC_TX_ABRT_SOURCE	Регистр статуса обрыва передачи.	R	0x0	0x80
IC_SLV_DATA_NACK_ONLY	Регистр генерирования SLV_DATA_NACK	W/R	0x0	0x84
IC_DMA_CR	Регистр контроля DMA интерфейса.	W/R	0x0	0x88
IC_DMA_TDLR	Регистр передачи данных через DMA интерфейс.	W/R	0x0	0x8C
IC_DMA_RDLR	Регистр приема данных через DMA интерфейс.	W/R	0x0	0x90
IC_SDA_SETUP	Регистр установки SDA	W/R	0x64	0x94
IC_ACK_GENERAL_CALL	Регистр вызова общего ACK	W/R	0x1	0x98
IC_ENABLE_STATUS	Регистр статуса включения	R	0x0	0x9C
IC_FS_SPKLEN	Регистр управления фильтрацией для standard-speed и fast-speed режимов	W/R	0x5	0xA0
IC_HS_SPKLEN	Регистр управления фильтрацией для high-speed режима.	W/R	0x1	0xA4

### 13.2.1 Регистр IC\_CON

Регистр управления.

Этот регистр может быть доступен для записи, только при выключенном контроллере I<sup>2</sup>C (IC\_ENABLE=0).

Формат регистра IC\_CON приведен в Таблица 13.2.

**Таблица 13.2. Формат регистра IC\_CON**

номер бита	Условное обозначение	Назначение
0	MASTER_MODE	Этот бит управляет включением режима master.  0: Режим master выключен. 1: Режим master включен. Перед установкой этого бита в 1, нужно убедиться, что бит 6 также установлен в 1.
1:2	SPEED	Режим скорости передачи данных. Настройка только для контроллера, работающего в режиме master.  1: Standard-speed (0 to 100 Kб/с) 2: Fast-speed (≤ 400 Kб/с) 3: High-speed (≤ 3.4 Mб/с)



номер бита	Условное обозначение	Назначение
3	IC_10BITADDR_SLAVE	<p>Когда контроллер работает в режиме slave, этот бит выбирает тип адреса контроллера (7-и или 10-и битный адрес).</p> <p>0: 7-и битный адрес. Контроллер игнорирует обращения по 10-и битным адресам, при этом учитываются только 7 младших битов регистра IC_SAR.</p> <p>1: 10-и битный адрес. Контроллер отвечает только на 10-и битный адрес, который соответствует значению регистра IC_SAR.</p>
4	IC_10BITADDR_MASTER_rd_only	<p>Только для чтения. Бит отображает тип текущего адреса удаленного slave-устройства, к которому обращается контроллер, работающий в режиме master:</p> <p>0: 7-и битный адрес 1: 10-и битный адрес</p> <p>Управление типом адреса производится битом IC_TAR(12).</p>
5	IC_RESTART_EN	<p>Бит определяет, может ли контроллер, работающий в режиме master, отправить последовательность RESTART.</p> <p>0: не может отправить RESTART. 1: может отправить RESTART.</p> <p>Если IC_RESTART_EN=0, контроллер не может выполнять следующие функции:</p> <ul style="list-style-type: none"> <li>• Отправлять START-байт.</li> <li>• Выполнять любые обмены в режиме high-speed.</li> <li>• Изменять направление обмена в режиме комбинированного обмена.</li> <li>• Выполнять операции чтения с 10-и битным адресом.</li> </ul> <p>В этом случае, вместо последовательности RESTART, будут отправлены последовательности STOP - START, что приведет к обрыву комбинированной передачи. При этом установится бит TX_ABORT (бит 6 регистра IC_RAW_INTR_STAT)</p>
6	IC_SLAVE_DISABLE	<p>Этот бит управляет выключением режима slave для контроллера. Если этот бит установлен, контроллер функционирует только как master</p> <p>0: slave включен 1: slave выключен</p> <p>Перед установкой этого бита в 0, нужно убедиться, что бит 0 также установлен в 0.</p>
3:31	—	Резерв

### 13.2.2 Регистр IC\_TAR

Регистр адреса удаленного slave-устройства.

Для того, чтобы регистр был доступен по записи, необходимо чтобы соблюдалось одно из следующих условий:

- контроллер выключен (IC\_ENABLE=0);
- контроллер включен (IC\_ENABLE = 1), И контроллер не выполняет каких-либо операций в режиме master (IC\_STATUS[5]=0), И включен режим мастера (IC\_CON[0]=1), И FIFO передатчика не содержит данных (IC\_STATUS[2]=1).

Формат регистра IC\_TAR приведен в Таблица 13.3.

**Таблица 13.3. Формат регистра IC\_TAR**

Номер бита	Условное обозначение	Назначение
0:9	IC_TAR	Адрес удаленного slave-устройства, для любых транзакций контроллера, работающего в режиме master. Когда происходит передача General Call, то содержимое этих битов игнорируется.
10	GC_OR_START	Если бит SPECIAL установлен в 1, то бит GC_OR_START выбирает между выполнением General Call или Start-байта.  0: General Call. После выполнения этой команды, из всех транзакций становится возможна только передача данных по адресу General Call. При попытке чтения с шины произойдет установка бита TX_ABRT (IC_RAW_INTR_STAT[6]). Контроллер будет выполнять передачи по адресу General Call, до тех пор пока не будет сброшен бит SPECIAL (IC_TAR[11]). 1: START-байт.
11	SPECIAL	Этот бит контролирует, будет ли происходить передача General Call, либо START-байт, или контроллер будет работать в обычном режиме.  0: Значение бита GC_OR_START игнорируется и контроллер будет в обычном режиме работать по адресу IC_TAR. 1: Будет выполнена специальная команда, в зависимости от значения бита GC_OR_START
12	IC_10BITADDR_MASTER	Этот бит контролирует тип адреса удаленного slave-устройства.  0: 7-и битный адрес. 1: 10-и битная адресация
13:31	-	Резерв

### 13.2.3 Регистр IC\_SAR

Регистр slave-адреса.

Формат регистра IC\_SAR приведен в Таблица 13.4.

**Таблица 13.4. Формат регистра IC\_SAR**

Номер бита	Условное обозначение	Назначение
0:9	IC_SAR	Регистр хранит адрес контроллера, работающего в режиме slave.  Если установлен 7-и битный формат адреса (IC_CON=0), то используются только биты IC_SAR[6:0]  Этот регистр доступен по записи только при выключенном контроллере (IC_ENABLE=0)
10:31	-	Резерв

### 13.2.4 Регистр IC\_HS\_MADDR

Регистр адреса master-код.

Формат регистра IC\_HS\_MADDR приведен в Таблица 13.5.

**Таблица 13.5. Формат регистра IC\_HS\_MADDR**

Номер бита	Условное обозначение	Назначение
0:2	IC_HS_MAR	Эти биты содержат значение master-кода для high-speed режима. Каждое master -устройство имеет уникальный master -код вида (00001xxx). На одной шине может располагаться до 8-и master -устройств. Этот регистр может быть доступен для записи, только когда контроллер выключен (IC_ENABLE=0).
3:31	–	Резерв

### 13.2.5 Регистр IC\_DATA\_CMD

Регистр управления данными.

Через этот регистр происходит запись из CPU в буфер передатчика (TX FIFO) и чтение в CPU из буфера приемника (RX FIFO).

Формат регистра IC\_DATA\_CMD приведен в Таблица 13.6.

**Таблица 13.6. Формат регистра IC\_DATA\_CMD**

Номер бита	Условное обозначение	Назначение
0:7	DAT	Поле данных. Это поле содержит передаваемые или принимаемые данные по шине I <sup>2</sup> C.
8	CMD	Бит направления передачи. Когда порт находится в режиме master, этот бит контролирует направление передачи: 1: Чтение. 0: Запись. Попытка выполнить операция чтения после General Call приведет к прерыванию TX_ABRT. Запись «0» в этот бит после прерывания RD_REQ также приведет к прерыванию TX_ABRT.
9:31	-	Резерв

### 13.2.6 Регистр IC\_SS\_SCL\_HCNT

Старший регистр счетчика предделителя частоты при работе модуля в standard-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра LC\_SS\_SCL\_HCNT приведен в Таблица 13.7.

**Таблица 13.7. Формат регистра IC\_SS\_SCL\_HCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_SS_SCL_HCNT	Регистр устанавливает значение счетчика предделителя для высокого уровня сигнала SCL в standard-speed режиме.  Минимальное значение - 6, максимальное - 65525.
16:31	-	Резерв

### 13.2.7 Регистр IC\_SS\_SCL\_LCNT

Младший регистр счетчика предделителя частоты при работе модуля в standard-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра LC\_SS\_SCL\_LCNT приведен в Таблица 13.8.

**Таблица 13.8. Формат регистра IC\_SS\_SCL\_LCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_SS_SCL_LCNT	Регистр устанавливает значение счетчика предделителя для низкого уровня сигнала SCL в standard-speed режиме.  Минимальное значение - 8.
16:31	-	Резерв

### 13.2.8 Регистр IC\_FS\_SCL\_HCNT

Старший регистр счетчика предделителя частоты при работе модуля в Fast-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра LC\_FS\_SCL\_HCNT приведен в Таблица 13.9.

**Таблица 13.9. Формат регистра IC\_FS\_SCL\_HCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_FS_SCL_HCNT	Регистр устанавливает значение счетчика предделителя для высокого уровня сигнала SCL в fast-speed режиме.  Минимальное значение - 6.
16:31	-	Резерв

### 13.2.9 Регистр IC\_FS\_SCL\_LCNT

Младший регистр счетчика предделителя частоты при работе модуля в Fast-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра LC\_FS\_SCL\_LCNT приведен в Таблица 13.10.

**Таблица 13.10. Формат регистра IC\_FS\_SCL\_LCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_FS_SCL_LCNT	Регистр устанавливает значение счетчика предделителя для низкого уровня сигнала SCL в fast-speed режиме.  Минимальное значение - 8.
16:31	-	Резерв

### 13.2.10 Регистр IC\_HS\_SCL\_HCNT

Старший регистр счетчика предделителя частоты при работе модуля в High-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра IC\_HS\_SCL\_HCNT приведен в Таблица 13.11.

**Таблица 13.11. Формат регистра IC\_HS\_SCL\_HCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_FS_SCL_HCNT	Регистр устанавливает значение счетчика предделителя для высокого уровня сигнала SCL в high-speed режиме.  Минимальное значение - 6.
16:31	-	Резерв

### 13.2.11 Регистр IC\_HS\_SCL\_LCNT

Младший регистр счетчика предделителя частоты при работе модуля в high-speed режиме.

Регистр доступен для записи только при выключенном контроллере (IC\_ENABLE=0).

Формат регистра IC\_HS\_SCL\_LCNT приведен в Таблица 13.12.

**Таблица 13.12. Формат регистра IC\_HS\_SCL\_LCNT**

Номер бита	Условное обозначение	Назначение
0:15	IC_HS_SCL_LCNT	Регистр устанавливает значение счетчика предделителя для низкого уровня сигнала SCL в high-speed режиме.  Минимальное значение - 8.
16:31	-	Резерв

### 13.2.12 Регистр IC\_INTR\_STAT

Регистр статуса прерываний.

Каждый бит этого регистра имеет соответствующий маскирующий бит в регистре IC\_INTR\_MASK. Значения битов прерывания до маскирования доступны в регистре IC\_RAW\_INTR\_STAT.

Формат регистра IC\_INTR\_STAT приведен в Таблица 13.13.

**Таблица 13.13. Формат регистра IC\_INTR\_STAT**

Номер бита	Условное обозначение	Назначение
0	R_RX_UNDER	См. Регистр IC_RAW_INTR_STAT.
1	R_RX_OVER	
2	R_RX_FULL	
3	R_TX_OVER	
4	R_TX_EMPTY	
5	R_RD_REQ	
6	R_TX_ABRT	
7	R_RX_DONE	
8	R_ACTIVITY	
9	R_STOP_DET	
10	R_START_DET	
11	R_GEN_CALL	
12:31	-	Резерв

### 13.2.13 Регистр IC\_INTR\_MASK

Регистр маскирования прерываний.

Биты этого регистра маскируют соответствующие биты регистра статуса прерываний. Значение 0 - маскирует прерывание, значение 1 - не маскирует.

Формат регистра IC\_INTR\_MASK приведен в Таблица 13.14.

**Таблица 13.14. Формат регистра IC\_INTR\_MASK**

Номер бита	Условное обозначение	Назначение
0	R_RX_UNDER	Эти биты маскируют прерывания, соответствующие битам регистра IC_INTR_STAT.
1	R_RX_OVER	
2	R_RX_FULL	
3	R_TX_OVER	
4	R_TX_EMPTY	
5	R_RD_REQ	
6	R_TX_ABRT	
7	R_RX_DONE	
8	R_ACTIVITY	
9	R_STOP_DET	
10	R_START_DET	
11	R_GEN_CALL	
12:31	-	Резерв

### 13.2.14 Регистр IC\_RAW\_INTR\_STAT

Регистр статуса немаскированных прерываний.

При выключении контроллера, биты регистра остаются активными до тех, пока контроллер не перейдет в состояние IDLE.

Формат регистра IC\_RAW\_CLR\_INTR\_STAT приведен в Таблица 13.15.

**Таблица 13.15. Формат регистра IC\_RAW\_INTR\_STAT**

Номер бита	Условное обозначение	Назначение
0	RX_UNDER	Бит устанавливается в 1, когда пользователь пытается читать данные из пустого буфера приемника.
1	RX_OVER	Бит устанавливается в 1, если буфер приемника заполнен, но по шине приходит еще один байт данных. Контроллер выдаст сигнал подтверждения на шину, но все данные, полученные после заполнения FIFO приемника, будут потеряны.
2	RX_FULL	Бит устанавливается в 1, когда буфер заполняется до уровня, обозначенного в регистре IC_RX_TL. Он автоматически сбрасывается, как только кол-во данных в буфере становится меньше этого уровня.
3	TX_OVER	Бит устанавливается в 1, во время передачи, если процессор пытается записать данные в IC_DATA_CMD при заполненном FIFO передатчика.
4	TX_EMPTY	Бит устанавливается в 1, когда кол-во данных в FIFO передатчика снижается до уровня, обозначенного в IC_TX_TL.
5	RD_REQ	Бит устанавливается в 1, когда модуль работает в режиме slave, а текущий master шины запрашивает чтение данных из модуля. Контроллер удерживает состояние ожидания (SCL=0), пока прерывание обрабатывается. Процессор должен ответить на это прерывание и начать выдавать запрашиваемые данные в IC_DATA_CMD регистр. Этот бит установится в 0 после чтения регистра IC_CLR_RD_REQ.
6	TX_ABRT	Этот бит устанавливается в 1, если модуль работает в режиме передатчика и не может произвести передачу. Когда этот бит устанавливается в 1, регистр IC_TX_ABRT_SOURCE отображает причину обрыва передачи.
7	RX_DONE	Когда контроллер работает в режиме slave-передатчика, этот бит устанавливается в 1, если master не подтверждает передачу байта. Это происходит, когда slave передал все данные.
8	ACTIVITY	Установка этого бита в 1 говорит о том, что контроллер проявил какую-либо активность. Сброс прерывания возможен одним из четырех способов: <ul style="list-style-type: none"> <li>• Выключение контроллера I<sup>2</sup>C</li> <li>• Чтение регистра IC_CLR_ACTIVITY</li> <li>• Чтение регистра IC_CLR_INTR</li> <li>• Системный сброс.</li> </ul>
9	STOP_DET	Этот бит устанавливается в 1, если на шине возникает состояние STOP, независимо от того работает ли контроллер в режиме slave или master.
10	START_DET	Этот бит устанавливается в 1, если на шине возникает START или RESTART состояние, независимо от того работает ли контроллер в режиме slave или master.
11	GEN_CALL	Этот бит устанавливается в 1, если получен General Call адрес и отправлено подтверждение. Сбрасывается в выключении контроллера или чтением регистра IC_CLR_GEN_CALL.
12:31	-	Резерв

### 13.2.15 Регистр IC\_RX\_TL

Регистр порога заполнения FIFO приемника.

Формат регистра IC\_RX\_TL приведен в Таблица 13.16.

**Таблица 13.16. Формат регистра IC\_RX\_TL**

Номер бита	Условное обозначение	Назначение
0:7	RX_TL	Содержит пороговое значение заполнения FIFO приемника. Контролирует число записей в буфере, по достижению которого, формируется прерывание RX_FULL (2-й бит регистра IC_RAW_INTR_STAT). Значение изменяется от 0 до 255.
8:31	-	Резерв

### 13.2.16 Регистр IC\_TX\_TL

Регистр порога заполнения FIFO передатчика.

Формат регистра IC\_TX\_TL приведен в Таблица 13.17.

**Таблица 13.17. Формат регистра IC\_TX\_TL**

Номер бита	Условное обозначение	Назначение
0:7	TX_TL	Содержит пороговое значение заполнения FIFO приемника. Контролирует число записей в буфере, по достижению которого, происходит прерывание TX_EMPTY (4-й бит регистра IC_RAW_INTR_STAT). Значение изменяется от 0 до 255.
8:31	-	Резерв

### 13.2.17 Регистр IC\_CLR\_INTR

Регистр сброса комбинированного и индивидуальных прерываний.

Формат регистра IC\_CLR\_INTR приведен в Таблица 13.18.

**Таблица 13.18. Формат регистра IC\_CLR\_INTR**

Номер бита	Условное обозначение	Назначение
0	CLR_INTR	Чтение этого регистра сбрасывает комбинированное прерывание, все индивидуальные прерывания, регистр IC_TX_ABORT_SOURCE.
1:31	-	Резерв

### 13.2.18 Регистр IC\_CLR\_RX\_UNDER

Регистр сброса прерывания RX\_UNDER.

Формат регистра IC\_CLR\_RX\_UNDER приведен в Таблица 13.19.



**Таблица 13.19. Формат регистра IC\_CLR\_RX\_UNDER**

Номер бита	Условное обозначение	Назначение
0	CLR_RX_UNDER	Чтение этого регистра сбрасывает прерывание RX_UNDER (бит 0 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.19 Регистр IC\_CLR\_RX\_OVER

Регистр сброса прерывания RX\_OVER.

Формат регистра IC\_CLR\_RX\_OVER приведен в Таблица 13.20.

**Таблица 13.20. Формат регистра IC\_CLR\_RX\_OVER**

Номер бита	Условное обозначение	Назначение
0	CLR_RX_OVER	Чтение этого регистра сбрасывает прерывание RX_OVER (бит 1 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.20 Регистр IC\_CLR\_TX\_OVER

Регистр сброса прерывания TX\_OVER.

Формат регистра TX\_OVER приведен в Таблица 13.21.

**Таблица 13.21. Формат регистра IC\_CLR\_TX\_OVER**

Номер бита	Условное обозначение	Назначение
0	CLR_TX_OVER	Чтение этого регистра сбрасывает прерывание TX_OVER (бит 3 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.21 Регистр IC\_CLR\_RD\_REQ

Регистр сброса прерывания RD\_REQ.

Формат регистра RD\_REQ приведен в Таблица 13.22.

**Таблица 13.22. Формат регистра IC\_CLR\_RD\_REQ**

Номер бита	Условное обозначение	Назначение
0	CLR_RD_REQ	Чтение этого регистра сбрасывает прерывание RX_UNDER (бит 5 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.22 Регистр IC\_CLR\_TX\_ABRT

Регистр сброса прерывания TX\_ABRT.

Формат регистра IC\_CLR\_TX\_ABRT приведен в Таблица 13.23.

**Таблица 13.23. Формат регистра IC\_CLR\_TX\_ABRT**

Номер бита	Условное обозначение	Назначение
0	CLR_RX_UNDER	Чтение этого регистра сбрасывает прерывание TX_ABRT (бит 6 регистра IC_RAW_INTR_STAT) и регистр IC_TX_ABRT_SOURCE.
1:31	-	Резерв

### 13.2.23 Регистр IC\_CLR\_RX\_DONE

Регистр сброса прерывания RX\_DONE.

Формат регистра IC\_CLR\_RX\_DONE приведен в Таблица 13.24.

**Таблица 13.24. Формат регистра IC\_CLR\_RX\_DONE**

Номер бита	Условное обозначение	Назначение
0	CLR_RX_DONE	Чтение этого регистра сбрасывает прерывание RX_DONE (бит 7 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.24 Регистр IC\_CLR\_ACTIVITY

Регистр сброса прерывания ACTIVITY.

Формат регистра IC\_CLR\_ACTIVITY приведен в Таблица 13.25.

**Таблица 13.25. Формат регистра IC\_CLR\_ACTIVITY**

Номер бита	Условное обозначение	Назначение
0	CLR_ACTIVITY	Чтение этого регистра сбрасывает прерывание ACTIVITY (бит 8 регистра IC_RAW_INTR_STAT), если контроллер I <sup>2</sup> C не активен.
1:31	-	Резерв

### 13.2.25 Регистр IC\_CLR\_STOP\_DET

Регистр сброса прерывания STOP\_DET.

Формат регистра IC\_STOP\_DET приведен в Таблица 13.26.

**Таблица 13.26. Формат регистра IC\_STOP\_DET**

Номер бита	Условное обозначение	Назначение
0	CLR_STOP_DET	Чтение этого регистра сбрасывает прерывание STOP_DET (бит 9 регистра IC_RAW_INTR_STAT)
1:15	-	Резерв

### 13.2.26 Регистр IC\_CLR\_START\_DET

Регистр сброса прерывания START\_DET.

Формат регистра IC\_CLR\_START\_DET приведен в Таблица 13.27.

**Таблица 13.27. Формат регистра IC\_CLR\_START\_DET**

Номер бита	Условное обозначение	Назначение
0	CLR_START_DET	Чтение этого регистра сбрасывает прерывание START_DET (бит 10 регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.27 Регистр IC\_CLR\_GEN\_CALL

Регистр сброса прерывания GEN\_CALL.

Формат регистра IC\_CLR\_GEN\_CALL приведен в Таблица 13.28.

**Таблица 13.28. Формат регистра IC\_CLR\_GEN\_CALL**

Номер бита	Условное обозначение	Назначение
0	CLR_GEN_CALL	Чтение этого регистра сбрасывает прерывание GEN_CALL (11 бит регистра IC_RAW_INTR_STAT)
1:31	-	Резерв

### 13.2.28 Регистр IC\_ENABLE

Регистр включения контроллера I<sup>2</sup>C.

Формат регистра IC\_ENABLE приведен в Таблица 13.29.

**Таблица 13.29. Формат регистра IC\_ENABLE**

Номер бита	Условное обозначение	Назначение
0	ENABLE	<p>0: Контроллер выключен 1: Контроллер включен.</p> <p>При выключении контроллера, происходит следующее:</p> <ul style="list-style-type: none"> <li>• Стирается содержимое FIFO приемника и FIFO передатчика.</li> <li>• Статус-биты регистра IC_INTR_STAT остаются активными до тех пока контроллер не перейдет в состояние IDLE.</li> </ul> <p>Если контроллер функционировал как передатчик, он завершит текущую передачу и остановится из-за того, что буфер передатчика станет пуст. Если контроллер функционировал как приемник, то завершив текущую передачу, контроллер не подтвердит завершение передачи битом acknowledge.</p>
1:31	-	Резерв

### 13.2.29 Регистр IC\_STATUS

Регистр статуса контроллера

Регистр доступен только по чтению, отображает статус текущей передачи и статус FIFO. Регистр статуса может быть прочтен в любое время. Когда происходит выключение контроллера записью 0 в 0-ой бит регистра IC\_ENABLE, происходит следующее:

- биты 1 и 2 устанавливаются в 1;
- биты 3 и 4 устанавливаются в 0;

Когда контроллер переходит в состояние IDLE:

- биты 5 и 6 устанавливаются в 0

Формат регистра IC\_STATUS приведен в Таблица 13.30.

**Таблица 13.30. Формат регистра IC\_STATUS**

Номер бита	Условное обозначение	Назначение
0	ACTIVITY	Статус активности шины  ACTIVITY = MST_ACTIVITY OR SLV_ACTIVITY
1	TFNF	FIFO передатчика не полон.  0 - Буфер передатчика полон 1 - Буфер передатчика не полон
2	TFE	FIFO передатчика абсолютно пуст.  0 - Буфер передатчика не пуст 1 - Буфер передатчика пуст
3	RFNE	FIFO приемника не пуст.  0 - Буфер приемника пуст 1 - Буфер приемника не пуст
4	RFF	Буфер приемника абсолютно полон.  0 - Буфер приемника не полон 1 - Буфер приемника полон
5	MST_ACTIVITY	Статус активности состояния master.  0: Состояние master контроллера не активно 1: Состояние master контроллера активно
6	SLV_ACTIVITY	Статус активности slave состояния.  0: Состояние slave контроллера не активно 1: Состояние slave контроллера активно
7:31	-	Резерв

### 13.2.30 Регистр IC\_TXFLR

Регистр уровня FIFO передатчика.

Этот регистр содержит число строк данных в FIFO передатчика. Регистр сбрасывается в следующих случаях:

- происходит выключение контроллера;
- происходит обрыв передачи. Устанавливается бит TX\_ABRT регистра IC\_RAW\_INTR\_STAT.

Формат регистра IC\_TXFLR приведен в Таблица 13.31.

**Таблица 13.31. Формат регистра IC\_TXFLR**

Номер бита	Условное обозначение	Назначение
0:5	TXFLTR	Уровень FIFO передатчика. Содержит число доступных записей данных в FIFO передатчика.
6:31	-	Резерв

### 13.2.31 Регистр IC\_RXFLR

Регистр уровня FIFO приемника.

Этот регистр содержит число строк данных в FIFO приемника. Регистр сбрасывается в следующих случаях:

- происходит выключение контроллера;
- происходит обрыв передачи по любой причине, описанной в регистре IC\_TX\_ABRT\_SOURCE.

Формат регистра IC\_RXFLR приведен в Таблица 13.32.

**Таблица 13.32. Формат регистра IC\_RXFLR**

Номер бита	Условное обозначение	Назначение
0:5	RXFLR	Уровень FIFO приемника. Содержит число доступных записей данных в FIFO приемника.
6:31	-	Резерв.

### 13.2.32 Регистр IC\_SDA\_HOLD

Регистр времени удержания сигнала SDA.

Этот регистр контролирует время удержания (выраженное в числе тактов частоты L3\_PCLK) сигнала SDA после заднего фронта сигнала SCL в обоих slave и master режимах работы контроллера. Значение этого регистра должно быть выше одного такта а для режима master и 7-и тактов для режима slave. Регистр доступен по записи только при IC\_ENABLED[0]=0.

Причем  $IC\_SDA\_HOLD \leq N\_SCL\_LOW - 2$ , где N\_SCL\_LOW это продолжительность низкого уровня сигнала SCL в тактах L3\_PCLK.

Формат регистра IC\_SDA\_HOLD приведен в Таблица 13.33.

**Таблица 13.33. Формат регистра IC\_SDA\_HOLD**

Номер бита	Условное обозначение	Назначение
0:15	IC_SDA_HOLD	Устанавливает требуемое значение времени удержания сигнала SDA в тактах L3 PCLK
16:31	-	Резерв

### 13.2.33 Регистр IC\_TX\_ABRT\_SOURCE

Регистр причины обрыва передачи.

Этот регистр имеет 16 бит, которые отображают причину установки в «1» бита TX\_ABRT. Регистр сбрасывается чтением регистра IC\_CLR\_TX\_ABRT или IC\_CLR\_INTR (кроме бита 9 - ABRT\_SBYTE\_NORSTRТ). Для сброса девятого бита, сначала должны быть выполнены следующие условия.

- должна быть включена возможность отправки последовательности RESTART, IC\_CON[5]=1;
- должен быть сброшен бит SPECAL (IC\_TAR[11]) или бит GC\_OR\_START (IC\_TAR[10]).

После выполнения этих условий бит ABRT\_SBYTE\_NORSTRТ сбрасывается также как и остальные.

Формат регистра IC\_TX\_ABRT\_SOURCE приведен в Таблица 13.34.

**Таблица 13.34. Формат регистра IC\_TX\_ABRT\_SOURCE**

Номер бита	Условное обозначение	Назначение	Источник
0	ABRT_7B_ADDR_NOACK	1: Контроллер в режиме master с 7-и битной адресацией не получил сигнала подтверждения от slave после отправления адреса.	Master - передатчик или master - приемник.
1	ABRT_10ADDR1_NOACK	1: Контроллер в режиме master с 10-и битной адресацией не получил сигнала подтверждения от slave после отправления первого 10-и битного адреса	Master - передатчик или master - приемник.
2	ABRT_10ADDR2_NOACK	1: Контроллер в режиме master с 10-и битной адресацией не получил сигнала подтверждения от slave после отправления второго 10-и битного адреса	Master - передатчик или master - приемник.
3	ABRT_TXDATA_NOACK	1: Контроллер в режиме мастера получил сигнал подтверждения от slave после отправления адреса. Но после отправления байта(-ов) данных по этому адресу, он не получил сигнала подтверждения от slave -устройства.	Master - передатчик
4	ABRT_GCALL_NOACK	1: Контроллер в режиме мастера отправил General Call, и ни один slave на шине не отправил подтверждающего сигнала.	Master - передатчик

Номер бита	Условное обозначение	Назначение	Источник
5	ABRT_GCALL_READ	1: Контроллер в режиме мастера отправил General Call, но пользователь объявил запрос на чтение (IC_DATA_CMD[8]=1)	Master - передатчик
6	ABRT_HS_ACKDET	1: Контроллер в режиме master и скоростном режиме high-speed получил подтверждение на high-speed master code.	Master
7	ABRT_SBYTE_ACKDET	1: Мастер отправил START-байт и получил сигнал подтверждения	Master
8	ABRT_HS_NORSTR	1: Отключена возможность отправлять последовательности RESTART (IC_RESTART_EN(IC_CON[5]) = 0) и пользователь пытается в режиме high-speed master осуществить передачу данных.	Master - передатчик или master - приемник.
9	ABRT_SBYTE_NORSTR	1: Отключена возможность отправлять последовательности RESTART (IC_RESTART_EN(IC_CON[5]) = 0) и пользователь пытается отправить START-байт.	Master
10	ABRT_10B_RD_NORSTR	1: Отключена возможность отправлять последовательности RESTART (IC_RESTART_EN(IC_CON[5]) = 0) и master отправляет запрос на чтение в режиме 10-и битной адресации	Master - приемник
11	ABRT_MASTER_DIS	1: Пользователь пытается инициализировать master обмен при выключенном master -режиме.	Master - передатчик или master - приемник.
12	ARB_LOST	1: Master проиграл арбитраж, или (если IC_TX_ABRT_SOURCE[14] также установлен) slave -передатчик проигрывает арбитраж. (Контроллер может находиться в режиме master и в режиме slave одновременно)	Master - передатчик или master - приемник
13	ABRT_SLVFLUSH_TXFIFO	1: Slave получил команду на чтение и некоторые данные находятся в FIFO передатчика, поэтому slave запрашивает TX_ABRT прерывание для стирания старых данных из FIFO передатчика.	Slave-передатчик
14	ABRT_SLV_ARBLOST	1: Slave теряет шину во время передачи данных. В это же время устанавливается IC_TX_ABRT_SOURCE[12].	Slave - передатчик
15	ABRT_SLVRD_INTX	1: Процессор отвечает запросу от slave на передачу данных удаленному master -устройству, а пользователь пишет 1 в CMD (IC_DATA_CMD[8])	Slave - передатчик
16:31	-	Резерв	

### 13.2.34 Регистр IC\_SLV\_DATA\_NACK\_ONLY

Регистр генерации сигнала отсутствия подтверждения - NACK.

Сигнал генерируется контроллером, работающем в режиме Slave-приемника. Запись в этот регистр может осуществляться при следующих условиях:

- контроллер выключен. IC\_ENABLE[0]=0;
- Slave-режим не активен. IC\_STATUS[6]=0.

Формат регистра IC\_SLV\_DATA\_NACK\_ONLY приведен в Таблица 13.35.

**Таблица 13.35. Формат регистра IC\_SLV\_DATA\_NACK\_ONLY**

Номер бита	Условное обозначение	Назначение
0	NACK	1: Генерировать NACK после получения байта данных. 0: Генерировать NACK/ACK в зависимости от нормальных критериев.

### 13.2.35 Регистр IC\_DMA\_CR

Регистр управления аппаратным интерфейсом запросов к PDMA.

Этот регистр используется для включения интерфейса взаимодействия с контроллером DMA. Регистр имеет отдельные биты на прием и на передачу. Запись в этот регистр возможна при включенном контроллере.

Формат регистра IC\_DMA\_CR приведен в Таблица 13.36.

**Таблица 13.36. Формат регистра IC\_DMA\_CR**

Номер бита	Условное обозначение	Назначение
0	RDMAE	0: Выключение канала FIFO DMA приемника. 1: Выключение канала FIFO DMA приемника.
1	TDMAE	0: Выключение канала FIFO DMA передатчика. 1: Выключение канала FIFO DMA передатчика.
2:31	-	Резерв

### 13.2.36 Регистр IC\_DMA\_TDLR

Регистр уровня данных передатчика DMA.

Формат регистра IC\_DMA\_TDLR приведен в Таблица 13.37.

**Таблица 13.37. Формат регистра IC\_DMA\_TDLR**

Номер бита	Условное обозначение	Назначение
0:4	DMATDL	Контролирует уровень данных в буфере передатчика по достижении которого будет возникать запрос к PDMA на запись данных в FIFO передатчика.
5:31	-	Резерв

### 13.2.37 Регистр IC\_DMA\_RDLR

Регистр уровня данных приемника DMA.

Формат регистра IC\_DMA\_RDLR приведен в Таблица 13.38.



**Таблица 13.38. Формат регистра IC\_DMA\_RDLR**

Номер бита	Условное обозначение	Назначение
0:4	DMARDL	Контролирует уровень данных в буфере приемника по достижении которого будет возникать запрос к PDMA на чтение данных из FIFO приемника.
5:31	-	Резерв

### 13.2.38 Регистр IC\_SDA\_SETUP

Регистр продолжительности SDA.

Этот регистр контролирует продолжительность временной задержки (в тактах частоты L3\_PCLK) введенной между передним фронтом сигнала SCL относительно изменения SDA

Формат регистра IC\_SDA\_SETUP приведен в Таблица 13.39.

**Таблица 13.39. Формат регистра IC\_SDA\_SETUP**

Номер бита	Условное обозначение	Назначение
0:7	SDA_SETUP	

### 13.2.39 Регистр IC\_ACK\_GENERAL\_CALL

Регистр подтверждения General Call

Этот регистр контролирует, каким сигналом ACK или NACK, будет отвечать контроллер на General Call адрес.

Формат регистра IC\_ACK\_GENERAL\_CALL приведен в Таблица 13.40.

**Таблица 13.40. Формат регистра IC\_ACK\_GENERAL\_CALL**

Номер бита	Условное обозначение	Назначение
0	ACK_GEN_CALL	0: NACK 1: ACK
1:31	-	Резерв

### 13.2.40 Регистр IC\_ENABLE\_STATUS

Этот регистр используется для получения информации о статусе контроллера, когда регистр IC\_ENABLE переключается из 1 в 0, т.е. когда происходит выключение контроллера.

Если IC\_ENABLE установлен в 1, то биты 1:2 преобразуются в 0, а бит 0 в 1.

Если IC\_ENABLE установлен в 0, то биты 1:2 будут действительны тогда, когда бит 0 будет читаться как 0.

Формат регистра IC\_ENABLE\_STATUS приведен в Таблица 13.41.

**Таблица 13.41. Формат регистра IC\_ENABLE\_STATUS**

Номер бита	Условное обозначение	Назначение
0	IC_EN	<p>0: Контроллер выключен 1: Контроллер включен</p> <p>CPU может безопасно читать этот бит в любое время. Когда этот бит читается как 0, CPU может безопасно читать SLV_RX_DATA_LOST (бит 2) и SLV_DISABLED_WHILE_BUSY (бит 1).</p>
1	SLV_DISABLED_WHILE_BUSY	<p>Slave выключен когда занят (передача, прием)</p> <p>Этот бит отображает обрыв текущей операции контроллера, работающего в режиме slave из-за выключения контроллера (изменение состояния регистра IC_ENABLE из 1 в 0). Этот бит устанавливается, если CPU пишет 0 в регистр IC_ENABLE, когда:</p> <ul style="list-style-type: none"> <li>Контроллер принимал байт адреса как slave-передатчик от удаленного master-устройства.</li> <li>Контроллер принимал байты адреса и данных как slave-приемник от удаленного master-устройства.</li> </ul> <p>Контроллер выдает сигнал NACK по завершению текущей фазы обмена в любом случае.</p> <p>CPU может безопасно читать этот бит, когда IC_EN (бит 0) читается как 0.</p>
2	SLV_RX_DATA_LOS	<p>Slave потерял принимаемые данные.</p> <p>Этот бит устанавливается, если прерывается операция slave-приемника с потерей байта данных из-за выключения контроллера (изменения состояния регистра IC_ENABLE из 1 в 0) в момент приема байта данных. Когда это происходит, контроллер выдает сигнал NACK по завершению приема.</p> <p>CPU может безопасно читать этот бит, когда IC_EN (бит 0) читается как 0.</p>
3:31	-	Резерв

### 13.2.41 Регистр IC\_HS\_SPKLEN

Регистр настройки параметров фильтрации помех для high-speed режима.

Этот регистр используется для хранения продолжительности (измеренной в кол-ве тактов PCLK) наиболее длинной фильтруемой помехи.

Формат регистра IC\_HS\_SPKLEN приведен в Таблица 13.42.

**Таблица 13.42. Формат регистра IC\_HS\_SPKLEN**

Номер бита	Условное обозначение	Назначение
0:7	IC_HS_SPKLEN	Регистр доступен для записи только при выключенном контроллере (IC_ENABLE=0). Регистр устанавливает продолжительность наиболее длинной фильтруемой помехи в линиях SDA и SCL (измеренной в кол-ве тактов L3_PCLK).
8:31	-	Резерв

## 14. I2S: контроллер интерфейса I2S

### 14.3 Основные особенности

- 1 стереоканал передачи данных
- Режим работы - Master
- Разрешение аудио сигнала до 32 бит
- Глубина буферов FIFO приемника/передатчика 8 слов
- Программируемый порог заполнения буферов FIFO на прерывание
- Опорная частота доступная с вывода микросхемы (PA15 – I2S\_EXTCLK) или с PLL (SYSCLK)

### 14.4 Конфигурация

Конфигурационные параметры контроллера доступны по ссылке - TBD

### 14.5 Регистры

В Таблица 14.1 приведен перечень программно-доступных регистров контроллера I<sup>2</sup>S.

**Таблица 14.1. Перечень программно-доступных регистров контроллера I<sup>2</sup>S**

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
IER	Регистр включения контроллера.	W/R	0x0	0x0
IRER	Регистр включения блока приема.	W/R	0x0	0x4
ITER	Регистр включения блока передачи.	W/R	0x0	0x8
CER	Регистр включения тактовой частоты.	W/R	0x0	0xC
CCR	Регистр конфигурации тактовой частоты.	W/R	0x10	0x10
RXFFR	Регистр сброса FIFO буферов блока приема.	W	0x0	0x14
TXFFR	Регистр сброса FIFO буферов блока передачи.	W	0x0	0x18
LRBR0	Регистр «левого» слова приемника.	R	0x0	0x20
LTHR0	Регистр «левого» слова нулевого передатчика.	W	0x0	0x20
RRBR0	Регистр «правого» слова приемника.	R	0x0	0x24
RTHR0	Регистр «правого» слова нулевого передатчика.	W	0x0	0x24
RER0	Регистр включения приемника.	W/R	0x1	0x28

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
TER0	Регистр включения нулевого передатчика.	W/R	0x1	0x2C
RCR0	Регистр конфигурации приемника.	W/R	0x5	0x30
TCR0	Регистр конфигурации нулевого передатчика.	W/R	0x5	0x34
ISR0	Регистр статуса прерывания нулевого канала.	R	0x10	0x38
IMR0	Регистр маски прерывания нулевого канала.	W/R	0x33	0x3C
ROR0	Регистр переполнения FIFO приемника.	R	0x0	0x40
TOR0	Регистр переполнения FIFO нулевого передатчика.	R	0x0	0x44
RFCR0	Регистр конфигурации FIFO приемника.	W/R	0x3	0x48
TFCR0	Регистр конфигурации FIFO нулевого передатчика.	W/R	0x3	0x4C
RFF0	Регистр сброса FIFO приемника.	W	0x0	0x50
TFF0	Регистр сброса FIFO нулевого передатчика.	W	0x0	0x54
LTHR1	Регистр «левого» слова первого передатчика.	W	0x0	0x60
RTHR1	Регистр «правого» слова первого передатчика.	W	0x0	0x64
TER1	Регистр включения первого передатчика	W/R	0x1	0x6C
TCR1	Регистр конфигурации первого передатчика.	W/R	0x5	0x74
ISR1	Регистр статуса прерывания первого канала.	R	0x10	0x78
IMR1	Регистр маски прерывания первого канала.	W/R	0x33	0x7C
TOR1	Регистр переполнения первого передатчика.	R	0x0	0x84
TFCR1	Регистр конфигурации FIFO первого передатчика.	W/R	0x3	0x8C
TFF1	Регистр сброса FIFO первого передатчика.	W	0x0	0x94
LTHR2	Регистр «левого» слова второго передатчика.	W	0x0	0xA0
RTHR2	Регистр «правого» слова второго передатчика.	W	0x0	0xA4
TER2	Регистр включения второго передатчика	W/R	0x1	0xAC
TCR2	Регистр конфигурации второго передатчика.	W/R	0x5	0xB4

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Смещение
ISR2	Регистр статуса прерывания второго канала.	R	0x10	0xB8
IMR2	Регистр маски прерывания второго канала.	W/R	0x33	0xBC
TOR2	Регистр переполнения второго передатчика.	R	0x0	0xC4
TFCR2	Регистр конфигурации FIFO второго передатчика.	W/R	0x3	0xCC
TFF2	Регистр сброса FIFO второго передатчика.	W	0x0	0xD4
LTHR3	Регистр «левого» слова третьего передатчика.	W	0x0	0xE0
RTHR3	Регистр «правого» слова третьего передатчика.	W	0x0	0xE4
TER3	Регистр включения третьего передатчика	W/R	0x1	0xEC
TCR3	Регистр конфигурации третьего передатчика.	W/R	0x5	0xF4
ISR3	Регистр статуса прерывания третьего канала.	R	0x10	0xF8
IMR3	Регистр маски прерывания третьего канала.	W/R	0x33	0xFC
TOR3	Регистр переполнения третьего передатчика.	R	0x0	0x104
TFCR3	Регистр конфигурации FIFO третьего передатчика.	W/R	0x3	0x10C
TFF3	Регистр сброса FIFO третьего передатчика.	W	0x0	0x114
RXDMA	Регистр DMA приема.	R	0x0	0x1C0
TXDMA	Регистр DMA передачи.	W	0x0	0x1C8
RTXDMA	Регистр сброса DMA передачи.	W	0x0	0x1CC
I2S_COMP_PARAM_2	Второй регистр параметров компонента.	R	0x48c	0x1F0
I2S_COMP_PARAM_1	Первый регистр параметров компонента.	R	0x24c067e	0x1F4
I2S_COMP_VERSION	Регистр ревизии компонента.	R	0x3130362a	0x1F8
I2S_COMP_TYPE	Регистр типа компонента.	R	0x445701a0	0x1FC

### 14.5.12 Регистр IER

Регистр включения контроллера I<sup>2</sup>S.

Формат регистра IER приведен в Таблица 14.2.

**Таблица 14.2. Формат регистра IER**

Номер бита	Условное обозначение	Назначение
0	IEN	Установка этого бита в 0 приводит к выключению всех каналов обмена и стиранию содержимого всех FIFO.  1: Контроллер I <sup>2</sup> S включен. 0: Контроллер I <sup>2</sup> S выключен.
1:31	-	Резерв

**14.5.13 Регистр IRER**

Регистр включения блока приема контроллера I<sup>2</sup>S.

Формат регистра IRER приведен в Таблица 14.3.

**Таблица 14.3. Формат регистра IRER**

Номер бита	Условное обозначение	Назначение
0	RXEN	1: Блок приема контроллера I <sup>2</sup> S включен. 0: Блок приема контроллера I <sup>2</sup> S выключен.
1:31	-	Резерв

**14.5.14 Регистр ITER**

Регистр включения блока передачи контроллера I<sup>2</sup>S.

Формат регистра ITER приведен в Таблица 14.4.

**Таблица 14.4. Формат регистра ITER**

Номер бита	Условное обозначение	Назначение
0	RXEN	1: Блок передачи контроллера I <sup>2</sup> S включен. 0: Блок передачи контроллера I <sup>2</sup> S выключен.
1:31	-	Резерв

**14.5.15 Регистр CER**

Регистр включения тактовых сигналов SCLK\_EN, SCLK\_GATE, WS\_OUT.

Формат регистра CER приведен в Таблица 14.5.

**Таблица 14.5. Формат регистра CER**

Номер бита	Условное обозначение	Назначение
0	CLKEN	1: сигналы SCLK_EN, SCLK_GATE, WS_OUT генерируются. 0: тактовые сигналы SCLK_EN, SCLK_GATE, WS_OUT не генерируются (идут низким уровнем).
1:31	-	Резерв

**14.5.16 Регистр CCR**

Регистр конфигурации генерации тактовых сигналов.

Формат регистра CCR приведен в Таблица 14.6.

**Таблица 14.6. Формат регистра CCR**

Номер бита	Условное обозначение	Назначение
0:2	SCLKG	<p>Это поле определяет момент перехода сигнала SCLK_GATE из 0 в 1.</p> <p>0: SCLK_GATE=0 всегда.  1: Через 12 тактов SCLK.  2: Через 16 тактов SCLK.  3: Через 20 тактов SCLK.  4: Через 24 тактов SCLK.</p> <p>Перед изменением этого параметра необходимо выключить генерацию тактовых сигналов (CER[0]=0).</p>
3:4	WSS	<p>Это поле определяет продолжительность высокого и низкого уровней сигнала выбора слова WS_OUT:</p> <p>0: 16 тактов SCLK.  1: 24 тактов SCLK.  2: 32 тактов SCLK.</p> <p>Перед изменением этого параметра необходимо выключить генерацию тактовых сигналов (CER[0]=0).</p>
5:31	-	Резерв

### 14.5.17 Регистр RXFFR

Регистр сброса всех FIFO блока приема.

Формат регистра RXFFR приведен в Таблица 14.7.

**Таблица 14.7. Формат регистра RXFFR**

Номер бита	Условное обозначение	Назначение
0	RXFFR	<p>Запись единицы в этот регистр приводит к стиранию данных во всех FIFO блока приема.</p> <p>Перед записью в этот регистр необходимо выключить блок приема (IRER[0]=0).</p>
1:31	-	Резерв

### 14.5.18 Регистр TXFFR

Регистр сброса всех FIFO блока передачи.

Формат регистра TXFFR приведен в Таблица 14.8.

**Таблица 14.8. Формат регистра TXFFR**

Номер бита	Условное обозначение	Назначение
0	TXFFR	<p>Запись единицы в этот регистр приводит к стиранию данных во всех FIFO блока передачи.</p> <p>Перед записью в этот регистр необходимо выключить блок передачи (ITER[0]=0).</p>
1:31	-	Резерв



### 14.5.19 Регистр LRBR0

Регистр чтения «левого» слова из приемника.

Формат регистра LRBR0 приведен в Таблица 14.9.

**Таблица 14.9. Формат регистра LRBR0**

Номер бита	Условное обозначение	Назначение
0:31	LRBR0	Через этот регистр происходит чтение «левого» слова данных из FIFO приемника. Прежде чем повторить эту операцию необходимо прочитать «правое» слово данных из FIFO приемника через регистр RRBR0. Размер принимаемого слова описан в регистре RCR0.

### 14.5.20 Регистр LTHR<sub>x</sub>

Регистр «левого» слова передатчика  $x$ , где  $x = 0,1,2,3$ .

Формат регистра LTHR<sub>x</sub> приведен в Таблица 14.10.

**Таблица 14.10. Формат регистра LTHR<sub>x</sub>**

Номер бита	Условное обозначение	Назначение
0:31	LTHR <sub>x</sub>	Через этот регистр происходит запись «левого» слова данных в FIFO передатчика $x$ . Прежде чем повторить эту операцию необходимо записать «правое» слово данных в FIFO передатчика $x$ через регистр RTHR <sub>x</sub> . Размер передаваемого слова описан в соответствующем регистре TCR <sub>x</sub> .

### 14.5.21 Регистр RRBR0

Регистр чтения «правого» слова из приемника.

Формат регистра RRBR0 приведен в Таблица 14.11.

**Таблица 14.11. Формат регистра RRBR0**

Номер бита	Условное обозначение	Назначение
0:31	RRBR0	Через этот регистр происходит чтение «правого» слова данных из FIFO приемника. Этой операции должно предшествовать чтение «левого» слова данных из FIFO приемника через регистр LRBR0. Размер принимаемого слова описан в регистре RCR0.

### 14.5.22 Регистр RTHR<sub>x</sub>

Регистр «правого» слова передатчика  $x$ . Где  $x=0,1,2,3$ .

Формат регистра RTHR<sub>x</sub> приведен в Таблица 14.12.

**Таблица 14.12. Формат регистра RTHR<sub>x</sub>**

Номер бита	Условное обозначение	Назначение
0:31	RTHR <sub>x</sub>	Через этот регистр происходит запись «правого» слова данных в FIFO передатчика $x$ . Этой операции должна предшествовать запись «левого» слова данных в FIFO передатчика $x$ через регистр LTHR <sub>x</sub> .

		Размер передаваемого слова описан в соответствующем регистре TCRx.
--	--	--

### 14.5.23 Регистр RER0

Регистр включения приемника.

Формат регистра RER0 приведен в Таблица 14.13.

**Таблица 14.13. Формат регистра RER0**

Номер бита	Условное обозначение	Назначение
0	RXCHEN0	Бит включает приемник. При включении, прием данных начинается с «левого» слова. Глобальное выключение контроллера (RER[0]=0) или блока приема (IRER[0]=0) переписывает значение этого бита. 1: Приемник включен. 2: Приемник выключен.
1:31	-	Резерв

### 14.5.24 Регистр TERx

Регистр включения передатчика x. Где x = 0,1,2,3.

Формат регистра TERx приведен в Таблица 14.14.

**Таблица 14.14. Формат регистра TERx**

Номер бита	Условное обозначение	Назначение
0	TXCHENx	Бит включает передатчик x. При включении, передача данных начинается со следующего «левого» слова. Глобальное выключение контроллера (TER[0]=0) или блока передачи (ITER[0]=0) переписывает значение этого бита. 1: Передатчик x включен. 2: Передатчик x выключен.
1:31	-	Резерв

### 14.5.25 Регистр RCR0

Регистр конфигурации приемника.

Формат регистра RCR0 приведен в Таблица 14.15.

**Таблица 14.15. Формат регистра RCR0**

Номер бита	Условное обозначение	Назначение
0:2	WLEN	Это поле используется для выбора длины принимаемого слова. Принимаемое слово всегда поступает в младшие биты регистра LRBR0 (или RRBR0). При этом, если реальная длина принимаемого слова больше установленного значения, то будут прочитаны только его старшие биты.

Номер бита	Условное обозначение	Назначение
		<p>Пример:</p> <p>Допустим, что установленная длина принимаемого слова – 16 бит. А по линии SDI поступает 32-х разрядное слово 0x12345678. В этом случае, через регистр LRBR0 (или RRBR0) будет прочитано слово 0x00001234.</p> <p>000: зарезервировано.  001: 12 бит.  010: 16 бит.  011: 20 бит.  100: 24 бита.  101: 32 бита.</p> <p>Приемник должен быть выключен перед любыми изменениями этого значения. (RER0 = 0).</p>
1:31	-	Резерв

### 14.5.26 Регистр TCRx

Регистр конфигурации передатчика x, где x = 0,1,2,3.

Формат регистра TCRx приведен в Таблица 14.16.

**Таблица 14.16. Формат регистра TCRx**

Номер бита	Условное обозначение	Назначение
0:2	WLEN	<p>Длина передаваемого слова. Для передачи всегда используются младшие биты 32-х разрядного регистра LTHR<sub>x</sub> (или RTHR<sub>x</sub>).</p> <p>000: зарезервировано  001: 12 бит.  010: 16 бит.  011: 20 бит.  100: 24 бита.  101: 32 бита.</p> <p>Передатчик x должен быть выключен перед любыми изменениями этого значения. (TER<sub>x</sub> = 0).</p>
1:31	-	Резерв

### 14.5.27 Регистр ISR<sub>x</sub>

Регистр статуса прерываний канала x, где x = 0,1,2,3.

Формат регистра ISR<sub>x</sub> приведен в Таблица 14.17.

**Таблица 14.17. Формат регистра ISR<sub>x</sub>**

Номер бита	Условное обозначение	Назначение
0	RXDA	<p>При x = 0, этот бит содержит статус прерывания по наличию данных в приемнике.</p> <p>1: Превышен порог заполнения FIFO приемника  0: Не превышен порог заполнения FIFO приемника</p>

Номер бита	Условное обозначение	Назначение
1	RXFO	При $x = 0$ , этот бит содержит статус прерывания по переполнению приемника.  0: Не произошло потери данных 1: Произошла потеря данных.  В случае поступления данных по каналу SDIO, при заполненном буфере приемника, произойдет потеря этих данных. Данные в FIFO приемника не будут перезаписаны.
2:3	-	Резерв
4	TXFE	Статус прерывания по пустому FIFO передатчика $x$ .  1: Число записей в FIFO передатчика меньше порога заполнения. 0: Число записей в FIFO передатчика не меньше порога заполнения.
5	TXFO	Статус прерывания по переполнению передатчика $x$ .  0: Не произошло потери данных 1: Произошла потеря данных  В случае поступления данных в FIFO передатчика $x$ , при заполненном буфере передатчика, произойдет потеря этих данных. Данные в FIFO не будут перезаписаны
6:31	-	Резерв

### 14.5.28 Регистр IMRx

Регистр маскирования прерываний в канале  $x$ , где  $x = 0, 1, 2, 3$ .

Формат регистра IMRx приведен в Таблица 14.18.

**Таблица 14.18. Формат регистра IMRx**

Номер бита	Условное обозначение	Назначение
0	RXDAM	При $x = 0$ , этот бит содержит маску прерывания по наличию данных в приемнике. 1: Прерывание маскировано 0: Прерывание не маскировано
1	RXFOM	При $x = 0$ , этот бит содержит маску прерывания по переполнению приемника. 1: Прерывание маскировано 0: Прерывание не маскировано
2:3	-	Резерв
4	TXFEM	Маска прерывания по пустому FIFO передатчика $x$ . 1: Прерывание маскировано 0: Прерывание не маскировано
5	TXFOM	Маска прерывания по переполнению передатчика $x$ . 1: Прерывание маскировано 0: Прерывание не маскировано
6:31	-	Резерв

### 14.5.29 Регистр ROR0

Регистр переполнения FIFO приемника.

Формат регистра ROR0 приведен в Таблица 14.19.

**Таблица 14.19. Формат регистра ROR0**

Номер бита	Условное обозначение	Назначение
0	RXCHO	Чтение этого бита сбрасывает прерывание по переполнению приемника.  0: Не произошло потери данных 1: Произошла потеря данных.
1:31	-	Резерв

### 14.5.30 Регистр TORx

Регистр переполнения FIFO передатчика x, где x = 0,1,2,3.

Формат регистра TORx приведен в Таблица 14.20.

**Таблица 14.20. Формат регистра TORx**

Номер бита	Условное обозначение	Назначение
0	TXCHO	Чтение этого бита сбрасывает прерывание по переполнению передатчика x.  0: Не произошло потери данных 1: Произошла потеря данных.
1:31	-	Резерв

### 14.5.31 Регистр RFCR0

Регистр конфигурации FIFO приемника.

Формат регистра RFCR0 приведен в Таблица 14.21.

**Таблица 14.21. Формат регистра RFCR0**

Номер бита	Условное обозначение	Назначение
1:3	RXCHDT	Это поле содержит пороговое значение заполнения FIFO приемника, при достижении которого устанавливается прерывание по наличию данных в приемнике.  Приемник должен быть выключен перед любыми изменениями этого значения. (RER0 = 0).
4:31	-	Резерв

### 14.5.32 Регистр TFCRx

Регистр конфигурации FIFO передатчика x, где x = 0,1,2,3.

Формат регистра TFCRx приведен в Таблица 14.22.

**Таблица 14.22. Формат регистра TFCRx**

Номер бита	Условное обозначение	Назначение
0:3	TXCHET	Это поле содержит пороговое значение заполнения FIFO передатчика x. Если число записей в FIFO передатчика x меньше этого значения, то устанавливается прерывание по пустому FIFO передатчика x.

		Передатчик должен быть выключен перед любыми изменениями этого значения. (TERx = 0).
4:31	-	Резерв

### 14.5.33 Регистр RFF0

Регистр сброса FIFO приемника.

Формат регистра RFF0 приведен в Таблица 14.23.

**Таблица 14.23. Формат регистра RFF0**

Номер бита	Условное обозначение	Назначение
0	RXCHFR	Запись единицы в этот регистр приводит к стиранию данных в FIFO приемника.  Перед записью в этот регистр необходимо выключить приемник (RER0[0]=0) либо блок приема (IRER[0]=0).
1:31	-	Резерв

### 14.5.34 Регистр TFFx

Регистр сброса FIFO передатчика x. Где x = 0,1,2,3.

Формат регистра TFFx приведен в Таблица 14.24.

**Таблица 14.24. Формат регистра TFFx**

Номер бита	Условное обозначение	Назначение
0	TXCHFR	Запись единицы в этот регистр приводит к стиранию данных в FIFO передатчика x.  Перед записью в этот регистр необходимо выключить передатчик x (TERx[0] = 0) либо блок передачи (ITER[0]=0).
1:31	-	Резерв

### 14.5.35 Регистр RXDMA

Регистр DMA приема.

Этот регистр позволяет цикличное чтение «левого» и «правого» слов данных из FIFO буфера приемника. После включения приемника, первое чтение этого регистра соответствует чтению регистра LRBR0, второе – RRBR0, третье – LRBR0, и т.д.

Выключение приемника в момент, когда из FIFO приемника прочитано «левое» слово, но не прочитано «правое», невозможно.

Формат регистра RXDMA приведен в Таблица 14.25.

**Таблица 14.25. Формат регистра RXDMA**

Номер бита	Условное обозначение	Назначение
0:31	RXDMA	Регистр для цикличного чтения стерео пар данных из приемника.

### 14.5.36 Регистр TXDMA

Регистр DMA передачи.

Этот регистр позволяет цикличную запись «левого» и «правого» слов данных в FIFO буферы всех включенных передатчиков. Так например, если включены нулевой и второй передатчики, то первая запись в этот регистр соответствует записи в регистра LTHR0, вторая - RTHR0, третья - LTHR2, четвертая – RTHR2, пятая – опять LTHR0 и т.д. Передатчики могут включаться и выключаться в процессе цикла записи в регистр TXDMA, однако, выключение приемника в момент, когда в него записано «левое» слово, но не записано «правое», невозможно.

Формат регистра TXDMA приведен в Таблица 14.26.

**Таблица 14.26. Формат регистра TXDMA**

Номер бита	Условное обозначение	Назначение
0:31	RXDMA	Регистр для цикличной записи стерео пар данных в передатчики.

### 14.5.37 Регистр RTXDMA

Регистр сброса DMA передачи.

Этот регистр устанавливает в качестве текущего передатчика циклической DMA передачи включенный передатчик с младшим индексом (LTHR0, при TER0[0] = 1). При этом, запись в этот регистр не будет иметь эффекта в момент, когда в FIFO текущего передатчика записано «левое» слово, но не записано «правое».

Формат регистра TXDMA приведен в Таблица 14.27.

**Таблица 14.27. Формат регистра RTXDMA**

Номер бита	Условное обозначение	Назначение
0	RTXDMA	Запись единицы в этот бит установит в качестве текущего передатчика циклической DMA передачи включенный передатчик с младшим индексом.
1:31	-	Резерв

### 14.5.38 Регистр I2S\_COMP\_PARAM\_2

Второй регистр параметров компонента.

Этот read-only регистр хранит информацию о конфигурации контроллера.

Формат регистра I2S\_COMP\_PARAM\_2 приведен в Таблица 14.28.

**Таблица 14.28. Формат регистра I2S\_COMP\_PARAM\_2**

Номер бита	Условное обозначение	Назначение
0:2	I2S_RX_WORDSIZE_0	Максимальный размер принимаемого слова: 0x0 = 12 бит 0x1 = 16 бит 0x2 = 20 бит 0x3 = 24 бит
3:5	I2S_RX_WORDSIZE_1	

		0x4 = 32 бит 0x5-0x7 = Резерв
6	-	Резерв
7:9	I2S_RX_WORDSIZE_2	Максимальный размер принимаемого слова:  0x0 = 12 бит 0x1 = 16 бит 0x2 = 20 бит 0x3 = 24 бит 0x4 = 32 бит 0x5-0x7 = Резерв
10:12	I2S_RX_WORDSIZE_3	
13:31	-	Резерв

### 14.5.39 Регистр I2S\_COMP\_PARAM\_1

Первый регистр параметров компонента.

Этот read-only регистр хранит информацию о конфигурации контроллера.

Формат регистра I2S\_COMP\_PARAM\_1 приведен в Таблица 14.29.

**Таблица 14.29. Формат регистра I2S\_COMP\_PARAM\_1**

Номер бита	Условное обозначение	Назначение
0:1	APB_DATA_WIDTH	Требуемая ширина шины APB для подключения к контроллеру:  0x0 = 8 бит 0x1 = 16 бит 0x2 = 32 бита 0x3 = резерв
2:3	I2S_FIFO_DEPTH_GLOBAL	Глубина FIFO – буферов всех каналов прима и передачи:  0x0 = 2 слова 0x1 = 4 слова 0x2 = 8 слов 0x3 = 16 слов
4	I2S_MODE_EN	Режим работы контроллера:  0x0 = Контроллер работает как Slave 0x1 = Контроллер работает как Master
5	I2S_TRANSMITTER_BLOCK	Наличие блока передачи:  0x0 = Контроллер не имеет блока передачи 0x1 = Контроллер имеет блок передачи
6	I2S_RECEIVER_BLOCK	Наличие блока приема:  0x0 = Контроллер не имеет блока приема 0x1 = Контроллер имеет блок приема
7:8	I2S_RX_CHANNELS	Число каналов приема:  0x0 – 1 канал 0x1 – 2 канала 0x2 – 3 канала 0x3 – 4 канала
9:10	I2S_TX_CHANNELS	Число каналов передачи:



Номер бита	Условное обозначение	Назначение
		0x0 – 1 канал 0x1 – 2 канала 0x2 – 3 канала 0x3 – 4 канала
11:15	-	Резерв
16:18	I2S_TX_WORDSIZE_0	Максимальный размер передаваемого слова:  0x0 = 8 бит 0x1 = 16 бит 0x2 = 32 бита 0x3 = резерв
19:21	I2S_TX_WORDSIZE_1	
22:24	I2S_TX_WORDSIZE_2	
25:27	I2S_TX_WORDSIZE_3	
28:31	-	Резерв

#### 14.5.40 Регистр I2S\_COMP\_VERSION

Регистр версии компонента.

Формат регистра I2S\_COMP\_VERSION приведен в Таблица 14.30.

**Таблица 14.30. Формат регистра I2S\_COMP\_VERSION**

Номер бита	Условное обозначение	Назначение
0:31	I2S_COMP_VERSION	Версия DesignWare компонента.

#### 14.5.41 Регистр I2S\_COMP\_TYPE

Регистр типа компонента.

Формат регистра I2S\_COMP\_VERSION приведен в Таблица 14.31.

**Таблица 14.31. Формат регистра I2S\_COMP\_TYPE**

Номер бита	Условное обозначение	Назначение
0:31	I2S_COMP_TYPE	Тип DesignWare компонента.

### 14.6 Функционирование контроллера I2S

#### 14.6.12 Включение и выключение контроллера I2S

Контроллер I<sup>2</sup>S должен быть включен перед любым обменом данными по шине I<sup>2</sup>S. Для включения контроллера необходимо записать 1 в бит IEN регистра IER. Для выключения контроллера необходимо записать 0 в бит IEN регистра IER.

После выключения контроллера произойдет следующее:

1. Все содержимое FIFO приемников и передатчика будет стерто.
2. Любые данные, находящиеся в процессе передачи или приема будут потеряны.
3. Будут выключены блоки приема и передачи.

4. Прекратится генерация внутренних сигналов управления выходной частотой SCLK\_EN, SCLK\_GATE и сигнала выбора слова WS\_OUT.

Когда контроллер I<sup>2</sup>S включен, любой обмен данными всегда начинается с «левого» слова данных (WS=0). Завершение обмена всегда происходит на следующий такт после «правого» слова данных (WS=1).

Блок передачи I<sup>2</sup>S должен быть включен перед любой записью данных на шину I<sup>2</sup>S. Для включения блока передачи необходимо записать 1 в бит TXEN регистра ITER. Для выключения необходимо записать 0 в бит TXEN регистра ITER.

После выключения блока передачи произойдет следующее:

1. Будут выключены все передатчики блока передачи (все TER<sub>x</sub>[0] = 0).
2. Данные, передаваемые в данный момент, будут потеряны, и линии SDO будут удерживать постоянный низкий уровень.
3. Все данные в FIFO каждого передатчика сохранятся, также возможно будет осуществлять дальнейшую запись данных в FIFO каждого передатчика.
4. Все предыдущие настройки параметров передатчиков (длина слова, пороговое значение заполнения FIFO и т.д.) сохранятся.
5. Когда блок передачи выключен, возможно выполнение следующих операций:
  - 5.1. Настройка параметров передатчиков.
  - 5.2. Стирание данных во всех FIFO блока передачи. Для этого необходимо записать 1 в бит TXFFR[0].
  - 5.3. Стирание данных в FIFO конкретного передатчика x. Для этого необходимо записать 1 в бит TFF<sub>x</sub>[0]. Эту операцию можно проводить и при включенном блоке передаче, но выключенном передатчике x.

Блок передачи включает в себя четыре независимых передатчика, каждый из которых может включаться и выключаться независимо от остальных. Включение и выключение передатчика x управляется через бит TXCHEN<sub>x</sub> регистра TER<sub>x</sub> при включенном блоке передачи (ITER[0]=1).

Блок приема контроллера I<sup>2</sup>S должен быть включен перед любым чтением данных с шины I<sup>2</sup>S. Для включения блока приема необходимо записать 1 в бит RXEN регистра IREX. Для выключения блока приема необходимо записать 0 в бит RXEN регистра IREX.

После выключения блока приема контроллера I<sup>2</sup>S произойдет следующее:

- приемник блока приема будет выключен (RER0[0]=0);
- данные, принимаемые в текущий момент, будут потеряны, и линия SDO будет удерживать постоянный низкий уровень;
- все данные в FIFO приемника сохранятся, также возможно будет осуществлять дальнейшее чтение данных из FIFO приемника;

- все предыдущие настройки параметров приемника (длина слова, пороговое значение заполнения FIFO и т.д.) сохраняются;
- когда блок приема выключен, возможно выполнение следующих операций:
  - настройка параметров приемника;
  - стирание данных из FIFO приемника. Для этого необходимо записать 1 в бит RXFFR[0], либо 1 в бит RFF0[0].

Блок приема состоит из одного приемника, который можно включать и выключать при включенном блоке приема.

### 14.6.13 Блок управления тактовым сигналом SCLK

Для генерации сигналов SCLK и WS\_OUT необходимо установить 1 в бит CLKEN регистра CER. Установка 0 в бит CLKEN регистра CER прекращает генерацию сигналов SCLKO и WS\_OUT. Любые изменения параметров генерации сигналов GATED\_SCLK и WS\_OUT должны производиться при выключенной генерации. Сигнал SCLK\_EN обеспечивает выключение генерации сигнала GATED\_SCLK при выключении контроллера I<sup>2</sup>S.

Сигнал SCLK\_GATE обеспечивает периодическое «запирание» выходного тактового сигнала, как показано на Рисунок 14.1.

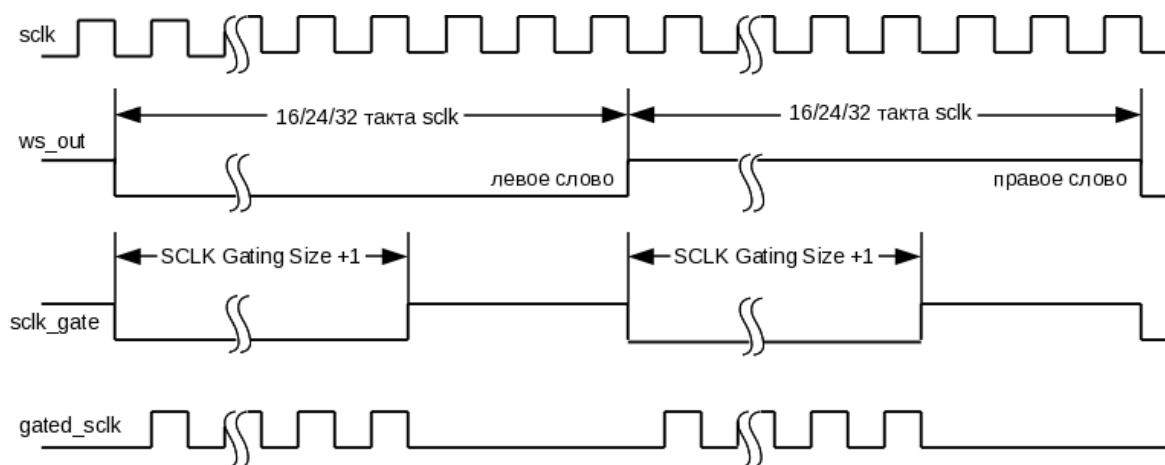


Рисунок 14.1. Формирование выходной частоты GATED\_SCLK

## 15. CAN: КОНТРОЛЛЕРНАЯ СЕТЬ

### 15.1 Введение

#### 15.1.1 Ядро CAN-CTRL

Ядро CAN-CTRL - это контроллер последовательной связи, который осуществляет последовательную связь в соответствии с протоколом CAN. Этот интерфейс шины CAN использует базовый принцип CAN и отвечает всем ограничениям CAN-спецификации 2.0B active. Кроме того, это ядро CAN может быть сконфигурировано в соответствии со спецификацией CAN с гибкой скоростью передачи данных CAN FD.

Протокол CAN использует многомастерную конфигурацию шины для передачи кадров (объектов связи) между узлами сети и управляет обработкой ошибок без нагрузки на центральный процессор. Контроллер шины CAN-CTRL позволяет пользователю устанавливать экономичные и надежные связи между различными компонентами. Ядро CAN-CTRL представляется микроконтроллеру как устройство входа/выхода с привязкой к памяти. Центральный процессор обращается к ядру CAN-CTRL для управления передачей или приемом кадров через двухпроводную систему шины CAN. Подключение к шине CAN показано ниже на Рисунок 15.1.

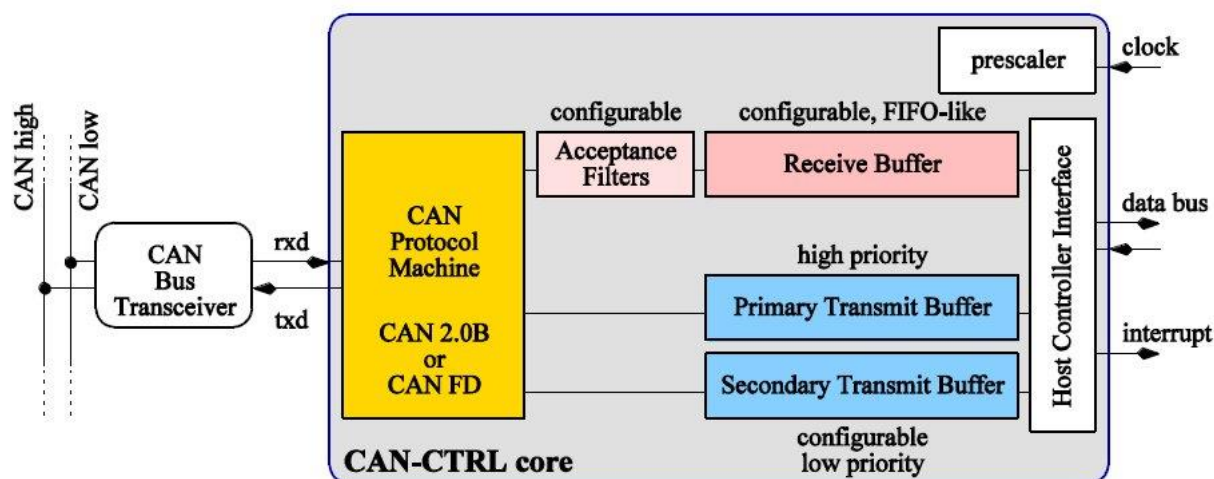


Рисунок 15.1. Подключение к шине CAN и основные функции ядра CAN-CTRL

#### 15.1.2 Протокол CAN

Коммуникация CAN организована в виде кадров. Существует два типа кадров: стандартные и расширенные кадры. Для CAN 2.0 максимальная полезная нагрузка данных составляет до 8 байт, в то время как для CAN FD с использованием одного кадра может быть передано до 64 байт.

Все узлы CAN равны с точки зрения доступа к шине. Суперузла нет, потому что CAN - это многомастерная шина.

Адресация данных осуществляется с использованием идентификаторов сообщений. В сети CAN только один узел должен передавать сообщения с определенным идентификатором. Все узлы получают все сообщения, и хост-контроллер узла должен решить, было ли оно адресовано соответствующим идентификатором сообщения. Чтобы уменьшить нагрузку на хост-контроллер, ядро CAN может использовать фильтры приема. Эти фильтры сравнивают все идентификаторы полученных сообщений с битовыми шаблонами, выбираемыми пользователем. Только в том случае, если сообщение проходит фильтр приема, оно будет сохранено в буфере приема и передано на главный контроллер.

Идентификаторы кадров CAN также используются для арбитража шины. Протокольный автомат CAN останавливает передачу сообщения с идентификатором с низким приоритетом, когда сообщение с идентификатором с более высоким приоритетом передается другим узлом CAN. Устройство протокола CAN автоматически пытается повторно передать остановленное сообщение в следующей возможной позиции передачи.

CAN 2.0B определяет скорость передачи данных до 1 Мбит/с. Для CAN FD нет фиксированного ограничения. Для CAN FD стандарт определяет переключение скорости передачи битов. Если включено, передача полезной нагрузки кадров может выполняться с более высокой скоростью, в то время как заголовок кадра передается с более низкой скоростью.

### 15.1.3 Классический CAN 2.0B и как включить CAN FD

CAN FD - это нисходящее совместимое расширение классического протокола CAN 2.0B. Каждый узел CAN FD способен принимать и передавать классические кадры CAN. CAN-CTRL можно настроить для поддержки только классический CAN или как классический CAN, так и CAN FD.

Параметр синтеза CAN\_FD определяет, включен ли CAN FD в аппаратное обеспечение или исключен (см. главу "Параметры конфигурации"). Для включения CAN FD в систему потребуется приблизительно на 6% больше площадь ячейки и больше слотов буферов приема и передачи. (Классический CAN несет полезную нагрузку до 8 байт, и CAN FD - до 64 байт.) Для современных систем рекомендуется включать функции CAN FD.

Если параметр синтеза CAN\_FD настроен на включение функций CAN FD, то входной вывод <can\_fd\_enable> может принудительно переключать CAN-CTRL только на классическую операцию CAN или может разблокировать CAN FD. Эта функция может быть использована для ограничения системы классическим CAN и последующей разблокировки CAN FD. (см. главу "Аппаратный интерфейс").

Если параметр синтеза CAN\_FD настроен на включение функций CAN FD, а входной вывод <can\_fd\_enable> настроен на включение CAN FD, то встроенное ПО хост-контроллера может выбирать во время выполнения для каждого кадра, будет ли кадр передаваться как классический кадр CAN или CAN FD. Это делается с помощью битов конфигурации в буфере передачи. Для принятых кадров биты состояния в сигнале буфера приема, если принятый кадр был классическим кадром CAN или CAN FD.

Дополнительная информация: Если параметр синтеза CAN\_FD установлен для исключения функций CAN FD, то CAN-CTRL - это чисто классический узел CAN, но он совместим с восходящим потоком или “CAN FD tolerant”. Это означает, что CAN-CTRL будет игнорировать кадры CAN FD и не уничтожит их. Поэтому его можно использовать в смешанной сети, где некоторые узлы используют классический CAN, а некоторые другие используют CAN FD.

## 15.1.4 Временная метка

### 15.1.4.1 Синхронизируемый по времени CAN

Опционально CAN-CTRL можно использовать для связи через синхронизируемый по времени CAN (TTCAN) в соответствии с ISO 11898-4. CAN-CTRL обеспечивает частичную аппаратную поддержку и требует взаимодействия с программным обеспечением хоста в режиме реального времени в этом режиме.

Основная концепция TTCAN заключается в том, чтобы иметь таймер для полученных сообщений с временной меткой и для запускающих сообщений для передачи. Один узел в сети CAN является мастером времени. Мастер времени передает опорное сообщение. С помощью опорного сообщения начинается время цикла. Время между двумя опорными сообщениями является базовым циклом. Внутри базового цикла сообщения могут передаваться во временных окнах. Системный администратор TTCAN определяет начало и продолжительность каждого временного окна во время автономной настройки.

Существует три типа окон времени:

- эксклюзивное окно времени (только одному узлу разрешено передавать один кадр с определенным идентификатором),
- свободное окно времени (неиспользуемое окно времени для дальнейшего расширения сети),
- окно времени арбитража (несколько узлов могут передавать кадр, и происходит арбитраж).

Если кадр получен, он получает фактическое время цикла в качестве метки времени. Для передач CAN-CTRL предлагает аппаратный триггер, который запускает передачу предопределенного сообщения в заданное время цикла.

CAN-CTRL автоматически определяет опорное сообщение при приеме и запускает время цикла. Аппаратный таймер представляет собой 16-разрядный таймер, работающий с битовым временем CAN, как определено для уровня 1, ISO 11898-4. CAN-CTRL можно использовать в качестве мастера времени.

Помимо аппаратного триггера для передачи сообщений CAN-CTRL предлагает сторожевой триггер для обнаружения отсутствующего опорного сообщения.

Частичная аппаратная поддержка означает, что хост-контроллер должен подготавливать действия узла для каждого окна времени. Например, хосту необходимо определить сообщение для следующей передачи и определить время запуска для него. Другими словами: хост должен позаботиться о столбцах передачи и системной матрице.

#### **15.1.4.2      Временная метка CiA 603**

CAN в автоматизации (CiA) в спецификации 603 определяет метод для временной метки по меньшей мере с 16 битами, который опционально поддерживается CAN-CTRL. Его можно использовать в дополнение к TT CAN.

Основная концепция CiA 603 заключается в том, чтобы иметь таймер свободного хода, который подсчитывает такты, а не время битов CAN. Точность должна составлять не менее 10 мкс (16 бит) или 1 мкс (32 бита или более). Временные метки получаются в SOF или EOF кадра CAN / CAN FD.

Предполагается, что CiA 603 поддерживает временную метку и синхронизацию времени AUTOSAR. Для AUTOSAR один узел в сети CAN является ведущим по времени. Мастер времени передает сообщение синхронизации (сообщение SYNC). Временная метка сообщения SYNC извлекается мастером времени и всеми слэями времени. Разница во времени между событием отправки сообщения SYNC и моментом, когда сообщение SYNC действительно будет передано, будет передана в последующем сообщении (FUP) мастером времени.

CiA определяет правила для считывания таймера, а также для изменения таймера. CAN-CTRL не включает таймер, но использует внешний таймер. CAN-CTRL включает в себя только механизм временной метки, регистр для хранения одной метки времени передачи (DTS) и память для хранения меток времени приема (RTS) для всех принятых сообщений.

### **15.1.5    Использование ядра CAN-CTRL**

Руководство по интеграции (INM) содержит краткое описание основных шагов по использованию ядра CAN-CTRL в специальной главе 15.1.5 “Как использовать ядро CAN-CTRL”. Его можно использовать в качестве обзора на первых этапах изучения того, как работает ядро, а также в качестве краткого содержания, когда основные функции поняты.



## 15.2 Характеристики

### 15.2.1 Список характеристик

- Поддерживает спецификацию CAN:
  - CAN 2.0B (полезная нагрузка до 8 байт, проверенная опорной моделью Bosch),
  - Дополнительная поддержка CAN FD (полезная нагрузка до 64 байт, ISO 11898-1:2015 или не соответствует стандарту ISO Bosch).
- Свободно программируемая скорость передачи данных:
  - CAN 2.0B определяет скорость передачи данных до 1 Мбит/с,
  - CAN FD ограничен приемопередатчиком и тактовой частотой CAN-CTRLcore.
- Программируемый предделитель скорости передачи данных в бодах (от 1 до 1/256),
- Отдельные частотные домены для интерфейса хоста и протокольного автомата CAN,
- Настраиваемый размер приемного буфера (RB):
  - Общий параметр выбирает количество слотов буфера,
  - Поведение, подобное FIFO,
  - Полученные сообщения с надписью “не принято” или “неверно” не перезаписывают уже сохраненные сообщения.
- Два буфера передачи:
  - один основной буфер передачи (PTB),
  - дополнительный настраиваемый вторичный буфер передачи (STB).
    - STB является необязательным. Общий параметр выбирает количество слотов буфера,
    - Работа в режиме FIFO или приоритетного решения.
- Независимые и программируемые внутренние 29-битные приемные фильтры:
  - Количество приемных фильтров, выбираемых общим параметром в диапазоне от 1 до 16.



- Расширенные характеристики:
  - Режим передачи одного кадра (для РТВ и/или для STB),
  - Режим только прослушивания,
  - Режим обратной связи (внутренний и внешний),
  - Режим ожидания приемопередатчика.
- Расширенный отчет о состоянии и ошибках:
  - Фиксация последнего произошедшего вида ошибки и арбитражной потерянной позиции,
  - Программируемый предел предупреждения об ошибках.
- Различные интерфейсы хост-контроллера:
  - 32-разрядный интерфейс синхронного хост-контроллера; оболочка для 8-разрядных хостов,
  - 32-разрядная спецификация протокола AMBA APB версии v2.0,
  - 32-разрядный протокол AMBA 3 AHB-Lite версии v1.0,
  - 32-разрядная версия Avalon-MM 2018.09.26, простой интерфейс (без конвейерной обработки),
  - 32-разрядный поперечный рычаг,
  - Дополнительный интерфейс для конкретного приложения к хост-контроллеру по запросу.
- Настраиваемые источники прерываний
- Один двухпортовый блок памяти или два псевдодвухпортовых блока памяти для буферов кадров,
- Метка времени:
  - Синхронизируемый по времени CAN ISO 11898-4 с частичной аппаратной поддержкой,
  - Временная метка CiA 603.
- Функции, связанные с безопасностью (опционально):

- Защита памяти ECC с дополнительной защитой адреса,
  - Пространственная избыточность для внутреннего логического ядра,
  - ISO 26262 “ASIL B ready”: Доступно руководство по технике безопасности и FMEDA.
- Полностью синхронный и синтезируемый дизайн HDL (Verilog 2001, VHDL 93),
  - Совместимость с AUTOSAR,
  - Оптимизирован для SAE J1939,
  - Драйвер Linux включен в комплект поставки.

### 15.2.2 Параметры

- ACF\_NUMBER\_MAX = 16
- RBUF\_SLOTS = 16
- STB\_ENABLE = 1
- STB\_SLOTS = 16
- STB\_PRIO = 1
- ACF\_NUMBER = 16
- CAN\_FD = 1
- UPWARD\_COMPATIBILITY = 1
- TTCAN = 1
- CIA603 = 1
- RAM\_ECC = 0
- RAM\_ES = 0

### 15.2.3 Прямая совместимость

Спецификация CAN включает зарезервированные биты (см. Рисунок 15.4) для расширения протокола. Это было использовано для создания спецификации CAN FD поверх спецификации CAN 2.0B. Зарезервированные биты передаются с низким уровнем (доминирующим), если они не используются. К сожалению, спецификация CAN 2.0B определяет поведение для случая, когда зарезервированный бит является высоким (рецессивным), чтобы принять это и продолжить работу с кадром. Следовательно, если кадр CAN FD (который имеет другую и неожиданную форму по сравнению с кадром CAN 2.0B) принимается узлом CAN 2.0B, который использует это поведение, то это приведет к кадру ошибки узлом CAN 2.0B, который уничтожает кадр. Такое поведение называется “CAN FD intolerant”.

Для обеспечения прямой совместимости с новыми спецификациями протокола должно иметь место так называемое событие исключения протокола, если узел обнаруживает высокий уровень зарезервированного бита. Это справедливо как для CAN 2.0B, так и для узлов CAN FD. Событие исключения протокола не приводит к каким-либо действиям для получателя. Приемник просто игнорирует этот кадр, не генерирует АСК, ожидает простоя шины и может передавать или принимать следующий кадр. Для узла CAN 2.0B это называется “CAN FD tolerant” и позволяет сосуществовать кадрам CAN 2.0B и FD в одной сети.

Более старые тесты соответствия CAN 2.0B проверяют поведение “CAN FD intolerant”, но рекомендуется использовать новое поведение “CAN FD tolerant” или в целом событие исключения протокола, чтобы быть совместимым по восходящей, независимо от того, является ли узел узлом CAN 2.0B или CAN FD.

CAN-CTRL предоставляет общий параметр для выбора поведения (глава "Параметры конфигурации", параметр UPWARD\_COMPATIBILITY).

## 15.2.4 Буферы сообщений

### 15.2.4.1 Концепция буферов сообщений

Концепция буферов сообщений проиллюстрирована ниже на Рисунок 15.2. Эта схема фокусируется на буферах и скрывает другие детали ядра CAN-CTRL. Все слоты буферов достаточно велики для хранения кадров максимальной длины.

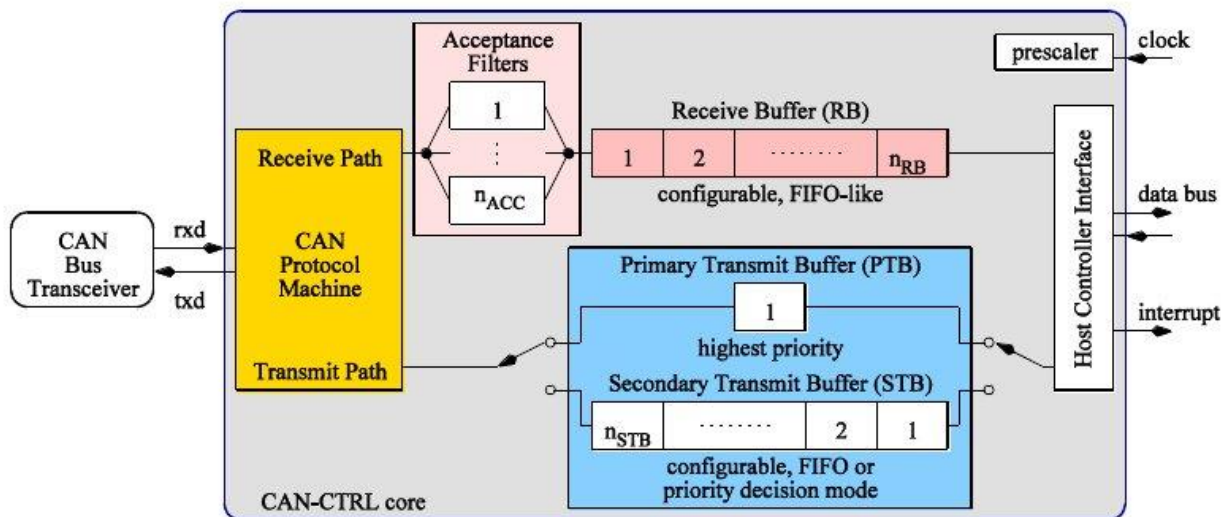


Рисунок 15.2. Концепция буферов сообщений

### 15.2.4.2 Буфер приема

Чтобы уменьшить нагрузку на кадр приема для хост-контроллера, ядро использует фильтры приема. Ядро CAN-CTRL проверяет идентификатор сообщения во время фильтрации приема. Если принятый кадр соответствует критериям фильтрации одного из фильтров

приема, то он будет сохранен в буфере приема (RB), который имеет поведение, подобное FIFO.

В зависимости от количества доступных слотов сообщений хост-контроллеру не нужно немедленно считывать входящие сообщения. Ядро CAN-CTRL способно генерировать прерывания при каждом полученном сообщении, когда RB заполнен или заполнен до выбираемого пользователем предела “почти заполнен”. Из-за поведения, подобного FIFO, хост-контроллер всегда считывает самое старое сообщение из RB.

#### **15.2.4.3 Буфер передачи**

Для целей передачи кадров предлагаются два буфера передачи (TB). Первичный TB (PTB) имеет более высокий приоритет, но способен буферизировать только один кадр. Вторичный TB (STB) имеет более низкий приоритет. Он может действовать в режиме FIFO или в режиме приоритета. Решение о приоритете между PTB и STB является фиксированным и полностью независимым от арбитража шины CAN. Арбитраж шины - это приоритетное решение, основанное на идентификаторах кадров.

STB можно командовать на передачу одного или всех сохраненных кадров. В режиме FIFO при каждой передаче сначала передается самый старый кадр внутри этого буфера. В режиме приоритета кадр с наивысшим приоритетом внутри этого буфера (на основе идентификатора кадра) передается первым.

Кадр, расположенный в PTB, всегда имеет более высокий приоритет для протокольного автомата CAN, чем кадры в STB, независимо от идентификаторов кадров. Передача PTB останавливает и задерживает передачу STB. Передача STB автоматически перезапускается после успешной передачи кадра PTB.

Передача RTP начинается со следующей позиции передачи, которая возможна по протоколу CAN (после следующего межкадрового промежутка). Из-за этого передача STB, выигравшая арбитраж и фактически переданная, будет завершена раньше.

Прерывание передач STB с использованием передачи PTB может произойти в следующих случаях:

1. STB получает команду на вывод всех сохраненных кадров, и главный контроллер решает дать команду на передачу PTB до завершения всех передач STB.
2. STB получает команду на вывод одного кадра, и главный контроллер решает дать команду на передачу PTB до завершения передачи STB.

Если главный контроллер ожидает завершения каждой командной передачи, то он может легко решить, какой буфер должен передавать следующий кадр. В качестве недостатка сообщение с идентификатором низкого приоритета может блокировать более важные

сообщения. Затем хост может прервать сообщение (глава 15.2.4.5 "Пример применения буфера передачи в приоритетном режиме").

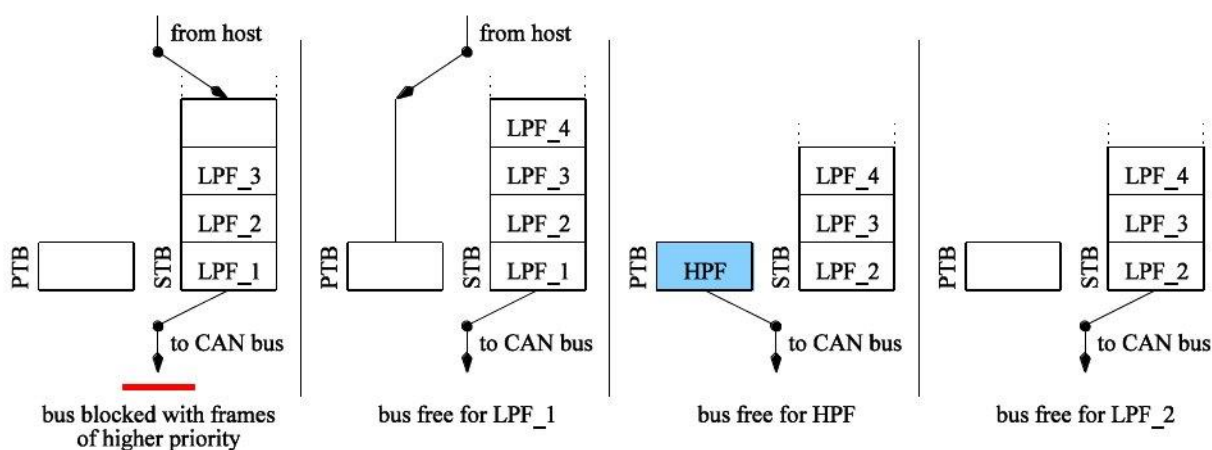
#### 15.2.4.4 Пример применения буфера передачи в режиме FIFO

В примере узел CAN используется для измерений датчиков. Системный инженер CAN решил обрабатывать эти измерения датчиков с низким приоритетом. Поэтому идентификаторы кадров с низким приоритетом используются для кадров CAN, несущих данные датчика.

Главный контроллер автоматически получает результаты измерений и помещает их в виде кадров CAN в STB. Из-за высокого трафика на шине CAN может случиться так, что кадры данных датчика не могут быть переданы немедленно, и несколько кадров остаются в STB в течение некоторого времени. Позже, если на шине CAN будет меньше трафика, они будут переданы.

В ситуации, когда в STB остается несколько кадров, может произойти событие, которое заставит хост-контроллер вывести важный высокоприоритетный кадр. В такой ситуации хост может использовать PTB. Кадр в PTB всегда будет передаваться перед всеми кадрами в STB.

Преимуществом наличия двух буферов передачи является возможность сохранения всех сообщений. Никакое сообщение не должно быть прервано (отброшено) в случае события с высоким приоритетом.



**Рисунок 15.3. Пример применения буфера передачи (TSALL=1)**

Приведенный пример применения пошагово объясняется выше на Рисунок 15.3. На этом рисунке LPF\_x - это сокращение от “низкоприоритетный кадр”, а HPF означает “высокоприоритетный кадр”. Таким образом, LPF\_1 блокируется другими кадрами с более высоким приоритетом от других узлов CAN в первой части рисунка. Хост-контроллер помещает четвертый кадр в STB (LPF\_4), а затем принимает решение о выводе HPF. Между

тем, шина CAN была свободна для передачи LPF\_1, за которым следуют HPF, LPF\_2 и так далее.

Решение о приоритете между РТВ и STB принимается протокольным автоматом CAN, но сам протокол CAN использует другой независимый механизм принятия решения о приоритете, который называется арбитражем шины. Для арбитража шины идентификатор кадра определяет уровень приоритета.

В приведенном выше примере можно было бы поместить кадр с идентификатором с очень низким приоритетом в РТВ, в то время как кадры с более высоким приоритетом остаются в STB. При принятии решения между РТВ и STB всегда выигрывает РТВ, независимо от идентификатора кадра. Задачей хост-контроллера является размещение кадров со значимыми приоритетами идентификаторов в соответствующих буферах.

#### **15.2.4.5 Пример применения буфера передачи в приоритетном режиме**

В режиме приоритета для STB CAN-CTRL автоматически изменяет порядок кадров внутри STB. В результате кадр с наивысшим приоритетом передается первым. Решение о приоритете такое же, как и для арбитража шины, и поэтому идентификатор кадра с меньшим значением имеет более высокий приоритет. Независимо от этого РТВ всегда имеет наивысший приоритет.

Изменение порядка кадров внутри STB выполняется автоматически, как только новый кадр подготовлен к передаче и записан в STB. В результате для передачи выбирается кадр с наивысшим приоритетом. Для хост-контроллера это принцип "выстрелил и забыл" (fire-and-forget): ему просто нужно записать новый кадр в STB, пока активен режим приоритета.

В общем случае кадр с более высоким приоритетом никогда не будет прерывать активную передачу. Например, выше на Рисунок 15.3 в самой левой части, кадр LPF\_1 в данный момент находится в процессе передачи, в то время как LPF\_4 записывается в STB. Даже если LPF\_4 имеет более высокий приоритет, чем LPF\_1, все равно LPF\_1 будет передаваться до тех пор, пока он не потеряет арбитраж на шине. Затем, после того, как узел проиграл арбитраж, LPF\_4 будет выбран для передачи из-за его приоритета.

Изменение приоритета автоматически выполняется в фоновом режиме, но занимает некоторое время. Во время большой нагрузки, когда в STB записывается много новых кадров с разными приоритетами, и узел теряет арбитраж или на шине возникают ошибки, может случиться так, что выбран не кадр с наивысшим приоритетом. Это может произойти, если этот кадр был написан слишком незадолго до крайнего срока принятия решения о том, какой кадр будет следующим. Но это очень редкое состояние, и в целом хозяина это не должно волновать.

Если в STB записаны два кадра с одинаковым приоритетом, то самый старый кадр будет передан первым, независимо от любого изменения порядка с другими кадрами с более низким или более высоким приоритетом.

#### 15.2.4.6 Прерывание передачи

Если возникает ситуация, когда сообщение в буфере передачи не может быть отправлено из-за его низкого приоритета, это приведет к блокировке буфера на длительное время. Чтобы избежать этого, хост-контроллер может отозвать запрос на передачу и прервать сообщение. Прерывание возможно для РТВ и STB, а также для передачи одного или всех кадров.

#### 15.2.4.7 Буфер передачи в режиме TTCAN

В режиме сканирования каждый слот в STB может быть адресован хостом. Каждый слот может быть определен как заполненный или пустой. Поэтому каждый слот может быть посвящен одному конкретному сообщению.

Передача в режиме сканирования запускается с помощью триггеров. Каждый триггер включает в себя указатель на слот сообщения.

РТВ не имеет особого приоритета в режиме сканирования и поэтому обрабатывается так же, как слот STB.

Для хост-приложения удобно выделять одно сообщение в один слот буфера. Затем задача на хосте может обновить сообщение в любое время, а затем, если триггер для этого сообщения активен, оно может быть передано. Но наличие достаточного количества слотов для всех сообщений требует большого количества аппаратного обеспечения. Из-за частичной аппаратной поддержки стандарта ISO 11898-4 можно использовать слот буфера для более чем одного сообщения в базовом цикле. Поэтому ISO 11898-4 возможен, даже если STB отключен, а ядро включает только один слот для сообщений РТВ.

### 15.2.5 Кадры CAN 2.0 и CAN FD

CAN FD - это расширение протокола CAN 2.0. Основными отличиями являются:

- Полезная нагрузка данных: до 8 байт для CAN 2.0 и до 64 байт для CAN FD,
- Одна настраиваемая скорость передачи данных для CAN 2.0, но 2 для CAN FD: медленная для арбитража и быстрая для фазы передачи данных.

Все типы кадров для CAN 2.0 и CAN FD показаны ниже на Рисунок 15.4. Сокращения поясняются в спецификации CAN и вкратце ниже в Таблица 15.1. Для классических кадров CAN (CAN 2.0) некоторые названия битов переименованы в соответствии со спецификацией CAN FD ISO, но здесь по-прежнему используются старые названия для упрощения обратной ссылки.



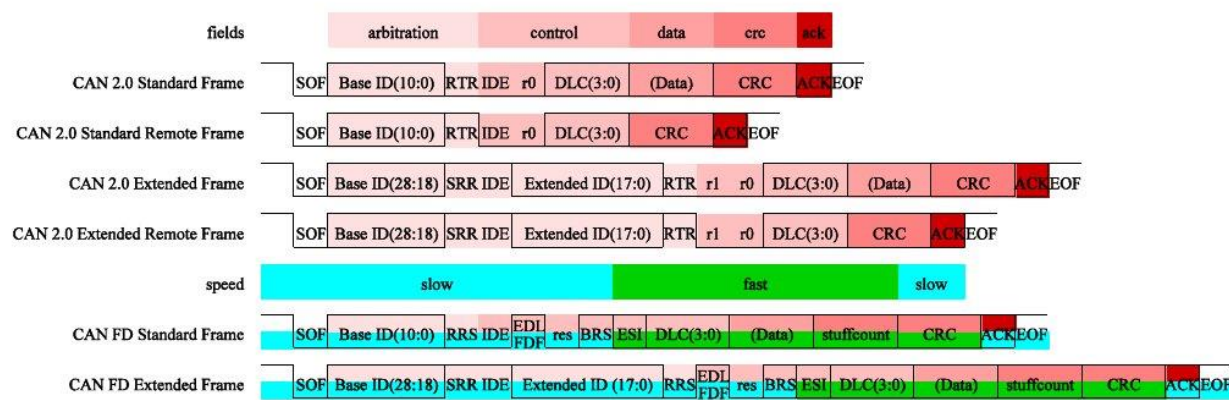


Рисунок 15.4. Типы кадров CAN 2.0 и CAN FD

Таблица 15.1. Битовые аббревиатуры CAN

Аббревиатура	Описание	Кадры
EDL	Длина расширенных данных (Extended Data Length)	Кадр CAN 2.0 или CAN FD
FDF	Индикатор формата FD (=EDL) (FD Format indicator (=EDL))	Кадр CAN 2.0 или CAN FD
RTR	Удаленный запрос передачи (Remote Transmission Request)	Кадр удаленного запроса или кадр данных
DLC	Код длины данных (Data Length Code)	Количество байтов полезной нагрузки
IDE	Расширение идентификатора (Identifier Extension)	Стандартный или расширенный кадр
ID	Идентификатор	
SRR	Удаленный запрос замещения (Substitute Remote Request)	
RRS	Замещение удаленного запроса (Remote Request Substitution)	
BRS	Переключение скорости передачи битов (Bit Rate Switch)	
ESI	Индикатор состояния ошибки (Error State Indicator)	
r1, r0, res	Зарезервированные биты	

В спецификации CAN FD от Bosch (не ISO) используется название EDL, в то время как в спецификации CAN FD ISO для того же бита используется название FDF. Оба названия являются синонимами. В этом документе используется название FDF.

Для ISO-кадров CAN FD количество элементов передается как часть поля CRC. Для кадров CAN FD, отличных от ISO, количество материала не является частью кадра. Кроме того, средства проверки CRC имеют различную инициализацию для кадров, отличных от ISO и ISO. Поэтому кадры ISO и не-ISO несовместимы.

Протокольный автомат CAN в ядре CAN-CTRL автоматически передает и принимает кадры и вставляет соответствующие управляющие и статусные биты. От хост-контроллера



требуется выбрать желаемый тип кадра (IDE, RTR, FDF), выбрать идентификатор и установить полезную нагрузку данных.

## 15.2.6 AUTOSAR и SAE J1939

И AUTOSAR, и SAE J1939 представляют собой программные стеки, которые можно использовать с CAN-CTRL.

Для автозапуска количество буферов передачи должно быть установлено как минимум на 3 слота (общий параметр STB\_SLOTS), и передачи могут быть прерваны (биты TPA и TSA в регистре TCMD). Метка времени и синхронизация времени поддерживаются с помощью аппаратной отметки времени в соответствии с CiA 603.

SAE J1939 использует расширенные идентификаторы (29 бит). Для удобства работы фильтры приема могут быть сконфигурированы так, чтобы принимать только расширенные идентификаторы (биты AIDE и AIDE в регистре ACF\_3).

## 15.2.7 Определения

Таблица 15.2. Определения названий

Аббревиатура	Определение
ACF	Приемный фильтр (Acceptance Filter)
PTB	Основной буфер передачи (высокий приоритет)(Primary Transmit Buffer (high priority))
RB	Буфер приема (Receive Buffer)
RDC	Компенсация задержки приемника (связанная с TDC)(Receiver Delay Compensation (related to TDC))
RTS	Метка времени приема (Reception Time Stamp)
SP	Точка выборки (Sample Point)
SSP	Вторичная точка выборки (CAN FD)(Secondary Sample Point (CAN FD))
STB	Вторичный буфер передачи (низкий приоритет)(Secondary Transmit Buffer (low priority))
TDC	Компенсация задержки передатчика (спецификация CAN FD)(Transmitter Delay Compensation (CAN FD specification))
TTCAN	CAN с синхронизацией по времени (ISO 11898-4) (Time-Triggered CAN (ISO 11898-4))
TTS	Метка времени передачи (Transmission Time Stamp)
TQ	Квант времени (спецификация CAN)(Time Quanta (CAN specification))

Приведены все определения регистров, включая определение доступа. Определения доступа даются с использованием аббревиатуры в виде <доступ>-<сброс>. Возможными сокращениями атрибута <access> являются “r” для чтения, “w” для записи и “rw” для доступа на чтение/запись. Атрибут <reset> может быть “0”, “1” и “u” для неинициализированных регистров. Пример: “rw-0” означает “читаемый и записываемый и сбрасываемый на 0”, в то время как “r-u” означает “читаемый и неинициализированный”. Неиспользуемые биты всегда равны r-0.

## 15.2.8 Перекрытие диапазонов изменения тактовых частот (Clock Domain Crossing)

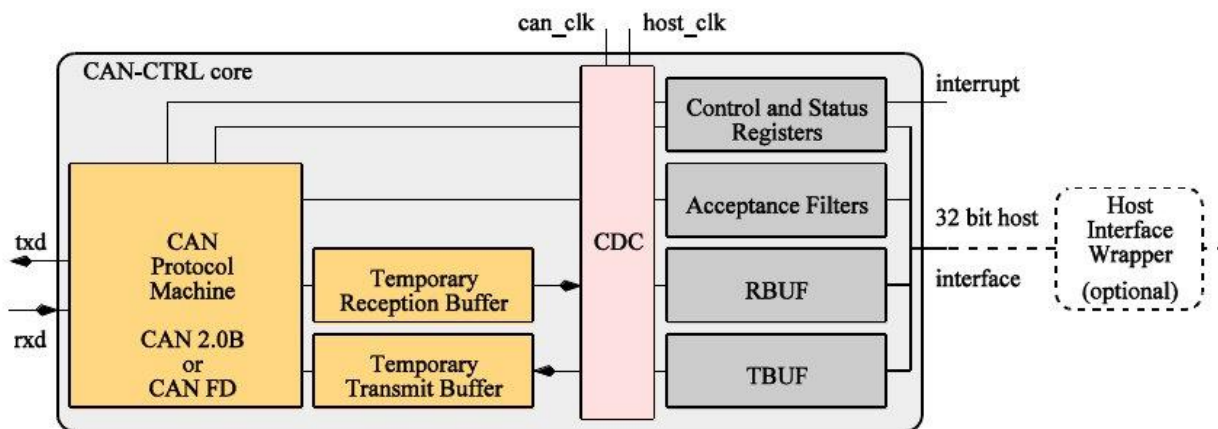


Рисунок 15.5. Перекрытие диапазонов изменения тактовых частот

Ядро использует два домена синхронизации: один для интерфейса хоста и один для протокольного автомата CAN. Это позволяет использовать устройство протокола CAN с тактовой частотой, которая наилучшим образом соответствует требуемым скоростям передачи данных по шине CAN, в то время как хост может свободно работать с другой (более высокой) тактовой частотой.

Интерфейс хоста, а также дополнительные оболочки интерфейса хоста (например, для шин AMBA) работают в домене синхронизации хоста. Сюда входят все запоминающие устройства и регистры, доступные с помощью шины данных хоста (см. главу 15.2.9 "Программный интерфейс"). Пересечение домена синхронизации (CDC) осуществляется с помощью синхронизаторов с двойной буферизацией и двунаправленных механизмов квитирования. Ниже приведены некоторые подробности о CDC:

- Регистры управления и состояния перемещаются в соответствующий частотный домен по отдельности. Например, флаги прерывания используют двунаправленное квитирование, в то время как регистры состояния используют однонаправленные синхронизаторы.
- Фильтрация приема выполняется в частотном домене хоста, и необходимые управляющие сигналы передаются с помощью CDC.
- Принятые кадры сохраняются в небольшом временном буфере приема. Этот временный буфер содержит только части кадров, и при необходимости эти части копируются в RBUF.

- Кадры, подлежащие передаче, копируются шаг за шагом из TBUF в небольшой временный буфер передачи.

Использование синхронизаторов и механизмов квитирования приводит к некоторой задержке. Для двунаправленного квитирования требуется до 3 тактовых сигналов в каждом домене. Но в большинстве случаев для хост-контроллера это не будет проблемой.

## 15.2.9 Программный интерфейс

CAN-CTRL изначально разрабатывался как периферийный компонент для 8-разрядных систем, и поэтому регистры управления и состояния определены как 8-разрядные группы (Таблица 15.3). Тем не менее, CAN-CTRL является 32-разрядным компонентом и предлагает интерфейсы, совместимые с NVIDIA, для 8- и 16- разрядных хостов. Сопоставление регистров для 8 / 16/ 32 интерфейсы хоста показаны ниже на Рисунок 15.6. В главе 15.6.2.1 "Интерфейсы хоста" приведены подробные сведения о различных интерфейсах хоста.

**Таблица 15.3. Карта регистров**

	Позиция бита								Название регистра	Размер, бит
	7	6	5	4	3	2	1	0		
с 0x00 по 0x4f	Регистры буфера приёма и метка времени приёма								RBUF (и RTS)	8
с 0x50 по 0x97	Регистры буфера передачи								TBUF	8
с 0x98 по 0x9f	Метка времени передачи								TTS	8
0xa0	RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTVE	BUSOFF	CFG_STAT	8
0xa1	TBSEL	LOM	STBY	TPE	TPA	TSONE	TSALL	TSA	TCMD	8
0xa2	FD_ISO <sup>*)</sup>	TSNEXT	TSMODE	TTTBM	-		TSSTAT(1:0)		TCTRL	8
0xa3	SACK	ROM	ROV	RREL	RBALL	-		RSTAT(1:0)	RCTRL	8
0xa4	RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF	RTIE	8
0xa5	RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF	RTIF	8
0xa6	EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF	ERRINT	8
0xa7	AFWL(3:0)				EWL(3:0)				LIMIT	8
0xa8	S_Seg_1(7:0)								S_Seg_1 <sup>*)</sup>	8
0xa9	-	S_Seg_2(6:0)							S_Seg_2 <sup>*)</sup>	8
0xaa	-	S_SJW(6:0)							S_SJW <sup>*)</sup>	8
0xab	S_PRESC(7:0)								S_PRESC <sup>*)</sup>	8
0xac	-			F_Seg_1(4:0)					F_Seg_1 <sup>*)</sup>	8
0xad	-				F_Seg_2(3:0)				F_Seg_2 <sup>*)</sup>	8
0xae	-				F_SJW(3:0)				F_SJW <sup>*)</sup>	8
0xaf	F_PRESC(7:0)								F_PRESC <sup>*)</sup>	8
0xb0	KOER(2:0)			ALC(4:0)					EALCAP	8
0xb1	TDCE <sub>N</sub>	SSPOFF(4:0)							TDC <sup>*)</sup>	8

<sup>\*)</sup> регистр может быть записан только в том случае, если установлен сброс бита в регистре CFG\_STAT.

	Позиция бита								Название регистра	Размер, бит
0xb2	RECNT								RECNT	8
0xb3	TECNT								TECNT	8
0xb4	-	SELMAS K	-	ACFADR					ACFCTRL	8
0xb5	-						TIMEPOS	TIMEEN	TIMECFG	8
0xb6	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1	AE_0	ACF_EN_0	8
0xb7	AE_1 5	AE_1 4	AE_1 3	AE_1 2	AE_1 1	AE_1 0	AE_9	AE_8	ACF_EN_1	8
0xb8	ACODE_x или AMASK_x(7:0)								ACF_0 <sup>*)</sup>	8
0xb9	ACODE_x или AMASK_x(15:8)								ACF_1 <sup>*)</sup>	8
0xba	ACODE_x или AMASK_x(23:16)								ACF_2 <sup>*)</sup>	8
0xbb	-	AIDE E	AIDE	ACODE_x или AMASK_x(28:24)				ACF_3 <sup>*)</sup>	8	
0xbc	VERSION(7:0)								VER_0	8
0xbd	VERSION(15:8)								VER_1	8
0xbe	TBE	TBF	TBPTR(5:0)					TBSLOT	8	
0xbf	WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC(1:0)		TTE N	TTCFG	8
0xc0	REF_ID(7:0)								REF_MSG_0	8
0xc1	REF_ID(15:8)								REF_MSG_1	8
0xc2	REF_ID(23:16)								REF_MSG_2	8
0xc3	REF_ID E	-	REF_ID(28:24)					REF_MSG_3	8	
0xc4	-		TTPTR(5:0)					TRIG_CFG_0	8	
0xc5	TEW(3:0)				-	TTYPE(2:0)			TRIG_CFG_1	8
0xc6	TT_TRIG(7:0)								TT_TRIG_0	8
0xc7	TT_TRIG(15:8)								TT_TRIG_1	8
0xc8	TT_WTRIG(7:0)								TT_WTRIG_0	8
0xc9	TT_WTRIG(15:8)								TT_WTRIG_1	8
0xca	-			MAEIF	MDEIF	MDWIF	MDWIE	MPEN	MEM_PROT	8
0xcb	-			HELOC(1:0)		TXB	TXS	ACFA	MEM_STAT	8
0xcc	MEAE E	ME1E E	MEBP1(5:0)					MEM_ES_0	8	
0xcd	-	ME2E E	MEBP2(5:0)					MEM_ES_1	8	
0xce	MENEC(3:0)				MEEEC(3:0)				MEM_ES_2	8
0xcf	-					MES	MEL(1:0)		MEM_ES_3	8
0xd0	-	FSTIM(2:0)			SWIF	SWIE	SEIF	XMRE N	SCFG	8
0x100	Регистр управления счётчиком Time-stamp. 0x0, RW.								CAN_TSCCR	32
0x104	Регистр установки счётчика Time-stamp. 0x0, WO.								CAN_TSCSR	32
0x108	Регистр значения счётчика Time-stamp. 0x0, RO.								CAN_TSCDR	32

Доступ на запись к адресуемому местоположению, не показанному на карте регистра (Таблица 15.3), не приводит ни к какому действию, а доступ на чтение приведет к значению 0x00.

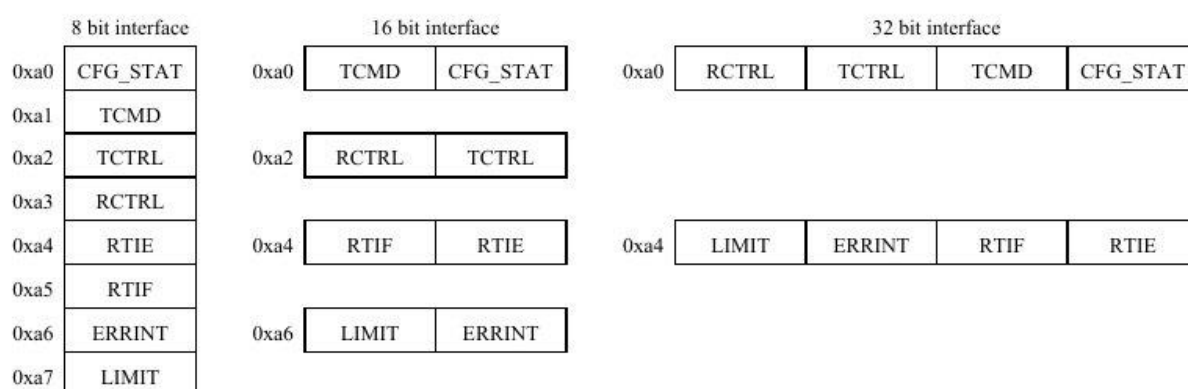
Адреса регистров приведены в качестве примера. Они могут быть выбраны с использованием общих параметров перед синтезом. В примере показана настройка по умолчанию. Пример для 8 / 16 / 32 хосты ниже на Рисунок 15.6 предполагают, что каждый байт может быть адресован, даже если хост использует более широкую собственную ширину.

Пожалуйста, обратите внимание на несколько пробелов внутри карты регистра. Это делается для лучшего выравнивания адресных сегментов для более широкого интерфейса хост-контроллера.

Пожалуйста, имейте в виду, что регистры конфигурации, предназначенные только для CAN FD, не могут использоваться, если IP-ядро ограничено только возможностями CAN 2.0B. Затем эти регистры привязываются к их сброшенным значениям.

Чтобы сбросить флаг прерывания, хост-контроллер должен записать 1 в этот флаг. Запись 0 не имеет никакого эффекта. Если новое событие прерывания происходит во время активного доступа на запись, то это событие установит флаг и переопределит сброс. Это гарантирует, что ни одно событие прерывания не будет потеряно.

Флаги прерывания будут установлены только в том случае, если установлен соответствующий бит разрешения прерывания.



**Рисунок 15.6. Карта памяти для битового интерфейса 8/ 16/ 32 (небольшой пример)**

Таблица 15.4. Отметка времени передачи TTS (от 0x98 до 0x9f)

Биты	Обозначение	Доступ	Назначение
63:0	TTS	r-0	<p>Метка времени передачи (Transmission Time Stamp (TTS)). TTS хранит метку времени последнего переданного кадра для метки времени Cja 603. Каждый новый кадр перезаписывает TTS, если TTSEN=1.</p> <p>В зависимости от общего параметра (глава "Параметры конфигурации") метка времени может иметь ширину 32 или 64.</p> <p>Неиспользуемые биты принудительно устанавливаются в 0.</p> <p>TTS предназначен для использования мастером времени для получения метки времени сообщения SYNC.</p>
7	RESET	rw-1	<p>Бит запроса на сброс:</p> <p>1 - Хост-контроллер выполняет локальный сброс CAN-CTRL, 0 - нет локального сброса CAN-CTRL.</p> <p>Некоторые регистры (например, для конфигурации узла) могут быть изменены только в том случае, если RESET=1.</p> <p>RESET битов приводит несколько компонентов в состояние сброса. Подробное определение дано в главе 15.2.11.11 "Программный сброс"</p> <p>RESET устанавливается автоматически, если узел переходит в состояние "шина выключена" (глава 15.2.11.7 "Обработка ошибок").</p> <p>Обратите внимание, что узел CAN будет участвовать в обмене данными CAN после переключения RESET на 0 после 11 бит CAN.</p> <p>Эта задержка требуется стандартом CAN (время простоя шины).</p> <p>Если RESET установлен в 1 и сразу же установлен в 0, то требуется некоторое время, пока RESET не будет считан как 0 и не станет неактивным.</p> <p>Причина в том, что частотный домен переходит от хоста к частотному домену CAN.</p> <p>RESET остается активным до тех пор, пока это необходимо, в зависимости от соотношения между часами хоста и CAN.</p>
6	LBME	rw-0	<p>Режим обратной связи, внешний (глава 15.2.11.10.4 "Режим обратной связи (LBMI или LBME)"): 0 - отключен, 1 - включен.</p> <p>LBME не должен быть включен во время активной передачи.</p> <p>Режим обратной связи, внутренний (глава 15.2.11.10.4 "Режим обратной связи (LBMI или LBME)").</p>
5	LBMI	rw-0	<p>Режим обратной связи, внутренний (глава 15.2.11.10.4 "Режим обратной связи (LBMI или LBME)"): 0 - отключен, 1 - включен.</p> <p>LBMI не должен быть включен, пока активна передача.</p>
4	TPSS	rw-0	<p>LBMI не должен быть включен, пока активна передача.</p> <p>Основной режим одиночного выстрела для РТВ (глава 15.2.11.10.1 "Однократная передача"): 0 - отключен, 1 - включен.</p>

Биты	Обозначение	Доступ	Назначение
3	TSSS	rw-0	Режим передачи вторичного одиночного выстрела для STB (глава 15.2.11.10.1 "Однократная передача"): 0 - отключен, 1 - включен.
2	RACTIVE	r-0	Прием АКТИВЕН (бит состояния приема) 1 - Контроллер в данный момент принимает кадр, 0 - Нет активности приема.
1	TACTIVE	r-0	Передача АКТИВНА (бит состояния передачи) 1 - Контроллер в данный момент передает кадр. 0 - Нет активности передачи.
0	BUSOFF	rw-0	Шина выключена (бит состояния шины, глава 15.2.11.7 "Обработка ошибок") 1 - Состояние контроллера - "шина выключена". 0 - Состояние контроллера - "шина включена". Запись 1 в BUS OFF приведет к сбросу данных TECNT и RECNT. Это должно быть сделано только для отладки. Подробности см. в главе 15.2.11.10.6 "Сброс счетчика ошибок".

Таблица 15.5. Регистр команд TCMD (0ха1)

Биты	Обозначение	Доступ	Назначение
7	TBSEL	rw-0	Выбор буфера передачи (Transmit Buffer Select). Выбирает буфер передачи, который будет загружен сообщением. Используйте регистры TBUF для доступа. TBSEL должен быть стабильным все время записи регистров TBUF и при установке ТЕКСТА. 0 - РТВ (высокоприоритетный буфер), 1 - STB. Бит будет сброшен до значения аппаратного сброса, если (TTEN=1 и TTTBM=1).
6	LOM	rw-0	Режим только для прослушивания (глава 15.2.11.10.2 "Режим только для прослушивания (LOM)). 0 - отключен, 1 - включен. LOM нельзя установить, если установлены TPE, TSONE или TSALL. Передача не может быть начата, если LOM включен, а LBME отключен. LOM=1 и LBME=0 отключают все передачи. LOM=1 и LBME=1 отключают АСК для принятых кадров и кадров ошибок, но разрешают передачу собственных кадров.
5	STBY	rw-0	Режим ожидания приемопередатчика (глава 15.2.11.10.5 "Режим ожидания приемопередатчика"). 0 - отключен; 1 - включен. Этот бит регистра связан с выходным сигналом stby, который может использоваться для управления режимом приемопередатчика. STBY не может быть установлен в 1, если TPE=1, TSONE=1 или TSALL=1. Если хост устанавливает STBY на 0, то хосту необходимо выждать период, необходимый приемопередатчику для запуска, прежде чем хост запросит новую передачу.



Биты	Обозначение	Доступ	Назначение
4	TPE	rw-0	<p>Включение первичной передачи (Transmit Primary Enable).  1 - Разрешить передачу сообщения в высокоприоритетный РТВ,  0 - Нет передачи для РТВ.  Если TPE установлен, сообщение от РТВ будет передано в следующей возможной позиции передачи. Начатая передача от STB будет завершена раньше, но ожидающие новые сообщения задерживаются до тех пор, пока сообщение РТВ не будет передано. TPE остается установленным до тех пор, пока сообщение не будет успешно передано или не будет прервано с помощью TRA.  Хост-контроллер может установить TPE на 1, но не может сбросить его на 0. Это возможно только с использованием TRA и прерыванием сообщения. Бит будет сброшен на значение аппаратного сброса, если RESET=1, STBY=1, (LOM=1 и LBME=0) или (TTEN=1 и TTTBM=1).</p>
3	TRA	rw-0	<p>Первичное прерывание передачи (Transmit Primary Abort).  1 – прерывает передачу от РТВ, которая была запрошена TPE=1, но еще не началась. (Байты данных сообщения остаются в РТВ.),  0 – без прерывания.  Бит должен быть установлен хост-контроллером и будет сброшен CAN-CTRL. Установка TRA автоматически снимает TPE.  Хост-контроллер может установить TRA в 1, но не может сбросить его в 0.  В течение короткого времени, пока ядро CAN-CTRL сбрасывает бит, он не может быть установлен хостом.  Бит будет сброшен на значение аппаратного сброса, если RESET=1 или (TTEN=1 и TTTBM=1).  TRA не следует устанавливать одновременно с TPE.</p>
2	TSONE	rw-0	<p>Вторичный ОДИН кадр передачи (Transmit Secondary ONE frame).  1 – Разрешение передачи одного в STB. В режиме FIFO это самое старое сообщение, а в режиме приоритета это сообщение с наивысшим приоритетом. TSONE в приоритетном режиме трудно обрабатывать, потому что не всегда ясно, какое сообщение будет передано, если тем временем в STB будут записываться новые сообщения. Контроллер начинает передачу, как только шина освобождается и не поступает запрос РТВ (бит TPE).  0 – Нет передачи для STB.  TSONE остается установленным до тех пор, пока сообщение не будет успешно передано или не будет прервано с помощью TSA.  Хост-контроллер может установить TSONE в 1, но не может сбросить его в 0. Это возможно только при использовании TSA и прерывании сообщения.  Бит будет сброшен на значение аппаратного сброса, если RESET=1, STBY=1, (LOM=1 и LBME=0) или (TTEN=1 и TTTBM=1).</p>



Биты	Обозначение	Доступ	Назначение
1	TSALL	rw-0	<p>Вторичные BCE кадры передачи (Transmit Secondary ALL frames).</p> <p>1 – Разрешена передача всех сообщений в STB. Контроллер начинает передачу, как только шина освобождается и не поступает запрос PTB (бит TPE).</p> <p>0 – Нет передачи для STB.</p> <p>TSALL остается установленным до тех пор, пока все сообщения не будут успешно переданы или не будут прерваны с помощью TSA.</p> <p>Хост-контроллер может установить TSALL в 1, но не может сбросить его в 0. Это возможно только при использовании TSA и прерывании сообщений. Бит будет сброшен на значение аппаратного сброса, если RESET=1, STBY=1, (LOM=1 и LBME=0) или (TTEN=1 и TTTBM=1). Если во время передачи STB загружается новым кадром, то новый кадр также будет передан. Другими словами: передача, инициированная TSALL, завершается, когда STB становится пустым.</p>
0	TSA	rw-0	<p>Вторичное прерывание передачи (Transmit Secondary Abort).</p> <p>1 – Прерывает передачу от STB, которая была запрошена, но еще не начата. Для передачи TSONE прерывается только один кадр, а для передачи TSALL прерываются все кадры. Будут освобождены один или все слоты сообщений, которые обновят TSSTAT. Все прерванные сообщения теряются, потому что они больше не доступны. Если в приоритетном режиме передача TSONE прерывается, то неясно, какой кадр будет прерван, если тем временем в STB будут записываться новые кадры.</p> <p>0 – без прерывания.</p> <p>Бит должен быть установлен хост-контроллером и будет сброшен CAN-CTRL. Установка TSA автоматически снимает TSONE или TSALL соответственно. Хост-контроллер может установить TSA в 1, но не может сбросить его в 0. Бит будет сброшен на значение аппаратного сброса, если RESET=1. TSA не следует устанавливать одновременно с TSONE или TSALL.</p>

Установка и TSONE, и TSALL бессмысленна. Пока TSALL уже установлен, нельзя установить TSONE и наоборот.

Если и TSONE, и TSALL установлены одновременно, то TSALL выигрывает, а TSONE очищается ядром CAN-CTRL.

**Таблица 15.6. Регистр управления передачей TCTRL (0xa2)**

Биты	Обозначение	Доступ	Назначение
7	FD_ISO	rw-1	<p>Режим CAN FD ISO:</p> <p>0 - режим Bosch CAN FD (не ISO),</p> <p>1 - режим ISO CAN FD (ISO 11898-1:2015).</p> <p>Режим ISO CAN FD имеет другое значение инициализации CRC и дополнительное количество битов.</p> <p>Оба режима несовместимы и не должны смешиваться в одной сети CAN.</p> <p>Этот бит не оказывает никакого влияния на CAN 2.0B.</p> <p>Этот бит доступен для записи только в том случае, если RESET=1.</p>

Биты	Обозначение	Доступ	Назначение
6	TSNEXT	rw-0	<p>Вторичный буфер передачи NEXT:</p> <p>0 - никаких действий</p> <p>1 - слот STB заполнен, выберите следующий слот.</p> <p>После того, как все байты кадра будут записаны в регистры TBUF, хост-контроллер должен установить TSNEXT, чтобы сигнализировать о том, что этот слот заполнен. Затем ядро CAN-CTRL подключает регистры TBUF к следующему слоту. Как только слот помечен как заполненный, можно начать передачу, используя TSONE или TSALL.</p> <p>Можно одновременно установить TSNEXT и TSONE или TSALL в одном доступе для записи.</p> <p>TSNEXT должен быть установлен хост-контроллером и автоматически сбрасывается ядром CAN-CTRL сразу после его установки.</p> <p>Установка TSNEXT не имеет смысла, если TBSEL=0. В этом случае TSNEXT игнорируется и автоматически очищается. Это не наносит никакого вреда. Если все слоты STB заняты, TSNEXT остается установленным до тех пор, пока слот не освободится (глава 15.2.11.6 "Полный вторичный буфер передачи (A Full STB)").</p> <p>TSNEXT не имеет значения в режиме TTCAN и имеет фиксированное значение 0.</p>
5	TSMODE	rw-0	<p>TSNEXT не имеет значения в режиме TTCAN и имеет фиксированное значение 0.</p> <p>Вторичный буфер передачи. Режим работы:</p> <p>0 - режим FIFO,</p> <p>1 - режим приоритетного решения.</p> <p>В режиме FIFO кадры передаются в том порядке, в котором они были записаны в STB. В режиме определения приоритета кадр с наивысшим приоритетом в STB автоматически передается первым. Идентификатор кадра используется для определения приоритета. Более низкий идентификатор означает более высокий приоритет кадра. Кадр в PTB всегда имеет наивысший приоритет независимо от идентификатора.</p> <p>TSMODE должен переключаться только в том случае, если STB пуст.</p>
4	TTTBM	rw-1	<p>Режим буфера передачи TTCAN.</p> <p>Если TTEN=0, то TTTBM игнорируется, в противном случае действует следующее:</p> <p>0 - отдельные PTB и STB, поведение определяется TSMODE,</p> <p>1 - полная поддержка TTCAN: слоты буфера выбираются с помощью TBPTR и TTPTR.</p> <p>Для управляемой событиями связи CAN (TTEN=0) система предоставляет PTB и STB, а поведение STB определяется TSMODE. Затем TTTBM игнорируется. Для связи CAN с запуском по времени (TTEN=1) с полной поддержкой всех функций, включая передачу с запуском по времени, необходимо выбрать TTTBM=1. Затем все слоты TB адресуются с помощью TTPTR и TBPTR.</p> <p>Для связи синхронизируемого по времени CAN (TTEN=1) с поддержкой только меток времени приема можно выбрать TTTBM=0. Тогда буфер передачи работает как в режиме, управляемом событиями, и поведение может быть выбрано TSMODE. TTTBM должен переключаться, только если TBUF пуст.</p>
3:2	-	r-0	Зарезервировано

Биты	Обозначение	Доступ	Назначение
1:0	TSSTAT	r-0	<p>Биты вторичного состояния передачи (Transmission Secondary STATus bits).</p> <p>Если ТОГДА=0 или TTTBM=0:</p> <p>00 – STB пуст,  01 – STB заполнен меньше или равен половине,  10 – STB заполнен более чем наполовину,  11 – STB заполнен.</p> <p>Если STB отключен с помощью STB_DISABLE, то TSSTAT=00.</p> <p>Если TTEN=1 и TTTBM=1:</p> <p>00 – PTV и STB пусты,  01 – PTV и STB не пустые и не полные,  11 – PTV и STB заполнены.</p>

**Таблица 15.7. Регистр управления приемом RCTRL (0xa3)**

Биты	Обозначение	Доступ	Назначение
7	SACK	rw-0	Самоподтверждение: 0 – нет self-ACK, 1 - self-ACK при LBME=1.
6	ROM	rw-0	Режим переполнения буфера приема. В случае полного RBUF при получении нового сообщения ROM выбирает следующее: 1 – новое сообщение не будет сохранено. 0 – самое старое сообщение будет перезаписано.
5	ROV	r-0	Переполнение буфера приема: 1 - Переполнение. Как минимум одно сообщение потеряно. 0 – Нет переполнения. ROV очищается установкой RREL=1.
4	RREL	rw-0	Освобождение буфера приема: Хост-контроллер прочитал фактический слот RB и освободил его. После этого ядро CAN-CTRL указывает на следующий слот RB. RSTAT обновляется. 1 — Освобождение: Хост прочитал RB, 0 – Нет освобождения.
3	RBALL	rw-0	Буфер приема хранит ВСЕ кадры данных: 0 – нормальная работа, 1 – RB хранит правильные кадры данных, а также кадры данных с ошибками (глава 15.2.11.9.4 "Прием всех кадров данных (RBALL)").
2	-	r-0	Зарезервировано
1:0	RSTAT	r-0	Буфер приема состояния (Receive buffer STATus): 00 - пустой; 01 -> пустой и < почти полный (AFWL), 10 - ≥ почти полный (порог программируется AFWL), но не полный и нет переполнения, 11 - полный (остается установленным в случае переполнения – для сигнализации переполнения см. ROV).

**Таблица 15.8. Регистр разрешения прерываний приема и передачи RTIE (0xa4)**

Биты	Обозначение	Доступ	Назначение
7	RIE	rw-1	Разрешение прерывания приема: 0 — отключено, 1 — включено.
6	ROIE	rw-1	Разрешения прерывания переполнения RB: 0 — отключено, 1 — включено.
5	RFIE	rw-1	Разрешения полного прерывания RB: 0 — отключено, 1 — включено.
4	RAFIE	rw-1	Почти полное разрешение прерывания RB: 0 — отключено, 1 — включено.
3	TPIE	rw-1	Разрешение первичного прерывания передачи: 0 — отключено, 1 — включено.
2	TSIE	rw-1	Разрешение вторичного прерывания передачи: 0 — отключено, 1 — включено.
1	EIE	rw-1	Разрешение прерывания ошибки: 0 — отключено, 1 — включено.
0	TSFF	r-0	Если TTEN=0 или TTTBM=0: Флаг заполнения вторичного буфера передачи: 1 - STB заполнен максимальным количеством сообщений. 0 - STB не заполнена максимальным количеством сообщений.  Если STB отключен с помощью STB_DISABLE, то TSFF=0. Если TTEN=1 и TTTBM=1: Флаг заполнения слота буфера передачи: 1 — слот буфера, выбранный TVPTR, заполнен. 0 — слот буфера, выбранный TVPTR, пуст.

**Таблица 15.9. Регистр флага прерывания приема и передачи RTIF (0xa5)**

Биты	Обозначение	Доступ	Назначение
7	RIF	rw-0	Флаг прерывания приема: 1 - Данные или кадр удаленного запроса получены и доступны в буфере приема. 0 - Кадр не получен.
6	ROIF	rw-0	Флаг прерывания переполнения RB: 1 - По крайней мере одно полученное сообщение было перезаписано в RB. 0 - RB не перезаписан. В случае переполнения будут установлены как ROIF, так и RFIF.
5	RFIF	rw-0	Флаг полного прерывания RB: 1 - Все RBs заполнены. Если RB не будет выпущен до тех пор, пока не будет получено следующее подлинное сообщение, то самое старое сообщение будет потеряно. 0 - FIFO RB не заполнен.

Биты	Обозначение	Доступ	Назначение
4	RAFIF	rw-0	Флаг почти полного прерывания RB: 1 - количество заполненных слотов RB $\geq$ AFWL <sub>i</sub> , 0 - количество заполненных слотов RB $<$ AFWL <sub>i</sub> .
3	TPIF	rw-0	Флаг первичного прерывания передачи: 1 - Запрошенная передача PTB была успешно завершена, 0 - Передача PTB не завершена. В режиме сканирования TPLF никогда не будет установлен. Тогда только в том случае, если это действительно так.
2	TSIF	rw-0	Флаг вторичного прерывания передачи: 1 - Запрошенная передача STB была успешно завершена. 0 - Передача STB не была успешно завершена. В режиме TTCAN TSIF будет сигнализировать обо всех успешных передачах, независимо от места хранения сообщения.
1	EIF	rw-0	Флаг прерывания ошибки: 1 - Граница предела предупреждения об ошибке была пересечена в любом направлении, или бит BUSOFF был изменен в любом направлении. 0 - Изменений не произошло.
0	AIF	rw-0	Флаг прерывания преждевременного прекращения: 1 - После установки TPA или TSA соответствующие сообщения были преждевременно прекращены. Рекомендуется не устанавливать оба TPA и TSA одновременно, поскольку оба источника AIF. 0 - Преждевременное прекращение не выполнялось.  AIF не имеет связанного регистра включения. Дополнительную информацию см. также в главе 15.2.11.5 "Прерывание передачи сообщения".

Таблица 15.10. Регистр флага ERRINT и включение прерывания ошибки (0xа6)

Биты	Обозначение	Доступ	Назначение
7	EWARN	r-0	Достигнут предел предупреждения об ошибке: 1 - Один из счетчиков ошибок, RECNT или TECNT, равен или больше, чем EWL; 0 - Значения в обоих счетчиках меньше EWL.
6	EPASS	r-0	Активный режим пассивной ошибки: 0 - не активен (узел активен по ошибке); 1 - активный (узел является пассивным по ошибке).
5	EPIE	rw-0	Включение прерывания пассивной ошибки.
4	EPIF	rw-0	Флаг прерывания пассивной ошибки (EPIF). EPIF будет активирован, если статус ошибки изменится с активной ошибки на пассивную ошибку или наоборот, и если это прерывание включено.
3	ALIE	rw-0	Включение прерывания потери арбитража
2	ALIF	rw-0	Флаг прерывания потери арбитража
1	BEIE	rw-0	Включение прерывания ошибки шины
0	BEIF	rw-0	Флаг прерывания ошибки шины

Таблица 15.11. Регистр ограничений предупреждений (LIMIT)(0ха7)

Биты	Обозначение	Доступ	Назначение
7:4	AFWL(3:0)	rw-0x1	<p>Ограничение предупреждения почти полного буфера приёма. (receive buffer Almost Full Warning Limit). AFWL определяет внутреннее ограничение предупреждения AFWL<sub>i</sub>, где <math>n_{RB}</math> - это количество доступных слотов RB.</p> $AFWL_i = \begin{cases} AFWL &   \quad n_{RB} < 16 \\ 2 \cdot AFWL &   \quad 16 \leq n_{RB} < 32 \\ 4 \cdot AFWL &   \quad 32 \leq n_{RB} < 64 \\ \vdots &   \quad \vdots \end{cases}$ <p>AFWL<sub>i</sub> сравнивается с количеством заполненных слотов RB и запускает RAFIF, если они равны. Допустимый диапазон AFWL<sub>i</sub> = [1...<math>n_{RB}</math>].</p> <p>AFWL = 0 не имеет смысла и автоматически обрабатывается как 0x1.</p> <p>(Обратите внимание, что в этом правиле подразумевается AFWL, а не AFWL<sub>i</sub>.)</p> <p>AFWL<sub>i</sub> &gt; <math>n_{RB}</math> не имеет смысла и автоматически рассматривается как <math>n_{RB}</math>.</p> <p>AFWL<sub>i</sub> = <math>n_{RB}</math> является допустимым значением, но обратите внимание, что RFIF также существует.</p>
3:0	EWL(3:0)	rw-0xB	<p>Ограничение предупреждений программируемых ошибок = (EWL+1)*8. Значения возможных ограничений: 8, 16, ... 128. Значение EWL управляет EIF.</p> <p>EWL необходимо перенести с помощью CDC с хоста на частотный домен CAN.</p> <p>Во время передачи биты регистра EWL блокируются на запись для хоста на несколько кликов, пока CDC не будет завершен.</p>

Таблица 15.12. Регистр синхронизации битов S\_Seg\_1 (0ха8)

Биты	Обозначение	Доступ	Назначение
7:0	S_Seg_1(7:0)	rw-0x3	<p>Сегмент синхронизации бита 1 (медленная скорость). Точка выборки будет установлена на <math>t_{Seg\_1} = (Seg\_1 + 2) \cdot TQ</math> после начала битового времени.</p>

Таблица 15.13. Регистр синхронизации битов S\_Seg\_2 (0ха9)

Биты	Обозначение	Доступ	Назначение
7	-	r-0	Зарезервировано
6:0	S_Seg_2(6:0)	rw-0x2	<p>Сегмент синхронизации бита 2 (медленная скорость). Время <math>t_{Seg\_2} = (Seg\_2 + 1) \cdot TQ</math> после точки выборки.</p>

**Таблица 15.14. Регистр синхронизации битов S\_SJW (0xaa)**

Биты	Обозначение	Доступ	Назначение
7	-	r-0	Зарезервировано
6:0	S_SJW(6:0)	rw-0x2	Ширина скачка синхронизации (медленная скорость). Ширина скачка синхронизации $t_{sjw} = (SJW + 1) \cdot TQ$ - максимальное время для сокращения или удлинения битового времени для повторной синхронизации, где TQ - кванты времени.

**Таблица 15.15. Регистр синхронизации битов F\_Seg\_1 (0xac)**

Биты	Обозначение	Доступ	Назначение
7:5	-	r-0	Зарезервировано
4:0	F_Seg_1(4:0)	rw-0x3	Сегмент синхронизации битов 1 (быстрая скорость). Точка выборки будет установлена на $t_{seg\_1} = (Seg\_1 + 2) \cdot TQ$ после начала битового времени.

**Таблица 15.16. Регистр синхронизации битов F\_Seg\_2 (0xad)**

Биты	Обозначение	Доступ	Назначение
7:4	-	r-0	Зарезервировано
3:0	F_Seg_2(3:0)	rw-0x2	Сегмент синхронизации битов 2 (быстрая скорость). Время $t_{seg\_2} = (Seg\_2 + 1) \cdot TQ$ после точки выборки.

**Таблица 15.17. Регистр синхронизации битов F\_SJW (0xae)**

Биты	Обозначение	Доступ	Назначение
7:4	-	r-0	Зарезервировано
3:0	F_SJW(3:0)	rw-02	Ширина скачка синхронизации (быстрая скорость). Ширина скачка синхронизации $t_{sjw} = (SJW + 1) \cdot TQ$ - это максимальное время для сокращения или удлинения битового времени для повторной синхронизации, где TQ - кванты времени.

**Таблица 15.18. Регистры предварительного регистра частоты S\_PRESC (0xab) и F\_PRESC (0xaf)**

Биты	Обозначение	Доступ	Назначение
7:0	S_PRESC F_PRESC	rw-0x01	Предварительный регистр частоты (медленная и быстрая скорость). Предварительный регистр частоты делит тактовый сигнал системы, чтобы получить тактовые сигналы квантов времени $tq\_clk$ . Допустимый диапазон PREC=[0x00, 0xff] приводит к значениям делителя от 1 до 256.

**Таблица 15.19. Регистр компенсации задержки передатчика TDC (0xb1)**

Биты	Обозначение	Доступ	Назначение
7	TDCEN	rw-0	Включение компенсации задержки передатчика. TDC будет активирован во время фазы передачи данных кадра CAN FD, если BRS активен, если DCEN=1. Более подробную информацию о BMT см. в главе 15.5.5 "Переключение скорости передачи битов и точки выборки".
6:0	SSPOFF	rw-0x00	Смещение точки вторичной выборки. Задержка передатчика плюс SSPOFF определяет время вторичной точки выборки для TDC. SSP OFF задается в виде числа TQ.

Запись в S\_Seg\_1, S\_Seg\_2, S\_SJW, S\_PRESC, F\_Seg\_1, F\_Seg\_2, F\_SJW, F\_PRESC и TDC возможна только в случае, если RESET=1. Подробное описание битовой синхронизации шины CAN приведено в главе "Псевдодвухпортовая память". Значение сброса устанавливает битовую синхронизацию, которая описана в примере в главе 15.2.10 "Флаги прерывания".

Все параметры синхронизации заданы для медленной (префикс "S\_") и быстрой скорости (префикс "F\_"). Медленная скорость используется для фазы арбитража CAN 2.0 и CAN FD. Быстрая скорость используется для фазы передачи данных CAN FD.

**Таблица 15.20. Регистр захвата потерь арбитража и ошибок EALCAP**

Биты	Обозначение	Доступ	Назначение
7:5	KOER(2:0)	r-0x0	Тип ошибки (Код ошибки): 000 - нет ошибок, 001 - битовая ошибка (BIT ERROR), 010 - ошибка в форме (FORM ERROR), 011 - ошибка в материале (STUFF ERROR), 100 - ошибка подтверждения (ACKNOWLEDGEMENT ERROR), 101 - ошибка CRC (CRC ERROR), 110 - другие ошибки (OTHER ERROR). (доминирующие биты после собственного флага ошибки, полученный активный флаг ошибки слишком длинный, доминирующий бит во время пассивного флага ошибки после ошибки ACK). 111 - не используется. KOER обновляется с каждой новой ошибкой. Поэтому он остается нетронутым, когда кадры успешно передаются или принимаются.
4:0	ALC(4:0)	r-0x0	Захват потери арбитража (позиция бита в кадре, где арбитраж был потерян).



**Таблица 15.21. Регистры счетчика ошибок RECNT (0xb2) и TECNT (0xb3)**

Биты	Обозначение	Доступ	Назначение
7:0	RECNT	r-0x00	Количество ошибок приема (количество ошибок во время приема). RECNT увеличивается и уменьшается, как определено в спецификации CAN. RECNT не переполняется. Более подробную информацию о RECNT и состоянии “шина выключена” смотрите в главе 15.2.11.7 "Обработка ошибок". Если TXB=1, то счетчики ошибок замораживаются.
7:0	TECNT	r-0x00	Количество ошибок передачи (количество ошибок во время передачи). TECNT увеличивается и уменьшается, как определено в спецификации CAN. В случае “выключенного состояния шины” TECNT может переполниться. Более подробную информацию о TECNT и состоянии “шина выключена” смотрите в главе 15.2.11.7 "Обработка ошибок". Если TXB=1, то счетчики ошибок замораживаются.

**Таблица 15.22. Регистр управления приемным фильтром ACFCTRL (0xb4)**

Биты	Обозначение	Доступ	Назначение
7:6	-	r-0	Зарезервировано
5	SELMASK	rw-0	Выберите MASKY приема: 0 - Регистры ACF_x указывают на код приема, 1 - Регистры ACF_x указывают на маску приема. ACFADR выбирает один конкретный фильтр приема. (См. Рисунок 15.7 "Доступ к фильтрам приема")
4	-	r-0	Зарезервировано
3:0	ACFADR	rw-0	Адрес фильтра приема. ACFADR указывает на определенный фильтр приема. Выбранный фильтр доступен с помощью регистров ACF_x. SELMASK бита выбирает между кодом приема и маской для выбранного фильтра приема. (См. Рисунок 15.7 "Доступ к фильтрам приема") Значение ACFADR>ACF_NUMBER-1 не имеет смысла и автоматически обрабатывается как значение ACF_NUMBER-1. (См. главу "Параметры конфигурации" и Рисунок 15.7 "Доступ к фильтрам приема" для получения подробной информации об ACF_NUMBER.)

Регистры фильтра приема ACF\_x предоставляют доступ к кодам фильтра приема ACODE\_x и маскам фильтра принятия AMASK\_x в зависимости от настройки SELMASK. (См. Рисунок 15.7, а также Таблица 15.3). Доступ на запись к ACF\_x возможен только в том случае, если RESET=1. Если выбрана псевдодвухпортовая память, то доступ на чтение также возможен только при RESET=1.

Фильтры приема создаются с использованием истинной 32-разрядной памяти, и поэтому доступ на запись должен выполняться как 32-разрядная запись.

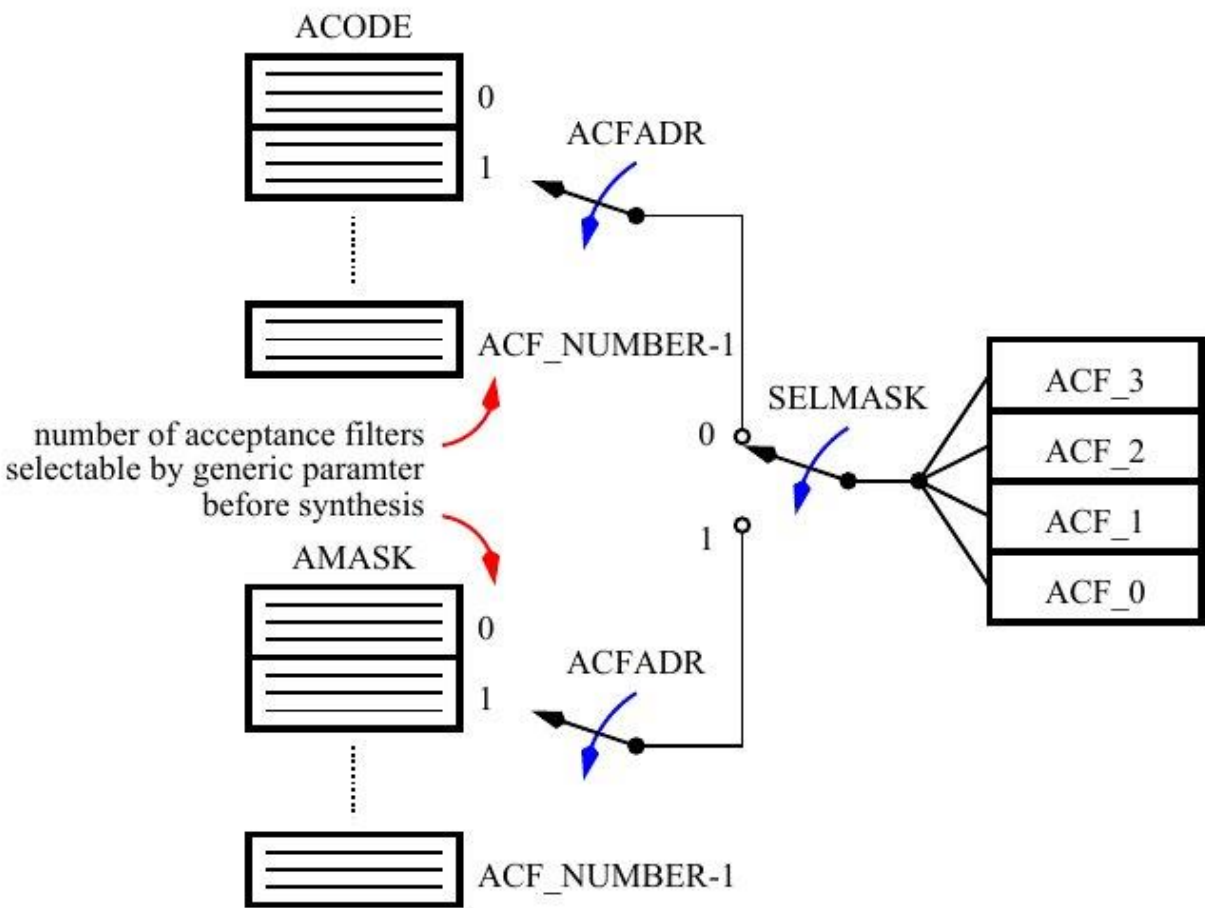


Рисунок 15.7. Доступ к фильтрам приема

Таблица 15.23. Код приема ACODE\_x (регистр ACF\_x (0xb8 to 0xbb))

Биты	Обозначение	Доступ	Назначение
7:0	ACODE_0 ACODE_x	rw-0x00 rw-u	Код приема: 1 - Значение бита ACC для сравнения с битом идентификатора полученного сообщения, 0 - значение бита ACC для сравнения с битом идентификатора полученного сообщения. CODE_x(10:0) будет использоваться для расширенных кадров. CODE_x(28:0) будет использоваться для расширенных кадров. Сброс при включении питания влияет только на фильтр 0. Все остальные фильтры остаются неинициализированными.

**Таблица 15.24. Маска приема AMASK\_x (регистр ACF\_x (0xb8 to 0xbb))**

Биты	Обозначение	Доступ	Назначение
7:0	AMASK_0 AMASK_x	rw-0xFF rw-u	<p>Маска приема:  1 - проверка приема для этих битов идентификатора приема отключена,  0 - проверка приема для этих битов идентификатора приема включена.  AMASK_x(10:0) будет использоваться для расширенных кадров.  AMASK_x(28:0) будет использоваться для расширенных кадров.  Отключенные биты приводят к принятию сообщения.  Поэтому конфигурация по умолчанию после сброса для фильтра 0 принимает все сообщения.  Сброс при включении питания влияет только на фильтр 0. Все остальные фильтры остаются неинициализированными.</p>

MASK\_x включает дополнительные биты в регистре ACF\_3 (см. Таблица 15.25), доступ к которым возможен только в том случае, если SELF MASK=1.

Эти биты могут использоваться для приема только стандартных или расширенных кадров с выбранной настройкой CODE / UMASK или для приема обоих типов кадров.

Сброс при включении питания влияет только на фильтр приема 0, и он настроен на прием обоих типов кадров после включения питания.

**Таблица 15.25. Биты в регистре ACF\_3, если SELMASK=1**

Биты	Обозначение	Доступ	Назначение
6	AIDEE	rw-0 rw-u	<p>Включение проверки бита IDE маски приема:  1 - фильтр приема принимает либо стандартные, либо расширенные кадры, как определено AIDE;  0 - фильтр приема принимает как стандартные, так и расширенные кадры.  Сброс при включении питания влияет только на фильтр 0. Все остальные фильтры остаются неинициализированными.</p>
5	AIDE	rw-0 rw-u	<p>Значение бита IDE маски приема, если AIDEE=1, то:  1 - фильтр приема принимает только расширенные кадры,  0 - фильтр приема принимает только стандартные кадры.  Сброс при включении питания влияет только на фильтр 0. Все остальные фильтры остаются неинициализированными.</p>

**Таблица 15.26. Включение фильтра приема ACF\_EN\_0 (0xb6)**

Биты	Обозначение	Доступ	Назначение
7:0	AE_x	rw-0x01 AE_x	<p>Включение фильтра принятия: 1 - фильтр приема включен, 0 - фильтр приема отключен. Каждый фильтр приема (AMASK/ ACODE) может быть индивидуально включен или отключен. По умолчанию после аппаратного сброса включен только фильтр с номером 0. Отключенные фильтры отклоняют сообщение. Только включенные фильтры могут принимать сообщение, если соответствующая конфигурация AMASK/ ACODE соответствует.</p> <p>Чтобы принимать все сообщения, необходимо включить один фильтр x, установив AE_x=1, AMASK_x=0xff и ACODE_x=0x00.</p> <p>Это конфигурация по умолчанию после аппаратного сброса для фильтра x=0, в то время как все остальные фильтры отключены.</p>

**Таблица 15.27. Включение фильтра приема ACF\_EN\_1 (0xb7)**

Биты	Обозначение	Доступ	Назначение
7:0	AE_x	rw-0x00	<p>Включение фильтра принятия: 1 - фильтр приема включен, 0 - фильтр приема отключен. Каждый фильтр приема (AMASK / ACODE) может быть индивидуально включен или отключен. Отключенные фильтры отклоняют сообщение. Только включенные фильтры могут принимать сообщение, если соответствующая конфигурация AMASK / ACODE соответствует.</p>

**Таблица 15.28. Информация о версиях VER\_0 (0xbc) и VER\_1 (0xbd)**

Биты	Обозначение	Доступ	Назначение
15:0	VER_0 VER_1	r	<p>Версия CAN-CTRL, задается как десятичное значение. VER_1 содержит основную версию, а VER_0 - второстепенную версию. Пример: версия 5x16N00S00 представлена значениями VER_1=5 и VER_0=16.</p>

**Таблица 15.29. Временная метка CiA 603 TIMECFG (0xb5)**

Биты	Обозначение	Доступ	Назначение
7:2	-	r-0	Зарезервировано

Биты	Обозначение	Доступ	Назначение
1	TIMEPOS	rw-1	Положение с временной меткой: 0 – SOF, 1 – EOF (см. главу 15.4 "Временная метка CiA 603"). TIMEPOS могут быть изменены только при TIMEEN=0, но можно изменить TIMEPOS с помощью того же доступа на запись, который устанавливает TIMEEN=1.
0	TIMEEN	rw-0	Включение временной метки: 0 – отключено, 1 – включено.

Таблица 15.30. Регистр-указатель слота ТВ TBSLOT (0xbe)

Биты	Обозначение	Доступ	Назначение
7	TBE	rw-0	Установите для слота ТВ значение “Пустой”: 1 - слот, выбранный TVPTR, должен быть помечен как “пустой”, 0 - никаких действий. TBE автоматически сбрасывается на 0, как только слот помечается как пустой и TSFF=0. Если передача из этого слота активна, то TBE остается установленным до тех пор, пока либо передача завершается или после ошибки передачи или потери арбитража передача больше не активна. Если TBF и TBE установлены, то TBE выигрывает.
6	TBF	rw-0	Установите для слота ТВ значение “Заполненный”: 1 - слот, выбранный TVPTR, должен быть помечен как “заполненный”, 0 - никаких действий. TBF автоматически сбрасывается на 0, как только слот помечается как заполненный и TSFF=1. Если TBF и TBE установлены, то TBE выигрывает.
5:0	TVPTR	rw-0x00	Указатель на слот для сообщений ТВ: 0x00 - Указатель на PTB, другие - Указатель на слот в STB. Слот сообщения, на который указывает TVLPTR, доступен для чтения/ записи с использованием регистров TBUF. Доступ на запись возможен только в том случае, если DIFF=0. Установка значения TBF равным 1 помечает выбранный слот как заполненный, а установка значения 1 помечает выбранный слот как пустой. TBSEL и TSNEXT не используются в режиме TTCAN и не имеют никакого значения. TVPTR может указывать только на буферные слоты, которые существуют в аппаратном обеспечении. Неиспользуемые биты TVPTR фиксируются на 0. TVPTR ограничен слотами PTB и 63 STB. В режиме TTCAN нельзя использовать больше слотов. Если TVPTR слишком велик и указывает на слот, который недоступен, то TBF и TBE сбрасываются автоматически, и никаких действий не происходит.

**Таблица 15.31. TTCAN: Конфигурация временного триггера TTCFG (0xbf)**

Биты	Обозначение	Доступ	Назначение
0	TTEN	rw-0	Включение временного триггера: 1 - TTCAN включен, таймер запущен, 0 - отключен.
2:1	T_PRESC	rw-0x0	Предварительный делитель частоты таймера TTCAN: 00b - 1, 01b - 2, 10b - 4, 11b - 8. Временная база TTCAN - это битовое время CAN, определяемое S_PRESC, S_SEG_1 и S_SEG_2. С помощью T_PRESC определяется дополнительный коэффициент предварительного масштабирования, равный 1, 2, 4 или 8. T_PRESC может быть изменен только в том случае, если TTEN=0, но можно изменить T_PRESC и установить TTEN одновременно с одним доступом на запись.
3	TTIF	rw-0	Флаг прерывания временного триггера. TTIF будет установлен, если TTIE задано и время цикла равно временному триггеру TT_TRIG. Запись единицы в TTIF сбрасывает ее. Написание нуля не оказывает никакого влияния. TTIF будет установлен только один раз. Если TT_TRIG не обновляется, то TTIF не будет установлен снова в следующем базовом цикле.
4	TTIE	rw-1	Включение прерывания временного триггера. Если TTIE установлен, то TTIF будет установлен, если время цикла равно временному триггеру TT_TRIG.
5	TEIF	rw-0	Флаг прерывания ошибки триггера. Условия, при которых будет установлен TEIF, определены в главе 15.3.4 "Типы триггера TTCAN". Нет бита для включения или отключения обработки TEIF.
6	WTIF	rw-0	Флаг прерывания сторожевого триггера. WTIF будет установлен, если количество циклов достигнет предела, определенного TT_WTRIG, и если WTIE будет установлен.
7	WTIE	rw-1	Включение прерывание триггера наблюдения.

**Таблица 15.32. TTCAN: Опорное сообщение REF\_MSG\_0 to REF\_MSG\_3 (0xc0 to 0xc3)**

Биты	Обозначение	Доступ	Назначение
28:0	REF_ID	rw-0x0000	Идентификатор опорного сообщения. Если REF_IDE равен: 1 - REF_ID(28:0) действителен (расширенный ID), 0 - REF_ID(10:0) действителен (стандартный ID). REF_ID используется в режиме TTCAN для обнаружения опорного сообщения. Это справедливо как для slave-устройств времени (прием), так и для master-устройства времени (передача). Если опорное сообщение обнаружено и ошибок нет, то метка Sync_Mark этого кадра станет отметкой Ref_Mark. REF_ID(2:0) не проверяется, и поэтому соответствующие биты регистра принудительно устанавливаются в 0. Эти биты используются до 8 потенциальных master-устройств времени. CAN-CTRL распознает опорное сообщение только по идентификатору. Полезная нагрузка не тестируется. Дополнительное примечание: master-устройство времени будет передавать опорное сообщение таким же образом, как и обычный кадр. REF_ID предназначен для обнаружения успешной передачи опорного сообщения.
31	REF_IDE	rw-0	Бит IDE опорного сообщения

В таблице Таблица 15.3 дано определение битовых позиций REF\_ID и REF\_IDE внутри REF\_MSG\_0 до REF\_MSG\_2.

Доступ на запись к REF\_MSG\_3 запускает передачу данных опорного сообщения в частотный домен CAN (перекрывание диапазонов изменения тактовых частот). Во время этого автоматического переноса активна блокировка записи для всех регистров REF\_MSG\_0 в REF\_MSG\_2. Передача занимает до 6 тактовых частот в тактовом домене CAN и до 6 тактовых частот в тактовом домене хоста. Доступ на запись к REF\_MSG\_2 необходим для того, чтобы сделать новое опорное сообщение активным.

**Таблица 15.33. TTCAN: Конфигурация триггера TRIG\_CFG\_0 (0xc4)**

Биты	Обозначение	Доступ	Назначение
5:0	ТПPTR	rw-0x00	Указатель слота ТВ триггера передачи. Если ТППТР слишком велик и указывает на слот, который недоступен, то устанавливается TEIF, и никакой новый триггер не может быть активирован после доступа на запись в ТТ_TRIG_1. Если ТППТР указывает на пустой слот, то они будут установлены в тот момент, когда будет достигнуто время триггера.
7:6	-	r-00	Зарезервировано

**Таблица 15.34. TTCAN: Конфигурация триггера TRIG\_CFG\_1 (0xc5)**

Биты	Обозначение	Доступ	Назначение
2:0	TTYPE(2:0)	rw-0x0	Тип триггера: 000b - Немедленный триггер - для немедленной передачи, 001b - Временный триггер - для триггеров приема, 010b - Триггер одиночной передачи - для эксклюзивных окон времени, 011b - Триггер начала передачи - для объединенных окон времени арбитража, 100b - Триггер остановки передачи - для объединенных временных окон арбитража, другие - никаких действий, Время срабатывания определяется TT_TRIG. TTPTR выбирает слот ТВ для триггеров передачи. Более подробную информацию смотрите в главе 15.3.4 "Типы триггеров TTCAN".
3	-	r-0	Зарезервировано
7:4	TEW(3:0)	rw-0x0	Окно включения передачи. Для триггера передачи одиночного импульса существует время до 16 тиков времени цикла, в течение которого кадр может начаться. TWE+1 определяет количество циклов. Дополнительные сведения см. в главе 15.3.4.3 "Триггер одиночной передачи". TEW=0 является допустимой настройкой и сокращает окно разрешения передачи до 1 цикла.

**Таблица 15.35. TTCAN: Время триггера TT\_TRIG\_0 и TT\_TRIG\_1 (0xc6 and 0xc7)**

Биты	Обозначение	Доступ	Назначение
7:0	TT_TRIG	rw-0x00	Время триггера. TT_TRIG(15:0) определяет время цикла для триггера. Для триггера передачи самой ранней точкой передачи SOF соответствующего кадра будет TT_TRIG+1.

Доступ на запись к TT\_TRIG\_1 запускает передачу данных определения триггера в частотный домен CAN (пересечение тактового домена) и активирует триггер.

Если триггер активен, то активна блокировка записи для всех регистров TRIG\_CFG\_0, TRIG\_CFG\_1, TT\_TRIG\_0 и TT\_TRIG\_1.

Блокировка записи становится неактивной при достижении времени срабатывания (TTIF устанавливается при включении TTIE) или обнаружении ошибки (устанавливается TEIF).



**Таблица 15.36. TTCAN: Время сторожевого триггера TT\_WTRIG\_0 и TT\_WTRIG\_1 (0xc8 и 0xc9)**

Биты	Обозначение	Доступ	Назначение
7:0	TT_WTRIG	rw-0xff	Время сторожевого триггера. TT_WTRIG(15:0) определяет время цикла для сторожевого триггера. Начальным сторожевым триггером является максимальное время цикла 0xffff. Более подробную информацию смотрите в главе 15.3.5 "Сторожевой триггер TTCAN".

Доступ на запись к TT\_WTRIG\_1 запускает передачу данных определения триггера в частотный домен CAN (перекрытие диапазонов изменения тактовых частот). Во время этой автоматической передачи активна блокировка записи для регистров TT\_WTRIG\_0 и TT\_WTRIG\_1. Передача занимает до 6 тактовых частот в частотном домене CAN и до 6 тактовых частот в частотном домене хоста. Доступ на запись к TT\_WTRIG\_1 необходим для активации нового триггера.

Защита памяти, описанная в таблицах Таблица 15.37 и Таблица 15.38, доступна только в том случае, если параметр синтеза RAM\_ECC>0 (см. главу "Параметры конфигурации"). Если не поддерживается, то биты имеют фиксированные значения. Более подробную информацию об использовании и поведении можно найти в главе "Защита памяти с использованием ECC".

**Таблица 15.37. MEM\_PROT: защита памяти (0xca)**

Биты	Обозначение	Доступ	Назначение
0	MPEN	rw-1	Включение защиты памяти: 0 - отключено, 1 - включено. MPEN доступен для записи только в том случае, если RESET=1. Если параметр синтеза RAM_ECC=0, то этот бит всегда равен 0.
1	MDWIE	rw-1	Память: включение прерывания предупреждения данных. 0 - отключено, 1 - включено. Если параметр синтеза RAM_ECC=0, то этот бит всегда равен 0.
2	MDWIF	rw-0	Память: флаг прерывания предупреждения данных. MDWIF будет установлен, если MDWIE установлен и защита памяти обнаружила ошибку в данные, которые были автоматически исправлены. Система продолжает работать.
3	MDEIF	rw-0	Память: флаг прерывания ошибки данных. Значение будет установлено, если защита памяти обнаружила ошибку в данных, которая не была исправлена.

Биты	Обозначение	Доступ	Назначение
4	MAEIF	rw-0	Память: флаг прерывания ошибки адреса. MAEIF будет установлен, если защита памяти обнаружила ошибку в адресе. Ошибка не была исправлена.
7:5	-	r-00	Зарезервировано

Таблица 15.38. MEM\_STAT: состояние памяти (0xcb)

Биты	Обозначение	Доступ	Назначение
0	ACFA	rw-0	Отключение приемного фильтра: 0 – нормальная работа ACF, 1 – ACF отключен: принимаются все полученные кадры. Если MDEIF или MAEIF установлены из-за ошибки в диапазоне адресов ACF, то устанавливается ACFA. Затем фильтрация приема отключается, и все кадры будут приняты. ACFA можно сбросить аналогично флагу прерывания, записав в него значение 1. Но поскольку ACFA будет установлен, пока прием все еще активен, его необходимо сбросить, если позже прием был завершен и т.д. RIF был установлен. ACFA также сбрасывается, если RESET=1.
1	TXS	r-0	Остановка передачи: 0 - нормальная работа, 1 - передача остановлена. Если MDEIF или MAEIF установлено из-за ошибки, когда перегруппировывающий автомат обращается к памяти, то любая новая передача прекращается. Если есть активная передача, то она будет завершена до остановки, но если во время этой передачи произойдет ошибка, то повторная передача не будет запущена. TXS сбрасывается, если RESET=1.
2	TXB	r-0	Блок передачи: 0 - нормальная работа, 1 - передача заблокирована. Если значение MDEIF или MAEIF установлено из-за ошибки, когда протокольный автомат CAN считывает данные для передачи, то передача немедленно блокируется. Если установлен параметр SEIF, то передача также немедленно блокируется. Если TXB=1, то счетчики ошибок замораживаются. TXB сбрасывается, если RESET=1.
4:3	HELOC	r-00	Местоположение ошибки памяти на стороне хоста: 00 – нет ошибки при доступе со стороны хоста, 01 – ошибка при доступе со стороны хоста в TBUF, 10 – ошибка при доступе со стороны хоста в RBUF, 11 – ошибка при доступе со стороны хоста в ACF. HELOC будет обновляться с каждой новой ошибкой во время доступа для чтения со стороны хоста. Этого достаточно, потому что ошибки чтения во время доступа для чтения со стороны CAN будут сигнализироваться ALFA, TX и TXB. HELOC будет обновляться только в случае ошибки, но не в случае предупреждения, вызванного исправленной однобитовой ошибкой.
7:5	-	r-0	Зарезервировано

Для проверки защиты памяти можно выполнить стимуляцию ошибок памяти, как определено ниже в четырех таблицах Таблица 15.39, Таблица 15.40, Таблица 15.41, Таблица 15.42.

Это доступно только в том случае, если параметры синтеза RAM\_ES=1 и RAM\_ECC>0 (см. главу "Параметры конфигурации").

В противном случае эти биты фиксируются равными 0.

**Таблица 15.39. MEM\_ES\_0: стимуляция ошибок памяти 0 (0хсс)**

Биты	Обозначение	Доступ	Назначение
5:0	MEBP1	rw-00	Ошибка памяти: позиция бита 1. MEBP1 определяет позицию одобитовой ошибки во время чтения памяти.
6	ME1EE	rw-0	Ошибка памяти: включение первой ошибки. 0 - отключено. 1 - включено. Если ME1EE E=1, то MEBP1 используется для стимуляции однобитовой ошибки. Чтобы стимулировать двухбитовую ошибку, также необходимо использовать MEBP2 и ME2EE. Ошибка возникает только для доступа, определенного NEC, NEC, MEL и ME.
7	MEAEE	rw-0	Ошибка памяти: включение ошибки адреса. 0 – отключено, 1 – включено. Если MEAEE=1, то бит адреса 2 переворачивается. Результатом является доступ к соседнему 32-битовому слову. Ошибка возникает только для доступа, определенного MEEEC, MENEC, MEL и MES. Если параметр синтеза RAM_ECC ≠ 2, то этот бит всегда равен 0.

**Таблица 15.40. MEM\_ES\_1: стимуляция ошибок памяти 1 (0хcd)**

Биты	Обозначение	Доступ	Назначение
5:0	MEBP2	rw-00	Ошибка памяти: позиция бита 2. MEBP2 определяет положение одной однобитовой ошибки во время чтения из памяти. Чтобы стимулировать двухбитовую ошибку, необходимо использовать MEBP1 и ME1EE.
6	ME2EE	rw-0	Ошибка памяти: включение второй ошибки. 0 - отключено, 1 - включено. Если ME2EE E=1, то MEBP2 используется для стимуляции однобитовой ошибки. Ошибка возникает только для доступа, определенного MEEEC, MENEC, MEL и MES.
7	-	r-0	Зарезервировано

**Таблица 15.41. MEM\_ES\_2: стимуляция ошибок памяти 2 (0хсе)**

Биты	Обозначение	Доступ	Назначение
3:0	MEEEC	rw-00	Ошибка памяти: счетчик включения ошибок. Стимуляция ошибок активируется, если MENEC=0 и MEEEC>0. MEEEC автоматически уменьшается после каждого события стимуляции ошибки (определенного MEL и MES), пока оно не достигнет значения 0x0.
7:4	MENEC	rw-00	Ошибка памяти: нет счетчика ошибок. Стимуляция ошибки задерживается, если MENEC>0. MENEC автоматически уменьшается после каждого доступа к чтению памяти (определенного MEL и MES), пока не достигнет значения 0x0.

**Таблица 15.42. MEM\_ES\_3: стимуляция ошибок памяти 3 (0хcf)**

Биты	Обозначение	Доступ	Назначение
1:0	MEL	rw-00	Расположение ошибок памяти. 00 – ACF, 01 – TBUF, 10 – RBUF, 11 – ACF. MEL определяет область памяти, которая будет поражена следующей ошибкой памяти вместе с MES.
2	MES	rw-0	Сторона ошибок памяти. 0 – сторона хоста, 1 – сторона CAN. MES определяет сторону, которая будет поражена следующей ошибкой памяти вместе с MEL.
7:3	-	r-0x0	Зарезервировано

Пространственная избыточность, описанная ниже в Таблица 15.43, доступна только в том случае, если параметр синтеза SAFETY\_ENABLE=1 (см. главу "Параметры конфигурации").

Если не поддерживается, то биты имеют фиксированные значения.

**Таблица 15.43. SCFG: конфигурация безопасности (0xd0)**

Биты	Обозначение	Доступ	Назначение
0	XMREN	rw-1	Включение DMR/ TMR: 0 – отключено, 1 – включено. Это зависит от параметра синтеза SAFETY_ENABLE, доступен ли DMR или TMR. XMREN можно изменить только в том случае, если RESET=1.

Биты	Обозначение	Доступ	Назначение
1	SEIF	rw-0	DMR / TMR: флаг прерывания ошибки безопасности. 0 – нет прерывания, 1 – ожидание прерывания. Если XMEN=1 и если существует некорректируемое несоответствие между избыточными экземплярами, когда будут установлены SELF и TXB.
2	SWIE	rw-1	DMR / TMR: включение прерывания предупреждения о безопасности. 0 – SWIF отключен, 1 – SWIF включен.
3	SWIF	rw-0	DMR / TMR: флаг прерывания предупреждения о безопасности: 0 – нет прерывания, 1 – ожидание прерывания. Если XMEN=1 и если существует исправимое несоответствие между избыточными экземплярами, то SWIFT будет установлен, если SWIPE=1.
6:4	FSTIM	rw-000	Инжекция ошибок. Стимуляция неисправностей возможна только в том случае, если XMREN=1.
7	-	r-0	Зарезервировано

Таблица 15.44. Регистр CAN\_TSCCR

Биты	Обозначение	Доступ	Назначение
31:1	-		Резерв
0	EN	RW	Управление time-stamp счётчиком: 0: отключен 1: включен (инкрементируется)

Таблица 15.45. Регистр CAN\_TSCSR

Биты	Обозначение	Доступ	Назначение
31:0	NEWCNT	WO	Запись в данный регистр устанавливает значение time-stamp счётчика. При чтении возвращается ноль.

Таблица 15.46. Регистр CAN\_TSCDR

Биты	Обозначение	Доступ	Назначение
31:0	CNT	RO	Возвращает текущее значение time-stamp счётчика. Запись в данный регистр игнорируется.

Таблица 15.47. Регистры приемного буфера RBUF – Стандартный формат (r-0)

Адрес	Позиция бита								Функция
	7	6	5	4	3	2	1	0	
RBUF	ID (7:0)								Identifier
RBUF+1	-					ID(10:8)			Identifier
RBUF+2	-								Identifier
RBUF+3	ESI	-							Identifier
RBUF+4	IDE=0	RTR	FDF	BRS	DLC(3:0)				Control

Адрес	Позиция бита			Функция
RBUF+5	KOER	TX	-	Status
RBUF+6	CYCLE_TIME(7:0)			TTCAN
RBUF+7	CYCLE_TIME(15:8)			TTCAN
RBUF+8	d1(7:0)			Data byte 1
RBUF+9	d2(7:0)			Data byte 2
.	...			.
RBUF+71	d64(7:0)			Data byte 64
RBUF+72	RTS(7:0)			CiA 603
.	...			.
RBUF+79	RTS(63:56)			CiA 603

**Таблица 15.48. Регистры буфера приема RBUF – расширенный формат (r-0)**

Адрес	Позиция бита								Функция
	7	6	5	4	3	2	1	0	
RBUF	ID(7:0)								Identifier
RBUF+1	ID(15:8)								Identifier
RBUF+2	ID(23:16)								Identifier
RBUF+3	ESI	-		ID(28:24)				Identifier	
RBUF+4	IDE=1	RTR	FDF	BRS	DLC(3:0)			Control	
RBUF+5	KOER			TX	-			Status	
RBUF+6	CYCLE_TIME(7:0)								TTCAN
RBUF+7	CYCLE_TIME(15:8)								TTCAN
RBUF+8	d1(7:0)								Data byte 1
RBUF+9	d2(7:0)								Data byte 1
.	...								.
RBUF+71	d64(7:0)								Data byte 64
RBUF+72	RTS(7:0)								CiA 603
.	...								.
RBUF+79	RTS(63:56)								CiA 603

Регистры RBUF (от 0x00 до 0x4f) указывают слот сообщения с самым старым полученным сообщением в RB, как видно ниже на рисунке Рисунок 15.8. Все регистры RBUF могут быть считаны в любом порядке.

KOER в RBUF имеет то же значение, что и биты KOER в регистре EALCAP. KOER в RBUF становится значимым, если RBALL=1 (см. главу 15.2.11.9.4 "Прием всех кадров данных (RBALL)").

Бит состояния TX в RBF устанавливается равным 1, если активирован режим обратной связи (глава 15.2.11.10.4 "Режим обратной петли (LBMI and LBME)") и ядро получило свой собственный переданный кадр. Это может быть полезно, если LBME=1 и другие узлы в сети также выполняют передачи.

Метка времени CYCLE\_TIME будет сохранена в RBUF только в режиме TITAN. Это время цикла в SOF этого кадра. Время цикла опорного сообщения всегда равно 0.

Метки времени приема (PTS) для временной метки Tia 603 сохраняются для каждого принятого сообщения в конце диапазона адресов RBUF. Поэтому, в отличие от TTS, RTS связан с фактическим выбранным слотом RBUF.

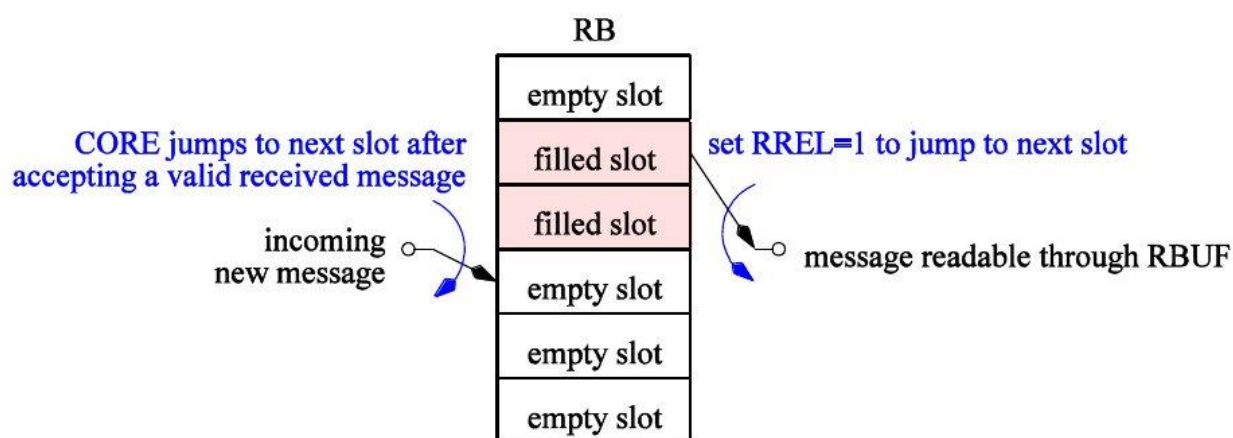


Рисунок 15.8. Схема FIFO-подобного RB (пример с 6 слотами)

Таблица 15.49. Регистры буфера передачи TBUF - стандартный формат (rw-u)

Адрес	Позиция бита								Функция
	7	6	5	4	3	2	1	0	
TBUF	ID(7:0)								Identifier
TBUF+1	-					ID(10:8)			Identifier
TBUF+2	-								Identifier
TBUF+3	TTSE N	-							Identifier
TBUF+4	IDE=0	RTR	FDF	BRS	DLC(3:0)				Control
TBUF+8	d1(7:0)								Data byte 1
TBUF+9	d2(7:0)								Data byte 2
.	...								.
TBUF+71	d64(7:0)								Data byte 64

Таблица 15.50. Регистры буфера передачи TBUF - расширенный формат (rw-u)

Адрес	Позиция бита								Функция
	7	6	5	4	3	2	1	0	
TBUF	ID(7:0)								Identifier
TBUF+1	ID(15:8)								Identifier
TBUF+2	ID(23:16)								Identifier
TBUF+3	TTSEN	-		ID(28:24)				Identifier	
TBUF+4	IDE=1	RTR	FDF	BRS	DLC(3:0)			Control	
TBUF+8	d1(7:0)								Data byte 1
TBUF+9	d2(7:0)								Data byte 2
.	...								.
TBUF+71	d64(7:0)								Data byte 64

Регистры TBUF (от 0x50 до 0x97) указывают на следующий пустой слот сообщения в STB, если TBSEL=1, или в PTB в противном случае. Более подробная информация приведена ниже на рисунке Рисунок 15.9. Все регистры TBUF могут быть записаны в любом порядке. Для STB необходимо установить TSNEXT, чтобы отметить заполненный слот и перейти к следующему слоту сообщения.

Пожалуйста, обратите внимание на разрыв в диапазоне адресации TBUF от TBUF+5 до TBUF+7. Это делается для лучшего выравнивания адресных сегментов. Ячейки памяти в промежутке могут считываться и записываться, но не имеют никакого значения для протокола CAN.

TBUF создается с использованием истинной 32-разрядной памяти, и поэтому доступ на запись должен выполняться как 32-разрядная запись.

Как RBUF, так и TBUF включают в себя некоторые управляющие биты, индивидуальные для каждого кадра (см. ниже Таблица 15.51). Для RBUF эти биты сигнализируют о состоянии соответствующих битов поля управления CAN принятого кадра CAN, в то время как для TBUF эти биты выбирают соответствующий бит поля управления CAN для кадра, который должен быть передан.

В отличие от RTS, который сохраняется для каждого принятого кадра, TTS сохраняется только для последнего переданного кадра, если TTSEN=1. TTS не связан с фактическим выбранным слотом TBUF.

**Таблица 15.51. Биты управления в RBUF и TBUF**

Бит	Описание
IDE	Расширение идентификатора: 0 – Стандартный формат: ID(10:0), 1 – Расширенный формат: ID(28:0).
RTR	Запрос на дистанционную передачу: 0 – кадр данных, 1 – кадр удаленного запроса. Дистанционными кадрами могут быть только кадры CAN 2.0. Для CAN FD нет дистанционного кадра. Поэтому RTR принудительно устанавливается в 0, если FD=1 in TBUF и REBUF. Если кадр CAN FD принимается с битом RRS=1, то это игнорируется, вместо этого ожидается полезная нагрузка данных для приема, и RTR в RBUF переопределяется, но CRC кадра вычисляется с RRS=1.
FDF	Кадр CAN FD: 0 – кадр CAN 2.0 (полезная нагрузка до 8 байт), 1 – кадр CAN FD (полезная нагрузка до 64 байт).
BRS	Переключение скорости передачи битов: 0 – номинальная/ низкая скорость передачи битов для всего кадра, 1 – переключение на передачу данных/ быструю скорость передачи битов для полезной нагрузки данных и CRC. Только кадры CAN FD могут переключать скорость передачи битов. Следовательно, BRS принудительно устанавливается в 0, если FDF=0.



Бит	Описание
ESI	<p>Индикатор состояния ошибки.</p> <p>Это бит состояния, доступный только для чтения для RBUF, и он недоступен в TBUF. Протокольный автомат машинально вставляет правильное значение ESI в передаваемые кадры.</p> <p>ESI включен только в кадры CAN FD и не существует в кадрах CAN 2.0.</p> <p>0 – узел CAN активен по ошибке, 1 – узел CAN является пассивным по ошибке.</p> <p>ESI в RBUF всегда низкий для кадров CAN 2.0.</p> <p>Состояние ошибки для передачи отображается с помощью бита EPASS в регистре ERRINT.</p>
TTSEN	<p>Включить метку времени передачи.</p> <p>Для метки времени CiA 603 получение метки времени передачи TTS может быть выбрано в TBUF:</p> <p>0 – нет получения метки времени передачи для этого кадра, 1 – обновление TTS включено.</p>

Код длины данных (DLC) в RBUF и TBUF определяет длину полезной нагрузки – количество байтов полезной нагрузки в кадре. Более подробную информацию см. ниже в таблице Таблица 15.52.

Кадры удаленного запроса (только для кадров CAN 2.0, где FDF=0) всегда передаются с 0 байтами полезной нагрузки, но содержимое DLC передается в заголовке кадра. Следовательно, можно закодировать некоторую информацию в биты DLC для дистанционных кадров. Но тогда необходимо соблюдать осторожность, если разным узлам CAN разрешено передавать дистанционный кадр с одним и тем же идентификатором. В этом случае все передатчики должны использовать один и тот же DLC, поскольку в противном случае это приведет к неразрешимой коллизии.

**Таблица 15.52. Определение DLC (в соответствии со спецификацией CAN 2.0 / FD)**

DLC (двоичный)	Тип кадра	Полезная нагрузка в байтах
0000 – 1000	CAN 2.0 и CAN FD	0 - 8
1001 – 1111	CAN 2.0	8
1001	CAN FD	12
1010	CAN FD	16
1011	CAN FD	20
1100	CAN FD	24
1101	CAN FD	32
1110	CAN FD	48
1111	CAN FD	64

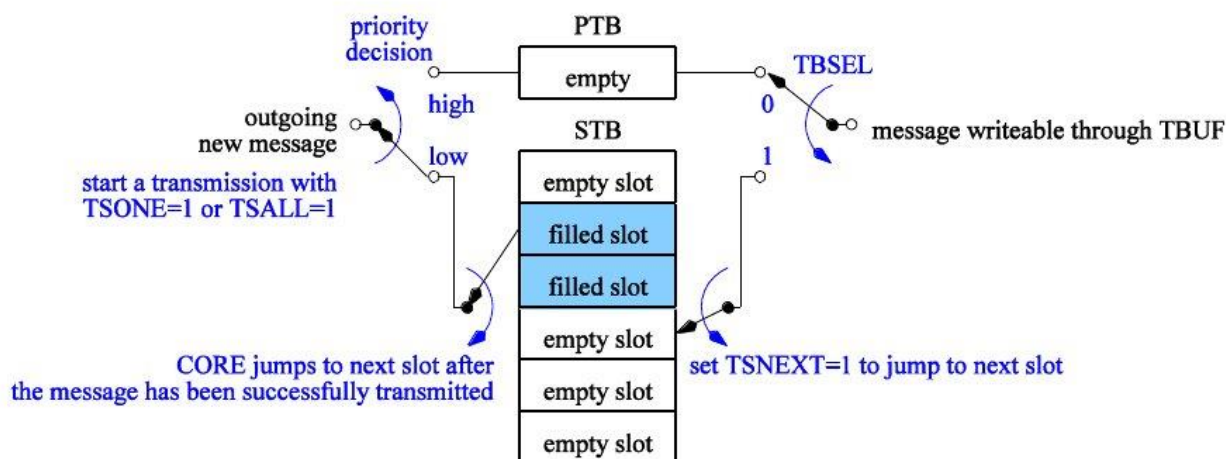


Рисунок 15.9. Схема PTB и STB в режиме FIFO (пустые PTB и 6 слотов STB)

TBUF регистрируется и доступен для чтения и записи. Поэтому хост-контроллер может использовать TBUF для последовательной побитовой подготовки сообщения, если это необходимо.

### 15.2.10 Флаги прерывания

CAN-CTRL предоставляет несколько флагов прерывания. Все они устанавливаются, если происходит соответствующее событие и соответствующий бит разрешения прерывания активен (если соответствующий бит разрешения прерывания существует). Чтобы сбросить флаг прерывания, хост-контроллер должен записать 1 в этот флаг. Запись 0 не имеет никакого эффекта.

Вывод верхнего уровня <host\_irq> устанавливается, если установлен хотя бы один из флагов прерывания. Этот сигнал предназначен для использования хост-контроллером в качестве отдельного запроса на прерывание.

В дополнение к <host\_irq> CAN-CTRL предоставляет вывод верхнего уровня <host\_if>, который представляет собой вектор, содержащий каждый отдельный флаг прерывания. Эти отдельные флаги предназначены для использования хост-системой для обработки отдельных прерываний. Ниже в Таблица 15.53 показано сопоставление флагов прерывания с <host\_if>.

Таблица 15.53. Выходной сигнал верхнего уровня <host\_if>

Позиция бита и флаги прерывания							
7	6	5	4	3	2	1	0
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
15	14	13	12	11	10	9	8
MDEIF	MDWIF	WTIF	TEIF	TTIF	EPIF	ALIF	BEIF

Позиция бита и флаги прерывания							
-	-	-	-	-	18	17	16
-	-	-	-	-	SWIF	SEIF	MAEIF

### 15.2.11 Общая работа

В этой главе описывается обработка связи CAN. Прежде чем связь станет возможной, ядро CAN-CTRL должны быть настроено в соответствии с таймингами шины CAN.

#### 15.2.11.1 Фильтры приема

Чтобы уменьшить нагрузку на принимаемый кадры для хост-контроллера, ядро использует фильтры приема. Ядро CAN-CTRL проверяет идентификатор сообщения во время фильтрации приема. Следовательно, длина каждого фильтра приема составляет 29 бит.

Если сообщение проходит один из фильтров, оно будет принято. Если оно принято, сообщение будет сохранено в RB, и, наконец, будет установлен RIF, если включен RYE. Если сообщение не принято, RIF не устанавливается и указатель RB FIFO не увеличивается. Сообщения, которые не будут приняты, будут отброшены и перезаписаны следующим сообщением. Ни одно сохраненное действительное сообщение не будет перезаписано любым не принятым сообщением.

Независимо от результата фильтрации приема, ядро CAN-CTRL проверяет каждое сообщение на шине и отправляет на шину подтверждение или кадр ошибки.

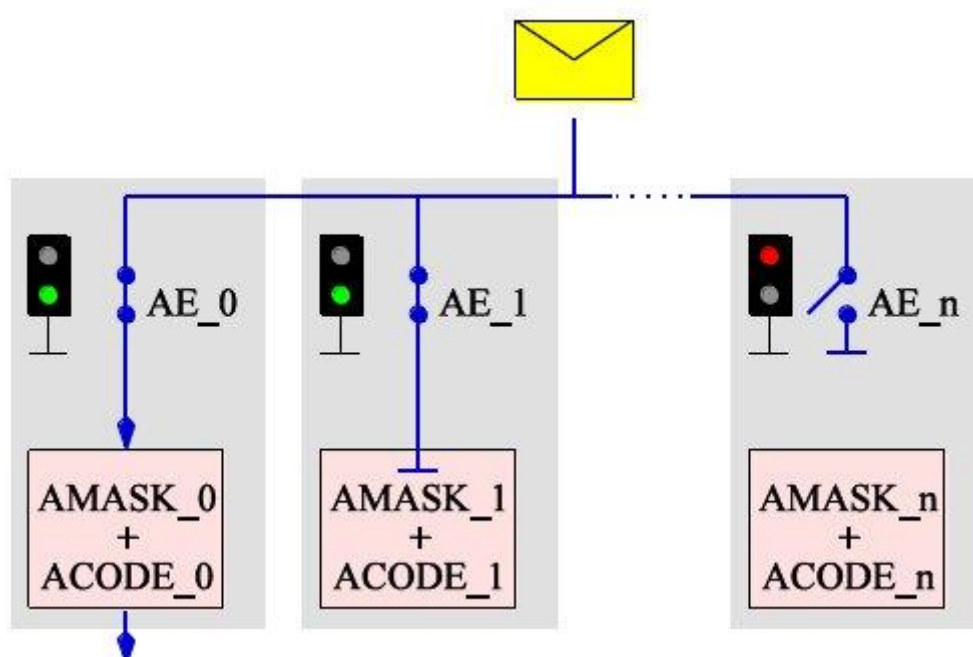


Рисунок 15.10. Пример фильтрации приема

Маска приема определяет, какие биты должны сравниваться, в то время как код приема определяет соответствующие значения. Установка битов маски в 0 позволяет сравнивать выбранные биты кода приема с соответствующими битами идентификатора сообщения. Биты маски, для которых установлено значение 1, отключаются для проверки приема, и это приводит к принятию сообщения.

Биты идентификатора будут сравниваться с соответствующими битами кода приема следующим образом:

- стандартный: ID(10:0) с ACODE(10:0),
- расширенный: ID(28:0) с ACODE(28:0).

Пример: Если AMASK<sub>x</sub>(0)=0, а все остальные биты AMASK<sub>x</sub> равны 1, то значение последнего бита идентификатора должно быть равно ACODE(0) для принятого сообщения. Все остальные биты идентификатора игнорируются фильтром.

Выше на Рисунок 15.10 приведен пример фильтрации приема с использованием нескольких фильтров. В этом примере фильтры 0 и 1 включаются соответствующими битами AE<sub>x</sub> в регистрах ACF\_EN<sub>x</sub>. Все остальные фильтры отключены и поэтому не принимают никаких сообщений. Для включенных фильтров комбинация AMASK<sub>x</sub> и ACODE<sub>x</sub> определяет, принято ли сообщение (как в примере для фильтра 0) или не принято (как в примере для фильтра 1).

Примечание: Отключение фильтра путем установки AE<sub>x</sub>=0 блокирует сообщения. В отличие от этого, отключение бита маски в AMASK<sub>x</sub> отключает проверку этого бита, что приводит к приему сообщений.

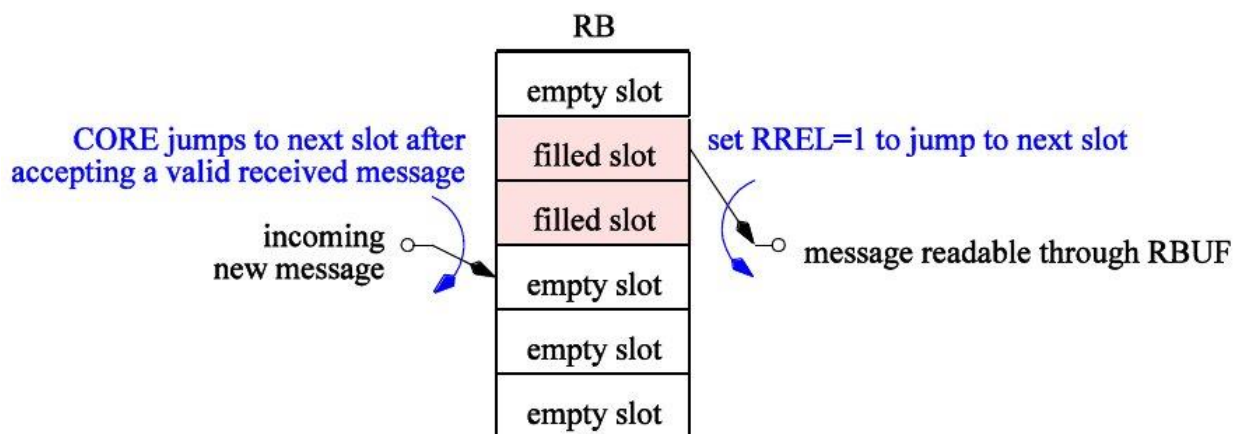
Определения AMASK и ACODE сами по себе не делают различий между стандартными или расширенными кадрами. Если бит AIDEE=1, то принимается значение AIDE, определяемое типом кадра. В противном случае, если AIDE=0, принимаются оба типа.

После сброса питания ядро CAN-CTRL настраивается на прием всех сообщений. ((Фильтр 0 включается при AE<sub>0</sub>=1, все биты в AMASK<sub>0</sub> устанавливаются равными 1, а AIDEE=0. Все остальные фильтры отключены. Фильтр 0 - это единственный фильтр, который имеет определенные значения сброса для AMASK / ACODE, в то время как все остальные фильтры имеют неопределенные значения сброса).

### 15.2.11.2 Прием сообщений

Полученные данные будут сохранены в RB, как показано ниже на Рисунок 15.11. RB настраивается с помощью параметра предварительного синтеза и имеет поведение, подобное FIFO.

Каждое полученное сообщение, которое является действительным и принятым, устанавливает RIF=1, если RIE включен. RSTAT устанавливается в зависимости от состояния заполнения. Когда количество заполненных буферов равно программируемому значению AFWL, то устанавливается RAFIF, если включен RAFIE. В случае, когда все буферы заполнены, RFIF устанавливается, если включен RFIE.



**Рисунок 15.11. Схема FIFO-подобного RB (пример с 6 слотами)**

RB всегда сопоставляет слот сообщения, содержащий самое старое сообщение, с регистрами RBUF.

Максимальная длина полезной нагрузки для сообщений CAN 2.0 составляет 8 байт, а для сообщений CAN FD - 64 байта.

Индивидуальная длина каждого сообщения определяется DLC. Из-за этого RB предоставляет слоты для каждого сообщения, и хост-контроллер должен установить RREL для перехода к следующему слоту RB. Все байты RBUF фактического слота могут быть считаны в любом порядке.

Если RB заполнен, следующее входящее сообщение будет временно сохранено до тех пор, пока оно не будет признано действительным (6 бит OF). Тогда если ROM=0, то самое старое сообщение будет перезаписано самым новым, или, если ROM=1, самое новое сообщение будет отброшено. В обоих случаях ROIF устанавливается, если включен ROIE. Если хост-контроллер считывает самое старое сообщение и устанавливает RREL до того, как новое входящее сообщение станет действительным, то ни одно сообщение не будет потеряно.

### 15.2.11.3 Обработка приема сообщений

Без фильтрации приема ядро CAN-CTRL будет сигнализировать о приеме каждого кадра, и хост должен будет решить, был ли он адресован. Это привело бы к довольно большой нагрузке на хост-контроллер.

Можно отключить прерывания и использовать фильтры приема, чтобы уменьшить нагрузку на хост-контроллер. Для базовой операции RIF устанавливается равным 1, если RIE

включен и ядро CAN-CTRL получило действительное сообщение. Чтобы уменьшить количество прерываний приема, можно использовать RAIE / RAIF (Почти полное прерывание RB) или RFIE / RFIF (Полное прерывание RB) вместо RIE / RIF (Прерывание приема). “Почти полный предел” программируется с помощью AFWL.

RB содержит несколько слотов RB, которые выбираются перед синтезом с использованием общего параметра. Считывание RB должно производиться следующим образом:

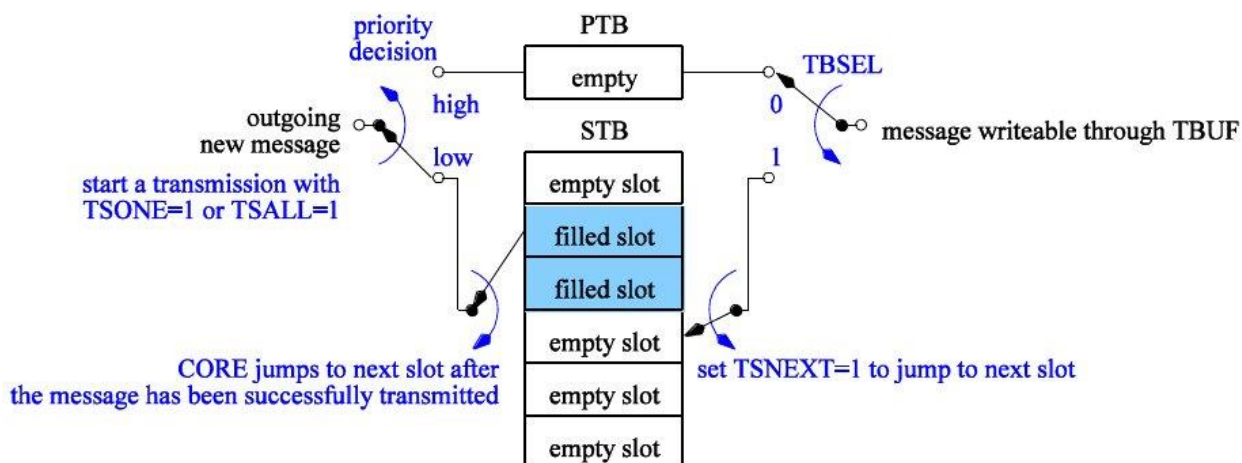
1. Считайте самое старое сообщение из RB FIFO, используя регистры RBUF.
2. Освободите слот RB с RREL=1. При этом выбирается следующее сообщение (следующий слот FIFO). RBUF будет обновляться автоматически.
3. Повторяйте эти действия до тех пор, пока RSTAT не выдаст пустой RB.

Если RB FIFO заполнен и новое полученное сообщение признано действительным (6-ой бит EOF), то одно сообщение будет потеряно (см. бит ROM). До этого события ни одно сообщение не теряется. Это должно дать хост-контроллеру достаточно времени для считывания по крайней мере одного кадра из RB после заполнения RB FIFO и возникновения выбранного прерывания. Чтобы включить это поведение, RB включает в себя на один (скрытый) слот больше, чем указано параметром синтеза RBUF\_SLOTS. Этот скрытый слот используется для получения сообщения, проверки его и проверки, соответствует ли оно фильтрам приема, прежде чем произойдет переполнение.

#### **15.2.11.4      Передача сообщений**

Перед началом любой передачи по крайней мере один из буферов передачи (TB или STB) должен быть загружен сообщением (см. Рисунок 15.12). Сигналы, если PTB заблокирован, а TSSTAT сигнализирует о состоянии заполнения STB. Регистры TBUF обеспечивают доступ как к PTB, так и к STB. Ниже приведен рекомендуемый процесс программирования:

1. Установите TBSEL на желаемое значение, чтобы выбрать либо PTB, либо STB.
2. Запишите кадр в регистры TBUF.
3. Для STB устанавливает TSNEXT=1, чтобы завершить загрузку этого слота STB.



**Рисунок 15.12. Схема PTB и STB в режиме FIFO (пустые PTB и 6 слотов STB)**

Максимальная длина полезной нагрузки для сообщений CAN 2.0 составляет 8 байт, а для сообщений CAN FD - 64 байта.

Индивидуальная длина каждого сообщения определяется DLC. Для кадров удаленного доступа (бит RTR) DLC становится бессмысленным, поскольку кадры удаленного доступа всегда имеют длину данных 0 байт. Хост-контроллер должен установить TSNEXT для перехода к следующему слоту STB. Все байты TBUF могут быть записаны в любом порядке.

Установка TSNEXT=1 не имеет смысла, если TBSEL=0 выбирает PTB. В этом случае TSNEXT автоматически очищается и не наносит никакого вреда.

Бит TPE должен быть установлен для начала передачи при использовании PTB. Чтобы использовать STB, TSONE должен быть настроен на запуск передачи одного сообщения или TSALL для передачи всех сообщений.

PTB всегда имеет более высокий приоритет, чем STB. Если оба буфера передачи имеют порядок передачи, сообщение PTB всегда будет отправляться первым, независимо от идентификаторов кадров. Если передача от STB уже активна, она будет завершена до того, как сообщение от PTB будет отправлено в следующую возможную позицию передачи (следующий межкадровый интервал). После завершения или прерывания передачи РТР ядро CAN-CTRL возвращается к обработке других ожидающих сообщений от STB. См. также главу 15.2.4.3 "Буфер передачи" для получения более подробной информации.

По завершении передачи устанавливаются следующие прерывания передачи:

- Для PTB, TPIF устанавливается, если TPIE включен.



- Для STB, использующего TSONE, TSIF устанавливается, если одно сообщение было завершено и TSIE включен.
- Для STB, использующего TSALL, TSIF устанавливается, если все сообщения были завершены и если TSIE включен. Другими словами: TSIE устанавливается, если STB пуст. Следовательно, если хост-контроллер записывает дополнительное сообщение в STB после запуска передачи TSALL, то дополнительное сообщение также будет передано до того, как будет установлен TSIF.

Бессмысленно устанавливать TSONE или TSALL, пока STB пуст. В таком случае TSONE и TSALL будут сброшены автоматически. Флаг прерывания установлен не будет, и кадр передаваться не будет.

### 15.2.11.5 Прерывание передачи сообщения

Если возникает ситуация, когда сообщение в буфере передачи не может быть отправлено из-за его низкого приоритета, это приведет к блокировке буфера на длительное время. Чтобы избежать этого, хост-контроллер может отозвать запрос на передачу, установив TPA или TSA соответственно, если передача еще не была запущена.

И TPA, и TSA используют один флаг прерывания: AIF. Протокольный автомат CAN выполняет прерывание только в том случае, если она ничего не передает на шину CAN. Следовательно, действительным является следующее:

- Во время арбитража шины прерывание не происходит.
  - Если узел проиграет арбитраж, после этого будет выполнено прерывание.
  - Если узел выиграет арбитраж, кадр будет передан.

Прерывание не происходит во время передачи кадра.

- Если кадр передается успешно, то на главный контроллер подается сигнал об успешной передаче. В этом случае сигнал об отмене не подается. Это делается с помощью соответствующих битов прерывания и состояния.
- После неудачной передачи, когда узел CAN не получает подтверждения, счетчик ошибок увеличивается, и будет выполнено прерывание.
- Если в STB остался хотя бы один кадр, в то время как хост отдал команду на передачу всех кадров (TSALL=1), то хосту сигнализируется как заверченный кадр, так и прерывание.



Из-за этих фактов прерывание передачи может занять некоторое время в зависимости от скорости передачи данных CAN и длины кадра. Если выполняется прерывание, это приводит к следующим действиям:

- ТРА освобождает РТВ, что приводит к значению  $TPE=0$ . Данные кадра все еще сохраняются в РТВ после освобождения РТВ.
- TSA освобождает один отдельный слот для сообщений или все слоты для сообщений STB. Это зависит от того, был ли использован TSONE или TSALL для запуска передачи. TSSTAT будет обновлена соответствующим образом. Освобождение кадра в STB приводит к удалению данных кадра, поскольку хост не может получить к нему доступ.

Одновременная настройка как ТРА, так и TSA не рекомендуется. Если хост-контроллер все равно решит это сделать, то он будет установлен, и обе передачи от РТВ и STB будут прерваны, если это возможно. Как уже говорилось, если одна передача будет завершена до того, как может быть выполнено прерывание, это приведет к сигнализированию об успешной передаче. Поэтому, если включено, могут быть установлены следующие флаги прерывания:

- AIF (один раз для прекращения передачи РТВ и STB),
- TPIF + AIF,
- TSIF + AIF,
- TPIF + TSIF (очень редко, произойдет только в том случае, если хост не обрабатывает TPIF немедленно),
- TPIF + TSIF + AIF (очень редко, произойдет только в том случае, если хост не обрабатывает TPIF и TSIF немедленно).

Чтобы очистить весь STB, необходимо установить как TSALL, так и TSA. Чтобы определить, не может ли сообщение быть отправлено в течение длительного времени из-за потери арбитража, хост может использовать ALIF / ALIE.

#### 15.2.11.6 Полный STB

После записи сообщения в STB,  $TSNEXT=1$  отмечает заполненный слот буфера и переходит к следующему свободному слоту сообщения.  $TSNEXT$  автоматически сбрасывается в 0 с помощью ядра CAN-CTRL после этой операции.

Если последний слот для сообщений был заполнен и, следовательно, все слоты для сообщений заняты, то  $TSNEXT$  остается установленным до тех пор, пока не освободится

новый слот для сообщений. Если TSNEXT=1, то запись в TBUF блокируется ядром CAN-CTRL.

Когда слот становится свободным, ядро CAN-CTRL автоматически сбрасывает значение TSNEXT на 0. Слот становится свободным, если кадр из STB передается успешно или если хост запрашивает прерывание (TSA=1). Если передача TSALL прерывается, то TSNEXT также сбрасывается, но дополнительно полный STB помечается как пустой.

#### 15.2.11.7 Обработка ошибок

С одной стороны, CAN-CTRL выполняет автоматическую обработку ошибок, что означает, что в большинстве случаев хост-контроллер не заботится об ошибках. Это включает в себя автоматическую повторную передачу сообщений и автоматическое удаление полученных сообщений с ошибками. С другой стороны, если требуется хостом, CAN-CTRL может дополнительно предоставлять подробную информацию об ошибках и сигнализировать хосту о каждой ошибке путем прерывания. Это позволяет запускать на хосте такое приложение, как монитор шины CAN.

Каждый узел CAN имеет 3 состояния обработки ошибок:

1. Активная ошибка (Error-Active): Узел автоматически передает активные кадры ошибок при обнаружении ошибки.
2. Пассивная ошибка (Error-Passive): Узел передает пассивный кадр ошибки при обнаружении ошибки. Это означает, что он не передает доминирующее значение на шину, но ожидает кадр ошибки, переданный другими узлами CAN.
3. Выключение шины (Bus Off): После слишком большого количества ошибок узел переходит в режим “выключение шины”, когда он перестает касаться шины.

Для обработки состояний 3-й ошибки каждый узел CAN имеет два счетчика ошибок: счетчик ошибок передачи и счетчик ошибок приема. Оба увеличиваются и уменьшаются, как определено спецификацией CAN, и узел переходит в соответствующее состояние ошибки при достижении уровня счетчика, определенного спецификацией CAN.

Счетчики ошибок увеличиваются при наличии ошибок. Опасные ошибки, которые, вероятно, вызваны этим узлом, приведут к увеличению счетчика на 8, ошибки, которые, вероятно, вызваны другими узлами, приведут к увеличению счетчика на 1. Допустимая передача кадра или допустимые приемы приводят к уменьшению счетчиков. Все это определяется спецификацией CAN и обрабатывается автоматически с помощью CAN-CTRL.

Кадр ошибки (лучше: флаг ошибки) отличается от кадра данных. Это доминирующий импульс, по крайней мере, для 6 последовательных доминирующих битов, что является

нарушением основного бита для других узлов. Хост-контроллер не может дать команду на передачу кадра ошибки. Это делается полностью автоматически с помощью CAN-CTRL.

Если для передачи кадра используется команда CAN-CTRL, то он автоматически пытается выполнить повторную передачу как можно быстрее, пока кадр не будет передан без ошибок или узел не отключит шину. Если получен кадр и CAN-CTRL обнаруживает ошибку, то полученные данные отбрасываются. Из-за автоматически переданного кадра ошибки отправитель кадра повторно передаст кадр. Кадры в RBUF никогда не перезаписываются кадрами с ошибками. Только допустимые принятые кадры могут привести к переполнению RBUF.

#### **15.2.11.8 Состояние выключения шины**

Состояние “шина выключена” сигнализируется с помощью бита состояния BUSOFF в регистре CFG\_STAT (таблица "Конфигурация и регистр состояния CFG\_STAT CFG\_STAT (0xa0)"). Узел CAN автоматически переходит в состояние “шина выключена”, если его счетчик ошибок передачи становится >255. Затем он не будет принимать участия в дальнейших коммуникациях до тех пор, пока снова не вернется в активное состояние с ошибкой. Установка BUSOFF на 1 также устанавливает прерывание EIF, если EIE включен. Узел CAN возвращается в активное состояние с ошибкой, если он сбрасывается при сбросе питания или если он получает 128 последовательностей из 11 рецессивных битов (последовательности восстановления). Пожалуйста, обратите внимание, что между каждой последовательностью восстановления шина может иметь доминирующее состояние.

В состоянии “bus off” RECNT и TECNT остаются неизменными. Пожалуйста, обратите внимание, что при входе в состояние bus off он переворачивается и поэтому может иметь небольшое значение. Поэтому рекомендуется использовать TECNT до того, как узел перейдет в состояние bus off, а затем в состояние бита BUSOFF.

Если узел восстанавливается из состояния “bus off”, то RECNT и TECNT автоматически сбрасываются на 0.

Если кадр ожидает передачи, но узел CAN перешел в состояние отключения шины, то этот кадр остается ожидающим. Если узел возвращается в состояние, вызванное ошибкой, после выключения шины, то будет предпринята попытка передачи ожидающего кадра. Если это нежелательно, то кадр должен быть прерван хост-контроллером.

#### **15.2.11.9 Расширенный отчет о состоянии и ошибках**

Во время передачи данных по шине CAN могут возникать ошибки при передаче. Следующие функции поддерживают их обнаружение и анализ. Это может быть использовано для расширенного мониторинга шины.

##### **15.2.11.9.1 Программируемый предел предупреждения об ошибках**

Ошибки во время приема/передачи подсчитываются по RECNT и TECNT. Программируемый предел предупреждения об ошибках EWL, расположенный в регистре LIMIT, может использоваться хост-контроллером для гибкого реагирования на эти события. Предельные значения могут быть выбраны с шагом 8 ошибок от 8 до 128:

Предел количества ошибок =  $(EWL+1)*8$

EIF прерывания будет установлен, если он включен на EIE при следующих условиях:

- граница предела предупреждения об ошибке была пересечена в любом направлении RECNT или TECNT, или
- бит BUSOFF был изменен в любом направлении.

#### 15.2.11.9.2 Захват потери арбитража (ALC)

Ядро способно определять точное положение бита в Поле арбитража, где арбитраж был потерян. Это событие может быть сигнализировано прерыванием ALIF, если оно включено. Значение ALC остается неизменным, если узел может выиграть арбитраж. Тогда ALC сохраняет старое значение последнего арбитражного проигрыша.

Значение ALC определяется следующим образом: кадр начинается с бита SOF, а затем передается первый бит идентификатора. Этот первый бит идентификатора имеет значение ALC 0, второй бит идентификатора имеет значение ALC 1 и так далее. Порядок битов во всех типах кадров CAN 2.0 и CAN FD приведен на Рисунок 15.4.

Арбитраж разрешен только в поле арбитража (Рисунок 15.4). Следовательно, максимальное значение ALC равно 31, что является битом RTR в расширенных кадрах.

Дополнительная подсказка: Если стандартный кадр удаленного запроса арбитражует с расширенным кадром, то расширенный кадр теряет арбитраж в бите IDE, и ALC будет равен 12. Узел, передающий стандартный кадр удаленного запроса, не заметит, что было проведено арбитражное разбирательство, поскольку этот узел выиграл.

Невозможно получить арбитражный убыток за пределами арбитражного поля. Такое событие является битовой ошибкой.

#### 15.2.11.9.3 Вид Ошибки (KOER)

Ядро распознает ошибки на шине CAN и сохраняет последнее событие ошибки в битах KOER. Ошибка шины CAN может быть сигнализирована прерыванием BEEF, если оно включено. Каждое новое событие ошибки перезаписывает предыдущее сохраненное значение KOER. Поэтому хост-контроллер должен быстро реагировать на событие ошибки. Коды ошибок описаны в таблице Таблица 15.12.

KOER обновляется с каждой новой ошибкой. Поэтому он остается нетронутым, когда кадры успешно передаются или принимаются. Это открывает возможность для отложенного расследования ошибок.

#### 15.2.11.9.4 Прием всех кадров данных (RBALL)

Если RBALL=1, то сохраняются все полученные кадры данных, даже те, которые содержат ошибку. Это справедливо и для режима обратной связи. В RBUF хранятся только кадры данных. Кадры ошибок или кадры перегрузки не сохраняются.

Если включена временная метка CiA 603 (глава 15.4 "Временная метка CiA 603")(TIMEEN=1) и временная метка настроена для EOF (TIMEPOS=1), то в случае ошибки временная метка будет получена в начале кадра ошибки.

Большинство возможных ошибок могут возникнуть только в том случае, если узел является передатчиком кадра. В этом случае кадр будет сохранен в RBUF только в том случае, если активирован режим обратной связи. В зависимости от типа ошибки части кадра, сохраненные в слоте RBUF, могут быть действительными, в то время как другие части неизвестны. В Таблица 15.54 перечислены возможные варианты.

**Таблица 15.54. RBALL и KOER**

KOER	Условие	Описание
no error	All	Успешный прием.
BIT ERROR	Receiver	Может произойти только в слоте ACK. Все сохраненные данные являются действительными.
	Transmitter	Полезная нагрузка всегда является недопустимой. Биты заголовка, включая идентификатор, могут быть допустимыми. На этапе арбитража обнаружение неправильного бита является частью арбитража и, следовательно, не является БИТОВОЙ ОШИБКОЙ. Но если бит материала на этапе арбитража обнаруживается передатчиком кадра неправильно, то это БИТОВАЯ ОШИБКА, и в этом случае биты заголовка являются недопустимыми. Поэтому битам заголовка нельзя доверять, но если заголовок совпадает с ожидаемым заголовком кадра, его можно использовать для дальнейшего исследования.
FORM ERROR	All	Рассматриваются только FORM ERRORS в кадре данных. Это включает ошибки в разделителе CRC, разделителе ACK или битах EOF. Все сохраненные данные являются действительными.
STUFF ERROR	Receiver	Положение STUFF ERROR неизвестно. Все сохраненные данные являются недействительными.
	Transmitter	Может произойти только во время арбитража. Все сохраненные данные являются недействительными.
ACK ERROR	Receiver	Может произойти только в том случае, если узел находится в LOM. Все сохраненные данные являются действительными.
	Transmitter	Может происходить только в режиме обратной связи без автоматического подтверждения. Все сохраненные данные являются действительными.
CRC ERROR	Receiver	Все сохраненные данные являются недействительными.

Пожалуйста, обратите внимание, что даже если в Таблица 15.54 указано, что (части) данных действительны при определенных условиях, необходимо соблюдать осторожность. На это нет никакой гарантии, и это зависит от причины ошибки, если это утверждение верно.

Пример: если узел имеет неправильную конфигурацию синхронизации битов, то нет правильной синхронизации и, следовательно, нет действительных данных. Поэтому утверждение о том, что (части) данных являются действительными, сигнализирует только о возможности того, что это так.

#### **15.2.11.10     Расширенные характеристики**

##### **15.2.11.10.1    Однократная передача**

Иногда автоматическая повторная передача нежелательна. Следовательно, порядок передачи сообщения только один раз может быть установлен отдельно для буферов передачи PTB битом TPSS и для буфера передачи STB битом TSSS.

В этом случае повторная передача не будет выполнена в случае ошибки или потери арбитража, если выбранная передача активна.

В случае немедленной успешной передачи нет никакой разницы с обычной передачей.

Но в случае неудачной передачи произойдет следующее:

TRIF устанавливается, если он включен, соответствующий слот буфера передачи очищается.

В случае ошибки KOER и счетчики ошибок обновляются. BEIF устанавливается, если включен BEIE, и другие флаги прерывания ошибки будут действовать соответствующим образом.

В случае проигранного арбитража ALIF устанавливается, если ALIE равно enable.

Следовательно, для однократной передачи TRIF самостоятельно не указывает, был ли кадр успешно передан. Поэтому однократную передачу следует использовать только вместе с BEIF и ALIF.

Если для TSALL используется однократная передача и в STB содержится более одного кадра, то для каждого кадра выполняется однократная передача. Независимо от того, если какой-либо из кадров не был успешно передан (например, из-за ошибки ACK), CAN-CTRL переходит к следующему кадру и останавливается, если STB пуст. Поэтому в этом сценарии только счетчики ошибок указывают на то, что произошло. Это может быть довольно сложно оценить, потому что, если один из двух кадров получил ошибки, хост не может определить, какой из них был успешным.

Если шина занята другим кадром, если запущена передача одного кадра, то CAN-CTRL ожидает, пока шина не будет работать в режиме ожидания, а затем попытается передать кадр с одним кадром.

#### **15.2.11.10.2 Режим только прослушивания (LOM)**

LOM обеспечивает возможность мониторинга шины CAN без какого-либо воздействия на шину. LOM с помощью CAN-CTRL совместим с функцией мониторинга шины, определенной в спецификации CAN FD.

Другим приложением является автоматическое определение битрейта, при котором хост-контроллер пробует различные настройки синхронизации до тех пор, пока не возникнет никаких ошибок.

Ошибки будут отслеживаться (KOER, BEEF) в LOM.

В LOM ядро не может записывать доминирующие биты на шину (нет активных флагов ошибок или флагов перегрузки, нет подтверждения). Это делается с помощью следующих правил:

- В LOM машина протокола действует так, как если бы она находилась в пассивном режиме ошибок, где генерируются только рецессивные флаги ошибок. Только протокольная машина действует так, как будто находится в пассивном режиме ошибки. Все остальные компоненты, включая регистры состояния, не затрагиваются.
- В LOM машина протокола не генерирует доминирующий ACK.
- Счетчики ошибок остаются неизменными независимо от любого условия ошибки.

Важные факты, касающиеся ACKs для LOM:

- Если кадр передается узлом, то ACK, видимый на шине, будет сгенерирован только в том случае, если к шине подключен по крайней мере один дополнительный узел, который не находится в LOM. Тогда ошибки не будет, и все узлы (также находящиеся в LOM) получают кадр.
- Если после ошибки ACK есть активный или пассивный флаг ошибки, то узел в LOM может обнаружить это как ошибку ACK.

Активация LOM не должна выполняться во время активной передачи. Об этом должен позаботиться главный контроллер. Дополнительной защиты от CAN-CTRL нет.

Если LOM включен, то передача не может быть запущена.



Режим обратной связи (внешний) LBME (глава 15.2.11.10.4 "Режим петли обратной связи (LBMI и LBME)") играет важную роль в поведении LOM. Если LBME отключен, то LOM действует, как описано выше, и узел не может записать ни одного доминирующего бита на шину. Но если активирован LBME, то узлу разрешается передавать кадр. LBME разрешает только передачу кадра, включая необязательный self-ACK (SACK), но узел не будет отвечать ACK на кадры от других узлов и не будет генерировать кадры ошибок или перегрузки. Таким образом, комбинация LOM и LBME представляет собой "бесшумный приемник, который способен передавать при необходимости".

#### 15.2.11.10.3 Проверка подключения к шине

Чтобы проверить подключен ли узел к шине необходимо выполнить следующие действия:

- Передайте кадр. Если узел подключен к шине, то его биты TX видны на его входе RX.
- Если к шине CAN подключены другие узлы, то ожидается успешная передача (включая подтверждение от других узлов). Ошибка не будет сигнализирована.
- Если узел является единственным узлом, подключенным к шине CAN (но соединение между шиной, приемопередатчиком и ядром CAN-CTRL в порядке), то первая обычная ошибка возникает в слоте ACK из-за отсутствия подтверждения от других узлов. Затем будет сгенерировано прерывание с ошибкой BEIF, если оно включено, и KOER="100" (ошибка ACK).
- Если соединение с приемопередатчиком или шиной прервано, то сразу после бита SOF будет установлено прерывание ошибки BEIF и KOER="001" (ошибка БИТА).

#### 15.2.11.10.4 Режим петли обратной связи (LBMI и LBME)

CAN-CTRL поддерживает два режима петли обратной связи: внутренний (LBMI) и внешний (LBME). Оба режима приводят к приему собственного переданного кадра, который может быть полезен для самопроверки. Подробности ниже на Рисунок 15.13.

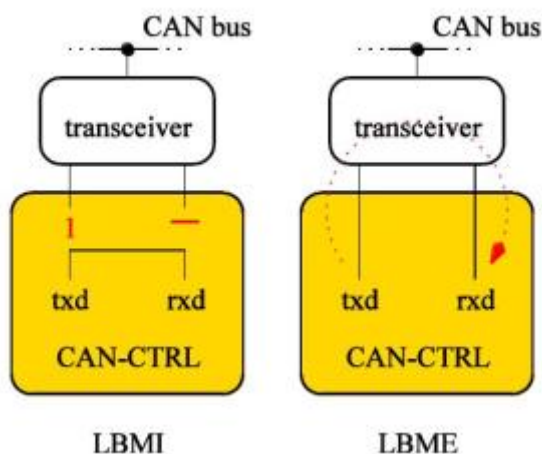
В LBMI CAN-CTRL отключен от шины CAN, а выход txd установлен на рецессивный. Поток выходных данных внутренне подается обратно на вход. В LBMI узел генерирует self-ACK, чтобы избежать ошибки ACK.

В LBME CAN-CTRL остается подключенным к приемопередатчику, и переданный кадр будет виден на шине. С помощью приемопередатчика CAN-CTRL получает свой собственный кадр. В LBME узел не генерирует self-ACK, когда SACK=0, но генерирует self-ACK, когда SACK=1. Следовательно, в LBME с SACK=0 при передаче кадра возможны два результата:

1. Другой узел тоже получает кадр и генерирует подтверждение. Это приведет к успешной передаче и приему.



2. Ни один другой узел не подключен к шине, и это приводит к ошибке АСК. Чтобы избежать повторных передач и увеличения счетчиков ошибок, рекомендуется использовать TPSS или TSSS, если неизвестно, подключены ли другие узлы.



**Рисунок 15.13. Режим петли обратной связи: внутренний и внешний**

В режиме петли обратной связи ядро получает свое собственное сообщение, сохраняет его в RBUF и устанавливает соответствующие флаги прерывания приема и передачи, если они включены.

LBMI может быть полезен для тестирования микросхем и программного обеспечения, в то время как LBME может протестировать приемопередатчик и подключения к нему.

Активация LBMI или LBME не должна выполняться во время активной передачи. Об этом должен позаботиться главный контроллер. Дополнительной защиты от CAN-CTRL нет.

Если узел подключен к шине CAN, то переключение обратно с LBMI на нормальную работу не должно выполняться простым установлением LBMI в 0, потому что тогда может случиться так, что это происходит как раз в тот момент, когда другой узел CAN передает. В этом случае возврат к нормальной работе должен быть выполнен путем установки значения бита RESET на 1. Это автоматически сбрасывает значение LBMI до 0. Наконец, RESET может быть отключен, и ядро возвращается к нормальной работе. В отличие от этого, LBME может быть отключен каждый раз.

LBME можно использовать в сочетании с LOM. Подробная информация приведена в главе 15.2.11.10.2 "Режим только прослушивания (LOM)".

Прием собственного переданного кадра запускает соответствующие прерывания передачи и приема. Прерывание приема будет установлено раньше, потому что кадр становится действительным для приемника при предпоследнем бите EOF, но для передатчика при последнем бите EOF.

#### 15.2.11.10.5 Режим ожидания приемопередатчика

С помощью бита регистра STBY управляется выходной сигнал stby. Его можно использовать для активации режима ожидания приемопередатчика. Это поведение совместимо с приемопередатчиком NXP TJA1049.

После включения режима ожидания передача невозможна, и поэтому TPE, TSONE и TSALL не могут быть установлены. С другой стороны, CAN-CTRL не позволяет устанавливать STBY, если передача уже активна (установлены TPE, TSONE или TSALL).

Если установлен параметр STBY, приемопередатчик переходит в режим низкого энергопотребления. В этом режиме он не может принимать кадр на полной скорости, но отслеживает шину CAN на предмет доминирующего состояния. Если доминирующее состояние активно в течение времени, определенного в спецификации приемопередатчика, приемопередатчик будет выдавать низкий уровень сигнала rxd. Если rxd низкий, CAN-CTRL автоматически очищает STBY до 0, что отключает режим ожидания для приемопередатчика. Это делается бесшумно, без прерывания работы хост-контроллера.

Переключение из режима ожидания в активный режим занимает некоторое время для приемопередатчика, и поэтому начальный кадр пробуждения не может быть успешно принят. Поэтому узел, недавно находившийся в режиме ожидания, не будет отвечать ACK. Если ни один узел CAN на шине не отвечает на кадр пробуждения с помощью ACK, то это приводит к ошибке ACK для передатчика кадра пробуждения. Затем передатчик автоматически повторит кадр. Во время повторения приемопередатчик вернется в активный режим, и CAN-CTRL примет кадр и ответит ACK.

Вкратце: один узел передает кадр для пробуждения. Если все остальные узлы находятся в режиме ожидания, то передатчик получает сообщение об ошибке ACK и автоматически повторяет кадр. Во время повторения узлы возвращаются в активный режим и будут отвечать ACK.

Сброс битов на STBY не влияет.

#### 15.2.11.10.6 Сброс счетчика ошибок

В соответствии с CAN / CAN FD, стандартный RECNT подсчитывает ошибки приема, а TECNT подсчитывает ошибки передачи. После слишком большого количества ошибок передачи узел CAN должен перейти в состояние отключения шины (глава 15.2.11.7 "Обработка ошибок" ). Сброс битов не изменяет счетчики ошибок или состояние отключения шины. Спецификация CAN / CAN FD определяет правила, как отключить состояние отключения шины и уменьшить количество счетчиков ошибок. Хороший узел восстановится после всего этого автоматически, если проблемы вызваны только временной ошибкой. Классическая спецификация CAN 2.0B требует такого автоматического поведения без взаимодействия с хост-контроллером, чтобы избежать проблемы с узлом "babbling idiot".

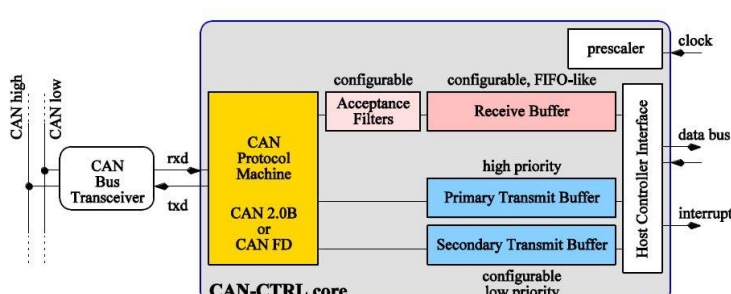
Спецификация CAN FD смягчает это и позволяет хост-контроллеру переопределять автоматическую обработку счетчика ошибок. Но это следует использовать с особой осторожностью и рекомендуется использовать только для целей отладки.

Запись 1 в разрядную шину сбрасывает счетчики ошибок и, следовательно, заставляет узел выйти из состояния отключения шины. Это делается без установки EIF.

### 15.2.11.11 Сброс программного обеспечения

Если бит RESET в CFG\_STAT установлен в 1, то программный сброс активен. Несколько компонентов принудительно переводятся в состояние сброса, если RESET=1, но сброс затрагивает не все компоненты. Некоторые компоненты чувствительны только к аппаратному сбросу. Значения сброса всех битов всегда одинаковы для программного и аппаратного сброса.

Таблица 15.55. Программный сброс

Регистр	Сброс	Комментарий
ACFADR	No	-
ACODE_x	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
ACFA	Yes	-
AE_x	No	-
AFWL	No	-
AIF	Yes	-
ALC	Yes	-
ALIE	No	-
ALIF	Yes	-
AMASK_x	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
BEIE	No	-
BEIF	Yes	-
BUSOFF	(No)	Сброс счетчика ошибок при установке BUSOFF=1 также приводит к сбросу BUSOFF.
EIE	No	-
EIF	No	-
EPASS	No	
EPIE	No	-
EPIF	Yes	-
EWARN	No	-
EWL	Yes	-
FD_ISO	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если 0.
FSTIM	No	-

Регистр	Сброс	Комментарий
F_PRESC	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
F_Seg_1	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
F_Seg_2	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
F_SJW	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
HELOC	Yes	-
KOER	Yes	-
LBME	Yes	-
LBMI	Yes	-
LOM	No	-
MAEIF	Yes	-
MDEIF	Yes	-
MDWIE	No	-
MDWIF	Yes	-
MEL	No	-
MEBP1	No	-
MEBP2	No	-
ME1EE	No	-
ME2EE	No	-
MEAEE	No	-
MEEEC	No	-
MENEC	No	-
MES	No	-
MPEN	No	Бит доступен для записи, если RESET=1, и заблокирован для записи, если 0.
RACTIVE	Yes	Прием немедленно отменяется, даже если прием активен. АСК не будет сгенерирован.
RAFIE	No	-
RAFIF	Yes	-
RBALL	Yes	-
RBUF	(Yes)	Все слоты RB помечены как пустые. RBUF содержит неизвестные данные.
RECNT	No	Сброс счетчика ошибок возможен при установке значения BUSOFF=1.
REF_ID	No	-
REF_IDE	No	-
RFIE	No	-
RFIF	Yes	-
RIE	No	-
RIF	Yes	-
ROIE	No	-
ROIF	Yes	-
ROM	No	-
ROV	Yes	Все слоты RB помечены как пустые.
RREL	Yes	-
RSTAT	Yes	Все слоты RB помечены как пустые.
SACK	Yes	-
SEIF	Yes	-
SELMASK	No	-
STBY	No	-
SWIE	No	-
SWIF	Yes	-

Регистр	Сброс	Комментарий
S_PRESC	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
S_Seg_1	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
S_Seg_2	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
S_SJW	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.
TACTIVE	Yes	Все передачи немедленно прекращаются с помощью сброса. Если передача активна, это приведет к ошибочному кадру. Другие узлы будут генерировать кадры ошибок.
TBE	Yes	-
TBF	Yes	-
TBPTR	No	-
TBSEL	Yes	TBUF исправлен, чтобы указывать на PTB.
TBUF	(Yes)	Все слоты STB помечены как пустые. Из-за TBSEL TBUF указывает на PTB.
TECNT	No	Сброс счетчика ошибок возможен при установке значения BUSOFF=1.
TEIF	Yes	-
TIMEEN	No	-
TIMEPOS	No	-
TPA	Yes	-
TPE	Yes	-
TSA	Yes	-
TSALL	Yes	-
TSMODE	No	-
TSNEXT	Yes	-
TSONE	Yes	-
TPIE	No	-
TPIF	Yes	-
TPSS	Yes	-
TSFF	Yes	Все слоты STB помечены как пустые.
TSIE	No	-
TSIF	Yes	-
TSSS	Yes	-
TSSTAT	Yes	Все слоты STB помечены как пустые.
TTEN	Yes	-
TTIF	Yes	-
TTIE	No	-
TPPTR	No	-
TTS	No	-
TTTBM	No	-
TTYPE	No	-
TT_TRIG	No	-
TT_WTRIG	No	-
T_PRESC	No	-
TXB	Yes	-
TXS	Yes	-
WTIE	No	-
WTIF	Yes	-
XMREN	No	Регистр доступен для записи, если RESET=1, и заблокирован для записи, если RESET=0.

## 15.3 Синхронизируемый по времени CAN

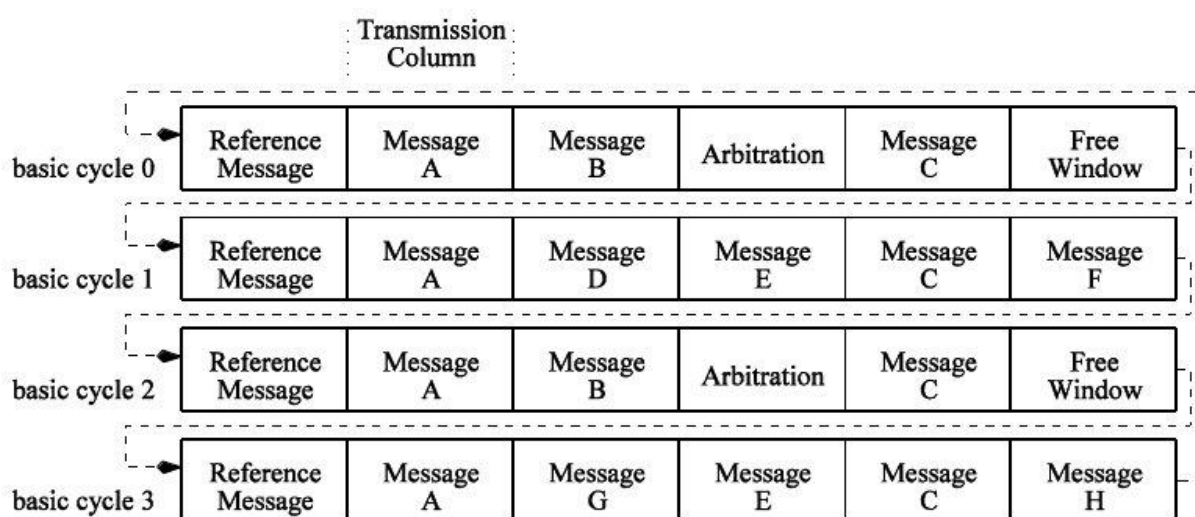
Синхронизируемый по времени CAN (TTCAN) - это режим работы в соответствии с ISO 11898-4, в котором кадры будут передаваться только в заранее определенные временные интервалы. Существует три типа временных окон:

- Эксклюзивное временное окно (только одному узлу разрешено передавать один кадр с определенным ID),
- Свободное временное окно (неиспользуемое временное окно для дальнейшего расширения сети),
- Временное окно арбитража (несколько узлов могут передавать кадр, и происходит арбитраж).

TTCAN является необязательным и может быть отключен с помощью общего параметра (глава "Параметры конфигурации").

Время определяется системным администратором TTCAN в автономном режиме и организуется в системной матрице, как показано ниже на Рисунок 15.14.

Строка называется базовым циклом, и она начинается с опорного сообщения. Опорное сообщение передается мастером времени. Другие сообщения могут передаваться любым узлом, включая мастер времени.



**Рисунок 15.14. Пример системной матрицы**

Время SOF сообщения - это Sync\_Mark. Sync\_Mark опорного сообщения - это метка ссылки. Синхронизация осуществляется с помощью 16-битного таймера, работающего в автономном режиме. Разница между таймером и значением Ref\_Mark - это время цикла.

Другими словами: каждый базовый цикл начинает время цикла с отметки Ref\_Mark. Время цикла сохраняется в RBUF в виде временной метки.

Продолжительность временного окна определяется администратором TTCAN, и его достаточно для передачи одного сообщения. Поэтому сообщения должны передаваться в режиме одиночного выстрела. Есть только одно исключение из этого правила: если несколько временных окон арбитража объединены, то можно включить повторную передачу. Но это должно быть отключено достаточно рано, чтобы не повлиять на следующее временное окно.

Чтобы передать кадр во временном окне, CAN-CTRL предлагает аппаратный триггер. Этот триггер должен быть настроен хостом. Нужный кадр должен быть сохранен в слоте TBUF, и триггер должен указывать на этот слот. Затем, если время цикла совпадает со временем, определенным для триггера, CAN-CTRL автоматически передает кадр в режиме одиночной съемки.

Если триггер становится активным, это будет сигнализировано хосту прерыванием. Затем хосту необходимо настроить следующий триггер для следующего временного окна. Для этого требуется ответ хоста в режиме реального времени. Продолжительность одного временного окна - это время, которое разрешено использовать хосту.

Если CAN-CTRL должен работать как мастер времени, то опорное сообщение должно быть помещено в слот TBUF, как и все другие сообщения, и будет передано, когда активируется аппаратный триггер. Обнаружение опорного сообщения выполняется автоматически всеми подчиненными устройствами времени и ведущим устройством времени. Это делается путем установки правильного бита REF\_ID и REF\_IDE опорного сообщения в регистрах REF\_MSG\_x. При обнаружении CAN-CTRL автоматически обновляет метку Ref\_Mark, которая запускает время цикла.

В дополнение к триггеру для сообщений CAN-CTRL включает в себя сторожевой триггер. Это следует использовать, чтобы проверить, не прошло ли слишком много времени с момента последнего опорного сообщения. Хост должен сконфигурировать триггер наблюдения для периодического случая, когда за каждым базовым циклом следует следующий базовый цикл, или для случая, вызванного событием, когда существует временной промежуток между базовыми циклами и следующий цикл запускается по событию. Если триггер наблюдения становится активным, это приводит к прерыванию наблюдения.

Помимо поддержки стандарта ISO 11898-4, CAN-CTRL предлагает сочетание управляемого события обмена данными CAN в сочетании с меткой времени для полученных сообщений. Этот режим выбирается, если бит регистра TTTBM=0. В этом режиме CAN-CTRL действует аналогично событийной связи CAN (TTEN=0), но могут быть обнаружены опорные сообщения и предоставлены временные метки для полученных сообщений. Кроме того, в этом режиме поддерживаются только триггеры времени и триггер часов.



### 15.3.1 TBUF в режиме TTCAN

Поведение, если значение TBUF зависит от бита регистра TTTBM. Если TTTBM=1, то каждый слот TBUF может быть адресован хост-контроллером, который требуется для использования триггеров передачи. В противном случае, если TTTBM=0, то поддерживаются только временные метки и временные триггеры.

#### 15.3.1.1 TBUF в режиме TTCAN, если TTTBM=1

В режиме TTCAN с TTTBM=1, STB используется в качестве массива слотов сообщений. Каждый слот может быть адресован с помощью TBPTR. Хост может пометить слот как заполненный или пустой с помощью TBF и TBE. Заполненные слоты заблокированы для записи. TBSEL и TSNEXT не имеют никакого значения в режиме TTCAN и игнорируются.

PTB может быть адресован с помощью TBPTR=0. Это делает PTB пригодным для использования как любой другой слот STB. На самом деле, PTB не имеет никаких особых свойств в этом режиме и связан с STB. Кроме того, это приводит к тому, что успешная передача всегда сигнализируется с использованием TSIF.

В режиме TTCAN нет операции FIFO и нет решения о приоритете для TBUF. Кроме того, для передачи может быть выбран только один кадр.

Триггер сообщения определяет время, когда сообщение должно быть передано (начало окна времени), и выбирает сообщение с помощью указателя TTPTR. Если событие триггера имеет место, то начинается передача выбранного сообщения. Наконец, устанавливается прерывание триггера, сигнализирующее хосту о том, что необходимо подготовить следующее действие.

Все передачи могут быть запущены только с помощью триггера. Подробности см. в главе 15.3.4 "Типы триггера TTCAN". TPE, TSONE, TSALL, TPSS и TPA фиксируются на 0 и игнорируются в режиме TTCAN.

#### 15.3.1.2 TBUF в режиме TTCAN, если TTTBM=0

TTTBM=0 предлагает комбинацию управляемого события обмена данными CAN и временной отметки для полученных сообщений.

В этом режиме PTB и STB обеспечивают такое же поведение, как если бы TTEN=0. Тогда PTB всегда имеет более высокий приоритет, чем STB, и STB может работать в режиме FIFO (TSMODE=0) или в режиме приоритета (TSMODE=1).

### 15.3.2 Работа TTCAN

После включения питания мастер времени должен выполнить инициализацию, как определено в ISO 11898-4. В сети CAN может быть до 8 потенциальных мастеров времени. Каждый из них имеет свой собственный идентификатор опорного сообщения (последние 3



бита идентификатора). Потенциальные мастера времени могут попытаться передать свое опорное сообщение в соответствии с их приоритетом. Мастера времени с более низким приоритетом должны попытаться передать свои опорные сообщения позже.

Если задано значение TTEN, то запущен 16-разрядный таймер. Если обнаружено опорное сообщение или если мастер времени успешно передает свое опорное сообщение, то CAN-CTRL копирует метку синхронизации этого сообщения в метку Ref\_Mark, которая устанавливает время цикла равным 0. При приеме опорного сообщения будет установлен RIF, соответственно при успешной передаче будет установлен TPIF или TSIF. Затем хосту необходимо подготовить триггер для следующего действия.

Триггером для действия может быть триггер приема. Это просто запускает прерывание, и его можно использовать для определения того, не получено ли ожидаемое сообщение. Такой триггер также может использоваться для других действий, но это зависит от хост-приложения.

Другой триггер - это триггер передачи. Это запускает кадр в слоте TBUF, где триггер указывает на TTPTR. Если слот TBUF помечен как пустой, то передача не запускается, но устанавливается триггерное прерывание.

Возможно, что одна задача хоста обновляет слот сообщений новым сообщением, если это необходимо хост-приложению. Это может быть, например, новое значение датчика. Позже, если TTCAN потребует передачи этого сообщения, оно будет активировано триггером. Другими словами: одна задача хоста должна отвечать за TTCAN, а другая задача хоста может обновлять слоты сообщений. Это требует, чтобы один слот сообщения использовался исключительно для одного сообщения (например, слот 1 для датчика температуры). Если доступно недостаточно слотов TBUF, то слоты должны быть общими для разных сообщений.

Хост-приложение TTCAN должно отслеживать системную матрицу. Если триггер становится активным, то хост-приложению необходимо подготовить следующий столбец передачи. Хост должен также обрабатывать основные циклы. Опорное сообщение включает в себя количество циклов.

Большинство операций в соответствии с ISO 11989-4 требуют передачи одним кадром. Дополнительные сведения об обработке успешных и неудачных передач см. в главе "Передача одного кадра".

### 15.3.3 Синхронизация TTCAN

CAN-CTRL поддерживает ISO 11898-4 уровня 1. Это включает в себя 16-битный таймер, работающий со скоростью 16-битного времени (определяется S\_PRESC, S\_SEG\_1, S\_SEG\_2). Дополнительный предделитель определяется T\_PRESS. Если THEN=1, то таймер ведет непрерывный отсчет.

В конце сообщения значением таймера является отметка Sync\_Mark. Если сообщение было опорным сообщением, то это значение копируется в Ref\_Mark. Время цикла - это значение таймера за вычетом метки Ref\_Mark. Это время, которое используется в качестве временной метки для полученных сообщений или в качестве времени запуска для сообщений, которые необходимо передать. Включена защита от переполнения, и поэтому время цикла всегда строго монотонно внутри базового цикла.

ISO 11898-4 не включает переключение скорости передачи данных CAN FD в бодах. Таким образом, CAN-CTRL всегда запускает таймер с низкой номинальной скоростью передачи данных. Таймер работает свободно, и на него не влияет синхронизация или переключение скорости передачи данных. Поэтому тик таймера не является синхронным с началом бита CAN.

Таймер работает в частотном домене CAN, в то время как все биты управления и состояния находятся в частотном домене хоста. Поэтому чистое значение таймера не может быть прочитано для хоста, поскольку для этого потребуется перекрытие диапазонов изменения тактовых частот. Триггерные события необходимо использовать для синхронизации действий, требуемых хостом, с таймером.

Прерывания из-за триггерных событий, прием или передача требуют перекрытие диапазонов изменения тактовых частот и, следовательно, включают некоторую задержку синхронизации. Это становится актуальным только в том случае, если после запуска хост-приложение решает начать передачу. (Поэтому вместо этого рекомендуется использовать триггер передачи.) Во всех остальных случаях у хост-приложения есть достаточно времени, чтобы подготовить все действия к следующему событию запуска (которое является следующим временным окном). Почти все хосты достаточно быстры, чтобы выполнять множество действий в течение кадра CAN (который является продолжительностью временного окна).

Синхронизация в соответствии с ISO 11898-4, уровень 1 не идеален. Время цикла отсчитывается не синхронно с битами CAN, поскольку бит CAN может быть сокращен или удлинен из-за синхронизации CAN. Поэтому время цикла одного узла может отличаться по сравнению со временем цикла других узлов. Во многих случаях это будет +/- 1 отсчет частоты, но не обязательно именно 1. Начало нового базового цикла с передачей опорного сообщения приведет к повторной синхронизации узлов.

Если триггер передачи становится активным, то соответствующая передача может быть начата только со следующего бита CAN. Следовательно, самая ранняя точка передачи SOF кадра будет находиться в TT\_TRIG+1.

### 15.3.4 Типы триггера TTCAN

Тип триггера определяется с помощью TTYPE. TTPTR - это указатель на слот сообщения TB, а TT\_TRIG определяет время цикла триггера.

Триггеры и связанные с ними действия должны завершиться до достижения максимального времени цикла 0xffff, поскольку это максимальная длина базового цикла.

За исключением немедленного триггера, все триггеры устанавливают TTIF, если TTIE включен.

Если TTTBM=1, поддерживаются только временные триггеры. Использование других триггеров в этом режиме приведет к установке TEIF.

Если триггер активируется после доступа на запись в TT\_TRIG\_1, то доступ на запись в TT\_CFG\_0, TT\_CFG\_1, TT\_TRIG\_0 и TT\_TRIG\_1 блокируется до тех пор, пока не будет достигнуто время триггера (TTIF устанавливается, если TTIE установлен) или обнаружена ошибка (TEIF установлен). Поэтому ни один новый триггер не может переопределить активный триггер. Блокировка записи также снимается, если TTEN=0.

#### **15.3.4.1 Триггер немедленной передачи**

Триггер немедленной передачи запускает немедленную передачу кадра, на который указывает TTPTR. TTIF не установлен. Чтобы запустить триггер, необходимо записать TT\_TRIG\_1. Значение, которое записывается, не заботится о триггере немедленной передачи.

TPE, TSONE, TSALL не могут быть использованы в режиме TTCAN. Непосредственным триггером является замена. Только передача одного кадра может быть начата с триггера немедленной передачи. Хост не должен командовать вторым триггером немедленной передачи до того, как первая передача будет завершена успешно или неудачно.

Для триггера немедленной передачи TSSS может выбрать одиночный режим. Передача может быть прервана с помощью TSA. (TPSS и TPA не имеют никакого значения.)

Если TTPTR триггера немедленной передачи указывает на пустой слот, то устанавливается значение TEIF.

#### **15.3.4.2 Временной триггер**

Триггер времени просто генерирует событие, устанавливая TTIF, и, следовательно, генерирует прерывание. Никаких других действий не выполняется.

Триггер времени можно использовать в качестве триггера приема. Если узел ожидает, что сообщение будет получено во временном окне, то, если сообщение отсутствует и RIF не установлен, для сигнализации об этом может быть использован триггер приема. Триггер приема должен быть установлен после последнего момента, когда ожидается, что сообщение будет успешно получено.

Если TT\_TRIG меньше, чем фактическое время цикла, то устанавливается THIEF.

Временной триггер может быть использован, если  $TTTBM=1$ . Это единственный тип триггера, доступный в этом режиме.

#### **15.3.4.3 Триггер одиночной передачи**

Триггер одиночной передачи предназначен для использования в эксклюзивных окнах времени, когда сообщение должно быть передано в режиме одиночной передачи. Выбранное сообщение определяется с помощью  $TPTR$ .

Одиночный режим используется автоматически независимо от состояния  $TSSS$ . Регистровый бит  $TSSS$  игнорируется и остается неизменным.

Триггеры одиночной передачи предназначены для использования в эксклюзивных окнах времени. Для этого ISO 11898-4 определяет окно enable передачи продолжительностью до 16 тактов времени цикла. Биты регистра  $TEW(3:0)+1$  определяют количество тактов. Кадр не может быть запущен, если шина занята другим кадром. Это не должно происходить в эксклюзивном окне времени, но окно enable передачи гарантирует отсутствие задержки запуска, которая привела бы к нарушению следующего окна времени. Если окно enable передачи закрывается и кадр не может быть запущен, то он прерывается. В результате слот ТВ кадра будет помечен как пустой, и AIF будет установлен, если AIE установлен. Данные кадра в слоте ТВ не будут затронуты, и поэтому слот нужно снова пометить как заполненный только в том случае, если те же данные должны быть переданы при следующей попытке.

Если  $TT\_TRIG$  меньше, чем фактическое время цикла, то  $TEIF$  устанавливается и никаких действий не выполняется.

#### **15.3.4.4 Триггер запуска передачи**

Триггер запуска передачи предназначен для использования в объединенных окнах времени арбитража, где несколько узлов могут передавать сообщения и может происходить арбитраж. Выбранное сообщение определяется с помощью  $PTR$ .

$TSSS$  определяет, используется ли режим повторной передачи или одиночный режим.

Если выбранный кадр не может быть передан (арбитраж потерян, несколько передач после ошибки), то передачу можно остановить с помощью триггера остановки передачи (глава 15.3.4.5 "Триггер остановки передачи").

Если  $TT\_TRIG$  меньше, чем фактическое время цикла, то  $TEIF$  устанавливается и никаких действий не выполняется.

#### **15.3.4.5 Триггер остановки передачи**

Триггер остановки передачи предназначен для остановки передачи, которая запускается с помощью триггера запуска передачи (глава 15.3.4.4 "Триггер запуска передачи").

Если передача остановлена, то кадр прерывается. Следовательно, AIF установлен, если установлен AIE, а слот ТВ кадра помечен как пустой. Данные кадра в слоте ТВ не будут затронуты, и поэтому слот должен быть помечен как заполненный только в том случае, если те же данные должны быть переданы при следующей попытке. Более подробная информация содержится в главе 15.2.11.5 "Прерывание передачи сообщения".

Поведение триггера остановки передачи аналогично завершению окна enable передачи, используемого триггером одиночной передачи (глава 15.3.4.3 "Триггер одиночной передачи"). Если триггер остановки передачи указывает на пустой слот (что означает, что сообщение уже было передано), то действие не выполняется, и TEIF не устанавливается.

Если TT\_TRIG меньше, чем фактическое время цикла, то TEIF устанавливается, но выполняется остановка.

### 15.3.5 Триггер наблюдения TTCAN

В отличие от общих типов триггеров, описанных в главе 15.3.4 "Типы триггера TTCAN", триггер наблюдения имеет специальный флаг прерывания WTIF. Если количество циклов равно значению, определяемому с помощью TT\_WTRIG, то значение WTIF установлено, и значение WTIE установлено.

Триггер наблюдения предназначен для использования, если прошло слишком много времени с момента последнего действительного опорного сообщения. Опорные сообщения могут приниматься в периодическом цикле или после события. Хост-приложение должно позаботиться об этом и соответствующим образом настроить триггер наблюдения.

Значение по умолчанию WTIE равно 1, и поэтому триггер наблюдения по умолчанию со значением 0xffff автоматически валиден. Чтобы отключить триггер наблюдения, WTIE должен быть установлен в 0.

Если TT\_WTRIG обновлен и он меньше, чем фактическое время цикла, устанавливается значение TEIF.

Триггер наблюдения может быть использован, если TTTBM=1.

### 15.3.6 Отключение TTCAN

Если TTCAN отключен с помощью общего параметра, то соответствующие регистры удаляются, а биты фиксируются на сброшенных значениях. Это справедливо для TTTBM, TBSLOT, TTCFG, REF\_MSG\_x, TRIG\_CFG\_x, TT\_TRIG\_x и TT\_WTRIG\_x. Кроме того, логические ячейки, ответственные за TTCAN, также удаляются, что уменьшает потребление площади ядром.

## 15.4 Временная метка CiA 603

### 15.4.1 Временная метка

CAN в автоматизации (CiA) определяет в спецификации 603 метод для временной метки с по меньшей мере 16 битами, который опционально поддерживается CAN-CTRL. Его можно использовать в дополнение к TTCAN или отдельно.

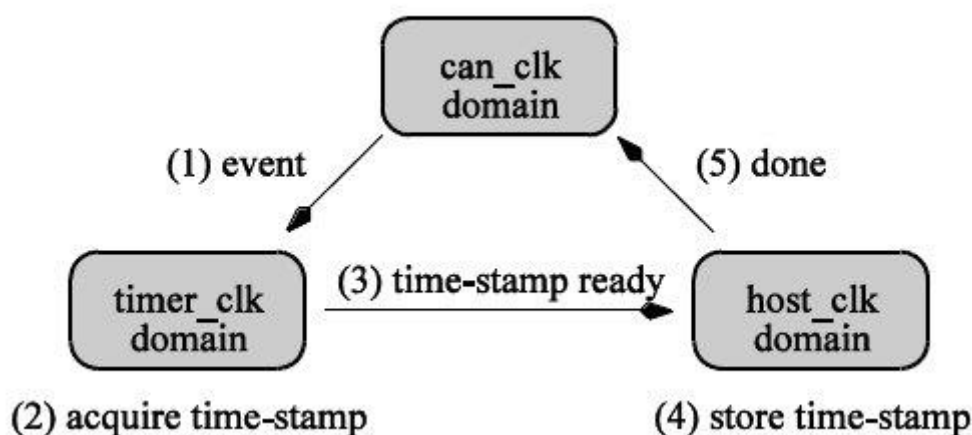
Для временных меток (time-stamping) CiA-603 используется внешний 32-х битный таймер. Таймер – счётчик работающий на частоте SYSCLK. Для получения значения счётчика и его установки используются регистры CAN\_TSCDR и CAN\_TSCSR. Запуск и остановка счётчика производится через регистр CAN\_TSCCR.

Основная концепция CiA 603 заключается в том, чтобы иметь таймер свободного хода, который подсчитывает тактовые циклы, а не время битов CAN. Точность должна составлять не менее 10 мкс (16 бит) или 1 мкс (32 бита или более). Для CAN-CTRL требуется внешний таймер свободного запуска (до 64 бит), подключенный к входному timer\_in и работающий с timer\_clk. Неиспользуемые биты timer\_in должны быть зафиксированы равными 0.

Параметр clock timer\_clk может быть

- Асинхронный для can\_clk и host\_clk,
- Синхронный с can\_clk,
- Синхронный с host\_clk,
- Синхронный с can\_clk и host\_clk.

CDC используется по мере необходимости. Поэтому, если timer\_clk не синхронизирован с can\_clk, то полученные временные метки зависят от соотношения этих обоих тактовых сигналов, и будет задержка из-за CDC.



### Рисунок 15.15. Временная метка и CDC

Как показано выше на Рисунок 15.15, после события делается копия таймера. Это временная метка времени. Эта временная метка сохраняется в RTS или TTS соответственно, где она может быть прочитана хостом. После того, как он сохранен, сигнал квитирования сигнализирует о завершении действия. Это описывает механизм CDC с 3 частотными доменами.

Временные метки получают в точке выборки SOF или бита EOF, где кадр считается действительным. Это может быть выбрано с помощью конфигурационного бита TIMEPOS. Семь рецессивных битов после разделителя ACK формируют EOF кадра CAN / CAN FD. Кадр становится действительным для приемника при предпоследнем бите EOF, но для передатчика при последнем бите EOF.

Временная метка на основе программного обеспечения, которая широко используется во многих системах, основана на прерывании приема и передачи. Поэтому рекомендуется указывать время в EOF.

Предполагается, что CiA 603 поддерживает временную метку и синхронизацию времени AUTOSAR. Для AUTOSAR один узел в сети CAN является ведущим по времени. Мастер времени передает сообщение синхронизации (SYNC message). Временная метка сообщения SYNC обрабатывается ведущим устройством времени и всеми подчиненными устройствами времени. Разница во времени между событием отправки сообщения SYNC и моментом фактической передачи сообщения будет передана в последующем сообщении (FUP) мастером времени. Поэтому CAN-CTRL поддерживает только одну временную метку для переданных кадров (TTS), но отдельные временные метки для всех принятых кадров (RTS). Генерация временной метки для передаваемых кадров может быть включена или отключена для каждого кадра индивидуально с помощью бита TTSEN внутри слота TBUF.

CiA определяет правила для считывания таймера, а также для изменения таймера. CAN-CTRL не включает таймер, но использует внешний таймер. CAN-CTRL включает в себя только механизм временной метки, регистр для хранения TTS и память для хранения RTS для каждого кадра.

Регистровый бит TIMEEN включает или отключает временную метку. Если отключены TTS и RTS, то они содержат недостоверные значения.

#### 15.4.2 Внешний таймер

Для временной метки CiA 603 необходимо использовать внешний таймер. Этот таймер не включен в CAN-CTRL, поскольку может оказаться, что такой таймер уже существует в системе, который может быть повторно использован для временной отметки CiA 603.



Внешний таймер должен быть, по крайней мере, счетчиком свободного хода, который увеличивается с `timer_clk`. Такой очень простой таймер также включен в набор тестов. Разрядность таймера можно свободно выбирать. CAN-CTRL сохраняет либо 32, либо 64 бита при получении временной метки. Если в таймере меньше битов, неиспользуемые биты должны быть заполнены нулями.

Таймер для временной метки CiA 603 должен иметь точность не менее 10 мкс (16 бит) или 1 мкс (32 бита или более) и не более 1нс. Он основан на физической секунде, а не на битовом времени CAN. Таймер должен начинаться с нуля и непрерывно отсчитываться вверх.

CiA 603 определяет предделитель для таймера для компенсации различных уровней точности от разных узлов в сети. В качестве альтернативы и замены предделителя для CAN-CTRL предлагается использовать 64-разрядный таймер и при необходимости выполнить точную адаптацию с помощью программного обеспечения.

CiA 603 определяет возможность изменения значения таймера. Если модификация возможна, то она должна быть произведена как можно быстрее или немедленно в идеальном случае. Если таймер не находится в том же тактовом домене, что и хост-контроллер, то модификация может занять некоторое время, но такая задержка должна быть сведена к минимуму.

CiA 603 требует возможности считывания таймера в любое время.

Таким образом, такой таймер представляет собой счетчик свободного хода, который может считываться и записываться хостом, в то время как такой доступ может включать некоторые задержки из-за пересечения домена синхронизации. Такой таймер вместе с меткой времени CiA 603, обеспечиваемой CAN-CTRL, поддерживает синхронизацию времени, как определено AUTOSAR.

### 15.4.3 Ширина бита таймера

Временная метка CiA 603 фокусируется на поддержке AUTOSAR. AUTOSAR обрабатывает синхронизацию времени, и поэтому временные метки должны быть преобразованы в реальное время. С учетом рекомендации о том, что точность должна составлять не менее 10 мкс (16 бит) или 1 мкс (32 бита или более), ширина бита в 32 бита достаточна для хорошей поддержки AUTOSAR. Следует отметить, что преобразование в реальное время приводит к некоторым вычислительным затратам, которые должны выполняться главным процессором, но это часть концепции AUTOSAR.

Использование разрядности 64 бита может быть предпочтительнее в пользовательских приложениях, где временная метка может быть использована немедленно в режиме реального времени без преобразования. 64 бит хватает на многие тысячи лет без переполнения. Такие пользовательские приложения должны учитывать тот факт, что все узлы в сети должны вести подсчет с одинаковой частотой `<timer_clk>` или в противном



случае необходимо учитывать коэффициенты тактовой частоты, если для вычислений используются временные метки других узлов.

## 15.5 Длительность бита CAN

### 15.5.1 Скорости передачи данных в битах

CAN 2.0B определяет скорость передачи данных до 1 Мбит/с. Для CAN FD нет фиксированного ограничения. Для реальных систем скорость передачи данных ограничена используемым приемопередатчиком и достижимой тактовой частотой для ядра CAN-CTRL, которая зависит от используемой библиотеки целевых ячеек.

Ядро CAN-CTRL может быть запрограммировано на произвольно выбранные скорости передачи данных, ограниченные только диапазоном битовых настроек в соответствующих регистрах синхронизации битов и предварительного набора.

### 15.5.2 Определения

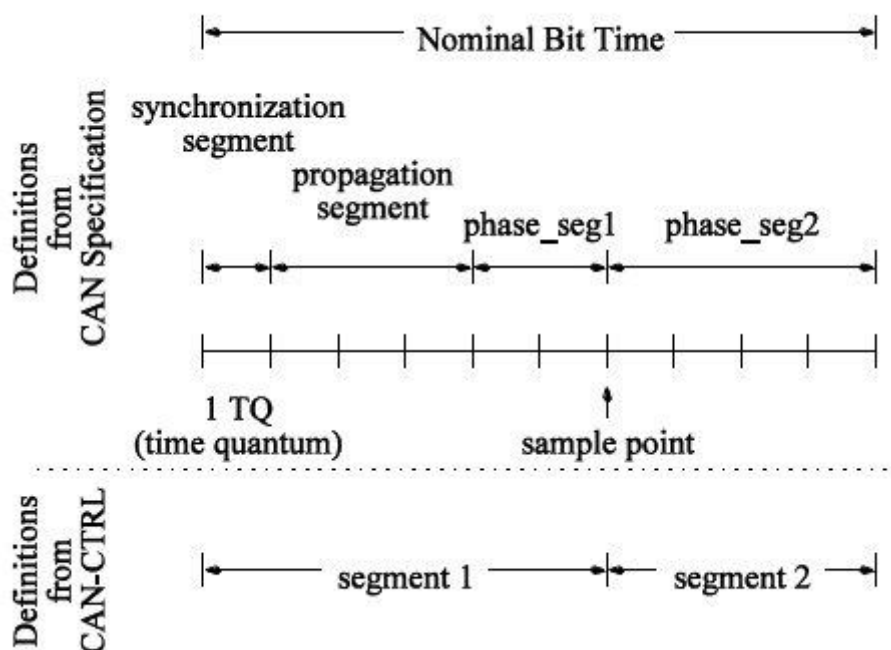


Рисунок 15.16. Спецификации синхронизации битов CAN

ВТ длительности бита CAN состоит из нескольких сегментов, как показано выше на Рисунок 15.16.

Каждый сегмент состоит из нескольких единиц временных квантов  $n \cdot TQ$ . Длительность квантов времени  $TQ$  равна:

$$TQ = \frac{n_{prescaler}}{f_{CLOCK}}$$

Значения  $n_{TQ}$  и  $n_{prescaler}$  должны выбираться в зависимости от тактовой частоты системы  $f_{CLOCK}$ , чтобы соответствовать  $BT_{real}$  как можно ближе к  $BT_{ideal} = 1/BR$ , где  $BR$  - скорость передачи данных по шине CAN в бодах.:

$$BT_{ideal} \approx BT_{real} = \frac{n_{prescaler} \cdot n_{TQ}}{f_{CLOCK}} = t_{Seg\_1} + t_{Seg\_2}$$

Спецификация CAN требует нескольких соотношений между длинами сегментов (Таблица 15.56), что приводит к соотношениям между  $t_{Seg\_1}$ ,  $t_{Seg\_2}$  и максимальной шириной перехода синхронизации  $t_{SJW}$ .

Пожалуйста, обратите внимание, что ниже в Таблица 15.56 перечислены минимальные диапазоны конфигурации, определенные спецификацией CAN / CAN FD.

**Таблица 15.56. Сегменты синхронизации CAN (Минимальные диапазоны конфигурации)**

Сегмент	Сброс		
SYNC_SEG	Сегмент синхронизации = 1 TQ		
PROP_SEG	Сегмент распространения [1...8] TQ [1...48] TQ [1...48] TQ [0...8] TQ	Скорость передачи данных CAN 2.0 Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	CAN FD не включен CAN FD включен
PHASE_SEG1	Сегмент фазового буфера 1 [1...8] TQ [1...16] TQ [1...16] TQ [1...8] TQ	Скорость передачи данных CAN 2.0 Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	CAN FD не включен CAN FD включен

Сегмент	Сброс		
PHASE_SEG2	Сегмент фазового буфера 2 [2...8] TQ [2...16] TQ [2...16] TQ [2...8] TQ	Скорость передачи данных CAN 2.0 Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	CAN FD не включен CAN FD включен
SJW	Ширина скачка синхронизации [1...4] TQ [1...16] TQ [1...16] TQ [1...8] TQ	Скорость передачи данных CAN 2.0 Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	CAN FD не включен CAN FD включен
IPT	Время обработки информации = [0...2] TQ PHASE_SEG2 ≥ IPT		

Как видно выше на Рисунок 15.16, ядро CAN-CTRL собирает SYNC\_SEG, PROP\_SEG и PHASE\_SEG1 в одну группу, а длина группы настраивается с помощью  $t_{Seg\_1}$ .

Ниже в Таблица 15.57 перечислены доступные диапазоны конфигурации.

Пожалуйста, обратите внимание, что ядро CAN-CTRL не проверяет, соблюдены ли все правила, и предлагает более широкий диапазон конфигурации, чем определено спецификацией CAN / CAN FD.

**Таблица 15.57. Настройки синхронизации CAN-CTRL (Доступные диапазоны конфигурации)**

Настройка	Требования			
$t_{Seg\_1}$	[2...65] TQ [2...65] TQ [2...17] TQ		Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	(медленный) (медленный) (быстрый)

Настройка	Требования			
$t_{Seg\_2}$	[1...8] TQ [1...32] TQ [1...8] TQ	$t_{Seg\_1} \geq t_{Seg\_2} + 2$ $t_{Seg\_1} \geq t_{Seg\_2} + 2$ $t_{Seg\_1} \geq t_{Seg\_2} + 1$	Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	(медленный) (медленный) (быстрый)
$t_{SJW}$	[1...16] TQ [1...16] TQ [1...8] TQ	$t_{Seg\_2} \geq t_{SJW}$ $t_{Seg\_2} \geq t_{SJW}$ $t_{Seg\_2} \geq t_{SJW}$	Скорость передачи данных CAN 2.0 Номинальная скорость передачи данных CAN FD Скорость передачи данных CAN FD	(медленный) (медленный) (быстрый)

Для скорости передачи данных CAN 2.0 и номинальной скорости передачи данных CAN FD (низкий) настройки регистра S\_Seg\_1, S\_Seg\_2, S\_SJW и S\_PRESC определяют соответствующие длины сегментов.

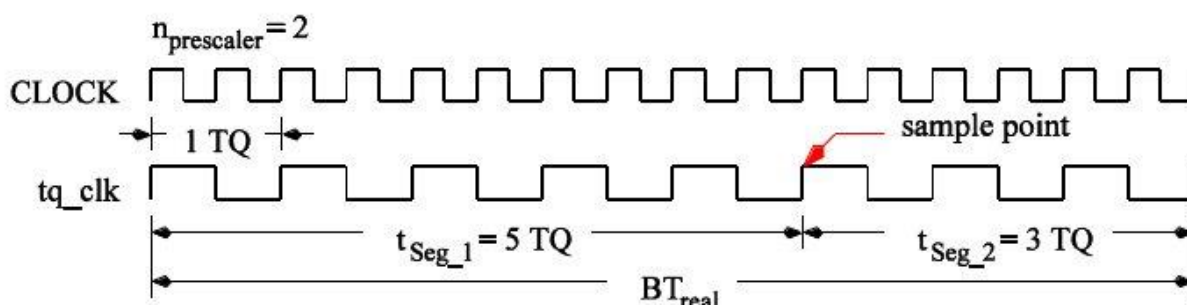
Для скорости передачи данных CAN FD (быстрый) допустимы регистры F\_Seg\_1, F\_Seg\_2, F\_SJW и F\_PRESC.

$t_{Seg\_1} = (S\_Seg\_1 + 2) \cdot TQ$	$t_{Seg\_1} = (F\_Seg\_1 + 2) \cdot TQ$
$t_{Seg\_2} = (S\_Seg\_2 + 1) \cdot TQ$	$t_{Seg\_2} = (F\_Seg\_2 + 1) \cdot TQ$
$t_{SJW} = (S\_SJW + 1) \cdot TQ$	$t_{SJW} = (F\_SJW + 1) \cdot TQ$
$n_{prescaler} = S\_PRESC + 1$	$n_{prescaler} = F\_PRESC + 1$

Ядро CAN FD переключается с низкой номинальной скорости передачи данных на быструю скорость передачи данных в точке выборки в бите BRS и переключается обратно в точке выборки в бите CRC-разделителя.

Иллюстрация подходящей конфигурации скорости передачи данных приведена ниже на рисунке Рисунок 15.17, где  $f_{iq\_clk} = f_{CLOCK} \div n_{prescaler}$ .

Битовое время  $BT_{real}$ , точка выборки и ширина скачка синхронизации SJW будут получены из  $tq\_clk$ .



**Рисунок 15.17. Тактовое деление для выборки битов**

Учитывая требования из таблицы Таблица 15.57, хост-контроллер должен определить длину сегмента 1, сегмента 2 и ширину скачка синхронизации для медленной скорости передачи данных и, если требуется, для CAN FD для быстрой скорости передачи данных.

Некоторые предложения, приведенные в качестве эмпирических правил:

- Сегмент 1 должен быть немного больше, чем сегмент 2. Тогда точка выборки находится немного позже, чем в середине длительности бита.
- Ширина скачка синхронизации (SJW) не должна быть больше сегмента 2. Если SJW слишком мала, узел CAN может быть слишком медленным для повторной синхронизации, если SJW слишком велика, то узел CAN может повторять синхронизацию слишком часто. Длина SJW в два раза меньше длины сегмента 2, по-видимому, является подходящим значением.
- Все узлы CAN, подключенные к шине CAN, должны по возможности выбирать аналогичные настройки.

Самая высокая скорость шины CAN может быть настроена с использованием минимальных значений синхронизации. Но есть некоторые вещи, которые необходимо учитывать:

- Если делители больше единицы, то все остальные параметры синхронизации могут быть установлены на ноль, но это нарушает правило  $t_{Seg\_1} \geq t_{Seg\_2} + 2$  (медленная скорость) соответственно  $t_{Seg\_1} \geq t_{Seg\_2} + 1$  (высокая скорость), и поэтому  $S\_Seg\_1$  и  $F\_Seg\_1$  должны быть установлены как минимум на 1. Но такой выбор временных параметров работает в теории, но может быть недостаточно надежным для реальных сетей.
- Если для делителей установлено значение один, то синхронизация становится затруднительной. В целом, спецификация CAN требует, чтобы время передачи одного бита составляло не менее 8 TQ для CAN 2.0 и CAN FD с низкой номинальной

скоростью передачи данных. Для CAN-CTRL одно битное время быстрой скорости передачи данных также должно составлять не менее 8 ТК, если для быстрого предварительного вычисления установлено значение 1.

- Таким образом, самая быстрая частота шины CAN - это частота `can_clk`, деленная на 8: предварительно заданное значение 1 и 8 бит времени TQ.

### 15.5.3 Пример конфигурации

Этот пример относится к Рисунок 15.18. Это пример конфигурации CAN 2.0 / медленная скорость передачи данных, но все инструкции могут быть легко переведены в CAN FD (быстрая) скорость передачи данных.

Для настройки ядра CAN-CTRL необходимо выполнить следующие шаги:

1. Установите бит RESET=1.
2. Установите регистры S\_Seg\_1 и S\_Seg\_2:

В приведенном примере скорость передачи данных по шине  $f_{BUS} = 1$  Мбод, а системная тактовая частота составляет 16 МГц.

Значения  $n_{TQ}$  и  $n_{prescaler}$  должны быть выбраны таким образом, чтобы они соответствовали  $BT_{real}$  как можно ближе к  $BT_{ideal}$ .

В этом примере выбраны  $n_{prescaler} = 2$  и  $n_{TQ} = 8$ , что приводит к идеальному совпадению:

$$BT_{ideal} = BT_{real} = 8TQ.$$

Вместе с  $BT_{real} = t_{Seg\_1} + t_{Seg\_2}$  и определениями временных сегментов, приведенными в главе 15.5.1 "Скорости передачи данных в битах",  $t_{Seg\_1} = 5TQ$  и  $t_{Seg\_2} = 3TQ$  могут быть выбраны в качестве подходящих значений, что в конечном итоге приводит к настройкам регистра S\_Seg\_1=3 и S\_Seg\_2=2.

3. Загрузите регистры кода приема и маски (необязательно).

4. Установите регистр S\_SJW:

При  $t_{Seg\_2} \geq t_{sjw}$  можно свободно выбрать  $t_{sjw} = 3$ , что в конечном итоге приводит к 2.

5. Загрузите регистр тактового предделителя S\_PRESC:  $n_{prescaler} = PRESC + 1$  приводит к S\_PRESC=1.

6. Установите бит RESET=0.

Данный порядок не является обязательным. Просто необходимо установить бит RESET=1 в начале, так как в противном случае невозможно загрузить регистры синхронизации битов, ACODE и AMASK. RESET=0 требуется после завершения настройки. Затем контроллер ожидает 8 рецессивных битов (конец кадра) и возобновляет свою нормальную работу.

7. Продолжайте настройку прерываний другими битами конфигурации и выполняйте команды.

#### **15.5.4 Калькулятор расчёта временных параметров бита CAN (CBC)**

Пакет выпуска включает в себя инструмент программного обеспечения для MS Windows для вычисления временных параметров бита CAN. Этот инструмент можно использовать для расчета CAN 2.0, а также номинальной скорости передачи данных CAN FD и скорости передачи данных.

Начиная с настройки тактовой частоты ядра CAN-CTRL и желаемой скорости передачи данных в бодах, этот инструмент выводит все возможные настройки для предделителя, сегмента 1 и сегмента 2. Для SJW инструмент выдает максимальное значение. Кроме того, инструмент выводит положение точки выборки в виде коэффициента битового времени.

Инструмент выдает, с одной стороны, время, кратное TQ, а также значение регистра. Например, "t\_seg1=7" означает, что сегмент 1 имеет длину 7 TQ, и значение регистра должно быть установлено на S\_Seg\_1=5.

По умолчанию инструмент пытается рассчитать настройки с допуском 0%. Другими словами: при заданной тактовой частоте и этих настройках будет точно синтезирована желаемая скорость передачи данных в бодах. Можно изменить настройку допуска, чтобы получить также настройки, которые не будут точно совпадать. Пожалуйста, обратите внимание, что инструмент не проверяет, соответствует ли выбранный допуск спецификации CAN.

Этот инструмент выполняет исчерпывающий поиск возможных настроек. Время вычисления увеличится при высокой скорости передачи данных в бодах и большом значении допуска. Инструменты выполняют поиск настроек с шагом в 1 Бод.

В дополнение к инструменту программного обеспечения в пакет выпуска входит лист MS Excel. Этот лист может быть полезен, если CAN-CTRL необходимо адаптировать к существующей сети CAN, где конфигурация синхронизации битов уже исправлена. Этот лист Excel содержит параметры конфигурации, рекомендованные CAN в автоматизации (CiA).

### 15.5.5 Переключение скорости передачи битов и точки выборки

В сети CAN, когда используются кадры CAN FD с переключением скорости передачи данных, точное положение точки выборки важно и должно быть для всех узлов как можно более одинаковым.

Скорость передачи данных переключается после точки отсчета бита BRS и точки отсчета разделителя CRC. Поэтому длины этих битов являются промежуточными. Как видно ниже на рисунке Рисунок 15.18, положение точки выборки имеет большое значение для абсолютной длины бита BRS. (1 ТК при низкой номинальной скорости передачи данных может быть намного больше, чем несколько ТК при высокой скорости передачи данных.) Если скорость передачи данных намного выше номинальной скорости передачи данных, то более ранняя или более поздняя точка выборки может привести к ложным выборкам при высокой скорости передачи данных.

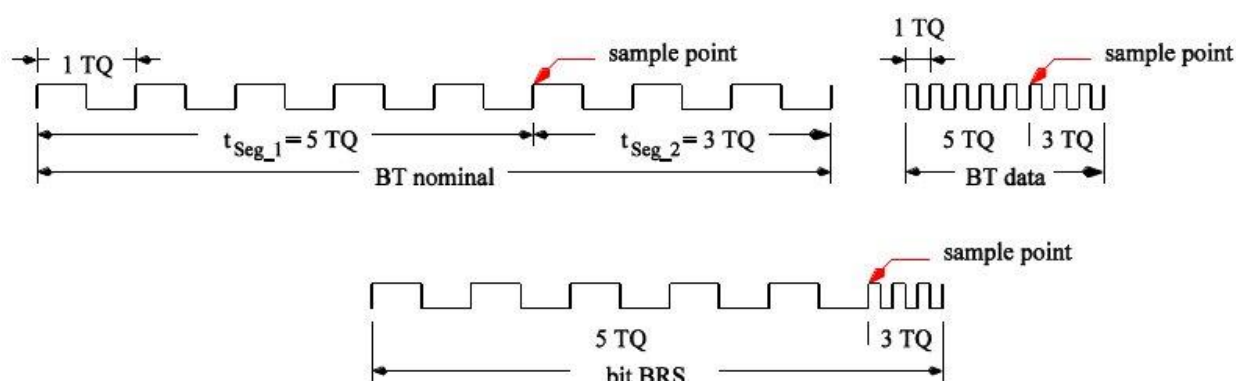


Рисунок 15.18. Переключение скорости передачи данных в бите BRS ( $S\_PRESC \neq F\_PRESC$ )

### 15.5.6 Конфигурация тактовой синхронизации для узлов CAN FD

Как указано выше в главе 15.5.5 "Переключение скорости передачи битов и точки выборки", положение точки выборки очень важно для переключения скорости передачи битов. Поэтому существует общая рекомендация для всех узлов CAN FD использовать точно такие же временные параметры в сети CAN. В отличие от классического CAN, где только скорость передачи данных должна быть одинаковой для всех узлов, CAN FD требует, чтобы точка выборки находилась в одном и том же положении. Другими словами, сегмент 1 и сегмент 2 должны быть настроены на одинаковую длину для всех узлов. Кроме того, рекомендуется использовать также одинаковую длину одного TQ, чтобы сделать синхронизацию для всех узлов равной.

Это требование может быть выполнено только в том случае, если все узлы работают с одинаковой базовой тактовой частотой (называемой `can_clk` для CAN-CTRL) или, по



крайней мере, с совместимыми тактовыми частотами. Рекомендуемые настройки - 20 МГц, 40 МГц или 80 МГц.

### 15.5.7 BMT и BMT

Для узлов CAN FD компенсация задержки передатчика (TDC) может быть дополнительно включена, в то время как компенсация задержки приемника (RDC) автоматически активирована. Эти особенности имеют следующую подоплеку: Для связи со скоростью передачи данных CAN FD может оказаться, что задержка передающего приемопередатчика или задержка шины больше, чем битовое время. Это должно быть компенсировано. Без TDC скорость передачи битов в фазе передачи данных кадров FD ограничена тем фактом, что передатчик обнаруживает битовую ошибку, если он не может принять свой собственный переданный бит, последний в точке выборки этого бита.

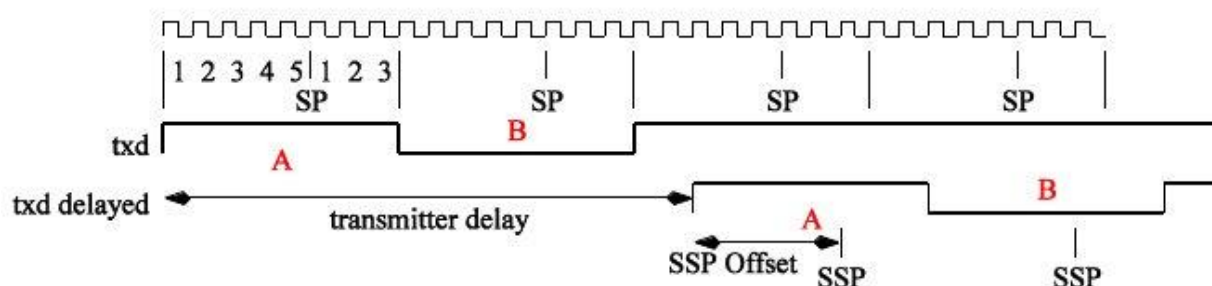


Рисунок 15.19. Задержка передатчика

На Рисунок 15.19 приведен пример эффекта большой задержки передатчика. На этом рисунке показан поток данных *txd*, начинающийся с битов А и В. Одно битовое время состоит из  $t_{Seg\_1} = 5TQ$  перед точкой выборки (SP) и  $t_{Seg\_2} = 3TQ$  позади. В этом примере задержка передатчика превышает 2 битовых раз. Поэтому исходный SP не может быть использован для выборки правильного значения бита, и спецификация CAN FD определяет дополнительную вторичную точку выборки (SSP). Если TDC включен с битом  $TDCEN=1$ , то CAN-CTRL автоматически определяет задержку передатчика. Положение SSP - это задержка передатчика плюс смещение SSP, которое определяется битами конфигурации SSPOFF. SSPOFF должен быть задан как число TQ, и предлагается установить  $t_{Seg\_1}$  равным SSPOFF. (Пожалуйста, помните, что  $F\_SEG\_1$  определяет  $t_{Seg\_1}$  при скорости передачи данных. Поэтому в примере, приведенном на Рисунок 15.19, выбрано значение  $SSPOFF=5$ , поскольку  $F\_SEG\_1=3$ .)

ISO 11898-1:2015 требует использовать только  $F\_PRESC=0$  или 1, если используется этот TDC. С учетом этого требования CAN-CTRL способен автоматически определять задержку передатчика, в 3 раза превышающую скорость передачи быстрых данных.

Задержка, превышающая битовое время, также должна учитываться для принимающих узлов. CAN FD определяет дополнительную жесткую синхронизацию на падающем фронте между битом  $FDF=1$  и следующим битом  $r0$ . (В это время активна низкая номинальная скорость передачи данных.) Синхронизация для CAN и CAN FD в общем случае

определяется для работы с временными шагами в один TQ. Но шаги в один TQ при номинальной скорости передачи данных могут быть слишком грубыми для синхронизации с высокой скоростью передачи данных. Поэтому дополнительная жесткая синхронизация в бите FDF должна нарушать это правило и должна синхронизироваться как можно точнее и ограничиваться только тактовой частотой системы. CAN-CTRL использует эту специальную синхронизацию, и это называется RDC. RDC автоматически выполняется во время приема, если  $FDF = 1$ , независимо от того, включен TDC или нет.

## 15.5.8 Рекомендации по тактовой синхронизации

Конфигурация синхронизации CAN FD требует использования одинаковой скорости передачи данных и одинаковой точки выборки для всех узлов в сети CAN. Поэтому рекомендуется использовать 20 МГц, 40 МГц или 80 МГц в качестве источника для машины протокола CAN (`can_clk`). В таблицах Таблица 15.59, Таблица 15.60, Таблица 15.61 приведены некоторые рекомендации по настройкам синхронизации битов, которые должны применяться ко всем узлам в сети CAN FD. Эти настройки являются рекомендациями из "CAN in Automation (CiA)".

Пожалуйста, обратите внимание, что время передачи битов должно составлять не менее 8 TQ, если для обеспечения стабильной связи для предварительного вызова установлено значение 1. Возможны более низкие настройки, но это может вызвать проблемы.

Решение о том, использовать TDC или нет, зависит от сети CAN. Таблицы дают только предположение, и TDC может быть включен или отключен по мере необходимости.

**Таблица 15.58. Аббревиатуры для таблиц "Рекомендации для `can_clk` с частотой 20 МГц", "Рекомендации для `can_clk` с частотой 40 МГц" и "Рекомендации для `can_clk` с частотой 80 МГц"**

Аббревиатура	Описание
PSP	Первичная точка выборки
SSP	Вторичная точка выборки
Seg 1	Сегмент 1
Seg 2	Сегмент 2
TDC	Компенсация задержки передатчика (см. SSPOFF)

**Таблица 15.59. Рекомендации для `can_clk` с частотой 20 МГц**

Скорость передачи битов [Mbit/s]	PSP [%]	SSP [%]	Предел итель	Длительность бита [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [clk]
0,5	80	(отключен)	1	40	32	8	8	-
0,5 (арбитраж)	90	-	1	40	32	8	8	-
0,25 (арбитраж)	80	-	1	80	64	16	16	-

Скорость передачи битов [Mbit/s]	PSP [%]	SSP [%]	Предел итель	Длительность бита [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [clk]
0,833	79	(отключен)	1	24	19	5	5	-
1	80	80	1	20	16	4	4	16
1,538	77	77	1	13	10	3	3	10
2	80	80	1	10	8	2	2	8
4	80	80	1	5	4	1	1	4
5	75	75	1	4	3	1	1	3

Таблица 15.60. Рекомендации для сап\_clk с частотой 40 МГц

Скорость передачи битов [Mbit/s]	PSP [%]	SSP [%]	Предел итель	Длительность бита [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [clk]
0,25 (арбитраж)	80	-	2	80	64	16	16	-
0,5 (arbitration)	80	-	1	80	64	16	16	-
0,5	80	(отключен)	2	40	32	8	8	-
0,833	79	(отключен)	2	24	19	5	5	-
1	80	80	1	40	32	8	8	32
1,538	77	77	1	26	20	6	6	20
2	80	80	1	20	16	4	4	16
3,077	77	77	1	26	20	3	3	10
4	80	80	1	10	8	2	2	8
5	75	75	1	8	6	2	2	6
6,667	83	83	1	6	5	1	1	5
8	80	80	1	5	4	1	1	4
10	75	75	1	4	3	1	1	3

Таблица 15.61. Рекомендации для сап\_clk с частотой 80 МГц

Скорость передачи битов [Mbit/s]	PSP [%]	SSP [%]	Предел итель	Длительность бита [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [clk]
0,25 (арбитраж)	80	-	4	80	64	16	16	-
0,5 (арбитраж)	80	-	2	80	64	16	16	-
0,5	80	(отключен)	4	40	32	8	8	-
0,833	79	(отключен)	4	24	19	5	5	-
1	80	80	2	40	32	8	8	64
1,538	77	77	2	26	20	6	6	40
2	80	80	2	20	16	4	4	32
3,077	77	77	2	13	10	3	3	20
4	80	80	1	20	16	4	4	16
5	75	75	1	16	12	4	4	12
6,667	83	83	1	12	10	2	2	10
8	80	80	1	10	8	2	2	8
10	75	75	1	8	6	2	2	6

## 15.6 Описание функциональных возможностей

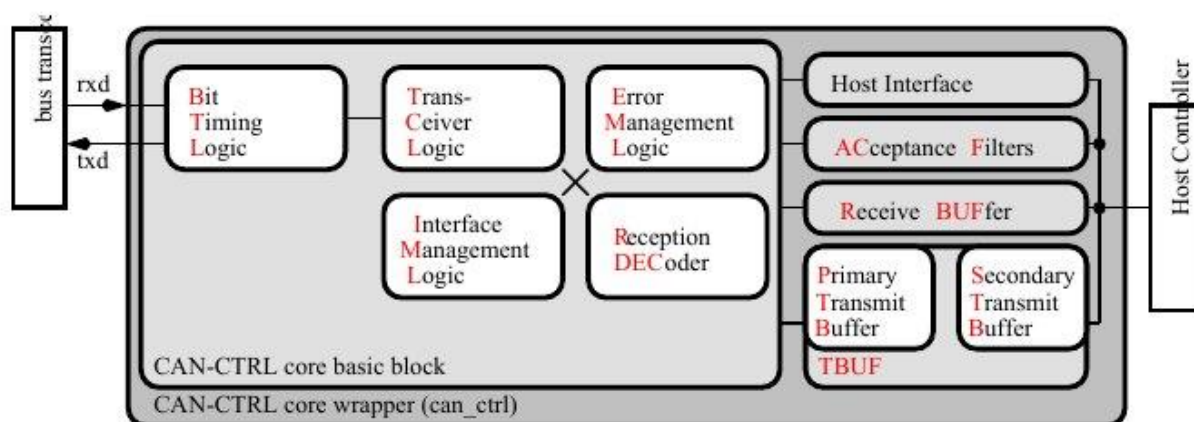


Рисунок 15.20. Функциональная схема

### 15.6.1 Базовый блок ядра CAN-CTRL

#### 15.6.1.1 Логика тактовой синхронизации (BTL)

BTL отслеживает шину и синхронизирует внутренние действия с потоком битов на шине CAN. Жесткая синхронизация выполняется, если в начале сообщения происходит переход линии шины “от рецессивного к доминантному”. Повторная синхронизация происходит при дальнейших переходах “от рецессивного к доминантному” во время приема сообщения. Временные регистры (предоставляемые host\_interface) управляют временными сегментами для компенсации времени задержки распространения и фазовых сдвигов и определения точки выборки.

BTL также включает в себя предделитель скорости передачи данных в бодах. Внешние системные тактовые сигналы будут разделены на запрограммированное значение. Результирующий период называется квантами времени (TQ).

#### 15.6.1.2 Логика приемопередатчика (TCL)

TCL состоит из машины состояний протокола. Он управляет всеми функциями приема и передачи кадров:

- проверка принятых битов и передача в буфер приема,
- генерация битов для передачи на шину (ID-, RTR-, IDE-, DLC-, Data-, CRC-, ...),
- арбитраж,
- обработка битных материалов,
- обработка ошибок,

- генерация кадров ошибок и перегрузки.

### **15.6.1.3      Логика управления интерфейсом (IML)**

Логика управления интерфейсом (IML) управляет поведением всего ядра в зависимости от команд (настройки регистра в интерфейсе хоста). IML генерирует адреса для буферов сообщений и предоставляет информацию о прерываниях и состоянии хост-контроллеру.

### **15.6.1.4      Логика управления ошибками (EML)**

EML состоит из счетчиков ошибок приема и передачи. Счетчики управляются в зависимости от соответствующего типа ошибок (Битовая ошибка, Ошибка заполнения, Ошибка формы, Ошибка CRC и Ошибка подтверждения).

### **15.6.1.5      Декодер приема (RDEC)**

RDEC сохраняет и декодирует биты заголовка принятых сообщений. Эта информация необходима "на лету" для обработки протокола и позже сохраняется в RBUF.

## **15.6.2    Оболочка ядра CAN-CTRL (can\_ctrl)**

### **15.6.2.1      Интерфейс хоста**

Интерфейс хоста включает в себя биты регистра управления и состояния. Сюда входит карта регистров, за исключением RBUF, TBUF и ACF.

### **15.6.2.2      Буфер приема (RBUF)**

RBUF содержит память для приемного буфера. Количество слотов RBUF настраивается перед синтезом с помощью общего параметра.

### **15.6.2.3      Буфер передачи (TBUF)**

TBUF состоит из Первичного буфера передачи PTB и Вторичного буфера передачи STB. PTB имеет более высокий приоритет, чем STB. Передача PTB приостанавливает передачу STB и выполняется первой. Правила протокола CAN требуют, чтобы передача STB, записанная на шину CAN, не прерывалась. Следовательно, передача PTB начнется позже (после межкадрового пространства).

Количество слотов STB настраивается перед синтезом с помощью общего параметра.

### **15.6.2.4      Приемные фильтры (ACF)**

ACF состоит из управляющей логики для приемной фильтрации и блоков памяти фильтров.

Количество приемных фильтров ACF настраивается перед синтезом с помощью общего параметра от 1 до 16 фильтров.

### 15.6.3 Буферная память

Ядро CAN-CTRL использует память для RBUF, PTB, STB и ACF. Исходный код HDL включает в себя настраиваемую модель памяти, используемую всеми этими компонентами. Эта модель памяти поддерживает FPGA BlockRAM или распределенную оперативную память, которые могут быть выбраны с помощью параметров синтеза индивидуально для каждого компонента.

Для синтеза ASIC обе поддерживаемые модели памяти будут синтезироваться до триггеров, что довольно трудоемко и неэффективно для больших объемов памяти. Для этой цели предлагается заменить исходный файл HDL, содержащий эти модели памяти (<can\_memory.v(hd)>), другим файлом, создающим экземпляр предпочтительного IP-ядра RAM.

## 16. VTU: многофункциональные блоки таймеров

В микросхеме присутствуют два блока таймеров.

### 16.1 Основные особенности

- четыре независимых 16-ти битных таймера;
- 8-ми битный prescaler для каждого из таймеров;
- два 8-ми разрядных генератора ШИМ сигналов;
- 16-ти разрядный генератор ШИМ сигнала;
- 16-ти разрядный счетчик с двумя каналами захвата.

### 16.2 Конфигурация

Конфигурационные параметры контроллеров доступны по ссылкам:

VTU0, VTU1 – TBD.

## 17. PWM: ШИРОТНО-ИМПУЛЬСНЫЙ МОДУЛЯТОР

### 17.1 Общая информация

Широтно-импульсный модулятор PWM (Pulse Width Modulator) генерирует выходные импульсы сложной формы при минимальном участии CPU. Блок отличается высокой гибкостью работы и простотой программирования.

Блок PWM состоит из 4-х каналов PWM\_CH.

Каждый канал PWM\_CH имеет следующие особенности:

- 32-битный специализированный счетчик с возможностью контроля периода, частоты и направления счета;
- два независимых выхода сформированных сигналов OUTA и OUTB;
- асинхронный программный контроль выходных сигналов;
- программное управление сдвигом фазы выходных сигналов относительно фазы выходных сигналов других каналов PWM;
- синхронизация относительно других каналов во время пуска и во время работы;
- возможность генерация запретной зоны от различных фронтов с различным периодом;
- контроль выходных сигналов по внешним входам TU;
- возможность наложения высокочастотного сигнала на выходы OUTA и OUTB;
- управление формированием запроса на прерывание CPU.

### 17.2 Регистры

Базовый адрес индивидуальных регистров каналов PWM:

- PWM\_CH0\_OFFSET = 0x0011\_1000;
- PWM\_CH1\_OFFSET = 0x0011\_1100;
- PWM\_CH2\_OFFSET = 0x0011\_1200;
- PWM\_CH3\_OFFSET = 0x0011\_1300;

Базовый адрес индивидуальных регистров каналов PWM:

- PWM\_GLOBAL\_OFFSET = 0x0011\_1X00;

Адрес индивидуального регистра вычисляется по формуле:

$$\text{ADDR\_REG} = \text{PWM\_CH*\_OFFSET} + \text{REG\_OFFSET}$$



Адрес глобального регистра вычисляется по формуле:

$$\text{ADDR\_REG} = \text{PWM\_GLOBAL\_OFFSET} + \text{REG\_OFFSET}$$

Перечень регистров приведен в Таблица 17.1.

**Таблица 17.1. Перечень программно-доступных регистров**

Условное обозначение	Название регистра	Исходное состояние	Тип доступа	Смещение
CTRRUN*	Регистр управления пуском/остановом таймера	0x00000000	R/W	0x80
CTRSTS*	Регистр статуса счетчика	0x01010101	R/W	0x84
CLKCTL	Регистр управления параметрами блока Count unit	0x0000003A	R/W	0x08
CTRPHS	Регистр фазы синхронизации блока Count unit	0x00000000	R/W	0x0C
CTRPRD	Регистр периода счета блока Count unit	0x00000000	R/W	0x10
CTRCNT	Регистр счетчика блока Count unit	0x00000000	R/W	0x14
CMPCCTL	Регистр управления параметрами блока сравнения Compare unit	0x00000000	R/W	0x20
CMPCA	Регистр сравнения CMPCA блока сравнения Compare unit	0x00000000	R/W	0x24
CMPCB	Регистр сравнения CMPCB блока сравнения Compare unit	0x00000000	R/W	0x28
EMCTLA	Управляющий регистр выхода OUTA блока Event manager	0x00000000	R/W	0x2C
EMCTLB	Управляющий регистр выхода OUTB блока Event manager	0x00000000	R/W	0x30
EMSWFR	Регистр программного управления выходами OUTA/OUTB с однократным действием блока Event manager	0x00000000	R/W	0x34
EMCSWFR	Регистр программного управления выходами OUTA/OUTB с продолжительным действием блока Event manager	0x00000000	R/W	0x38
DZCTL	Регистр управления блока Dead zone	0x00000000	R/W	0x40
DZRPFR	Регистр периода задержки после переднего фронта блока Dead zone	0x00000000	R/W	0x44
DXFPFR	Регистр периода задержки после заднего фронта блока Dead zone	0x00000000	R/W	0x48
CHCTL	Регистр управления блока Chopper	0x00000000	R/W	0x4C
TUSEL	Регистр выбора принципа работы блока Trip unit	0x00000000	R/W	0x50
TUCTL	Регистр определения реакции на событие блока Trip unit	0x00000000	R/W	0x54
TUINTM	Регистр маски прерывания блока Trip unit	0x00000000	R/W	0x58
TUSTS	Регистр статуса блока Trip unit	0x00000000	R	0x5C
TUCLR	Регистр очистки битов статуса блока Trip unit	0x00000000	R/W	0x60
TUFRC	Регистр программного формирования событий блока Trip unit	0x00000000	R/W	0x64
ICSEL	Регистр выбора принципа работы блока Interrupt control	0x00000000	R/W	0x68
ICCTL	Регистр определения реакции на событие блока Interrupt control	0x00000000	R/W	0x6C
ICSTS	Регистр статуса блока Interrupt control	0x00000000	R	0x70
ICCLR	Регистр очистки битов статуса блока Interrupt control	0x00000000	R/W	0x74
ICFRC	Регистр программного формирования событий блока Interrupt control	0x00000000	R/W	0x78

“\*” помечены глобальные регистры

## 17.2.1 Глобальные регистры

Особенностью глобальных регистров является то, что с их помощью осуществляется управление одновременно всеми каналами PWM.

### 17.2.1.1 Регистр управления пуском/остановом таймеров CRTRUN

Регистр CTRRUN (Таблица 17.2) используется для управления пуском/остановом таймера и режимом работы предделителя частоты всех каналов.

**Таблица 17.2. Регистр CTRRUN**

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:30]	-			
[29]	PRESMODE3	Управление режимом работы предделителя канала PWM_CH3 0 — предделитель формирует частоту только при включенном таймере (RUN3==2'b1x) 1 — предделитель формирует частоту не зависимо от RUN3. Генерация тактового сигнала начинается сразу после записи «1»	R/W	0
[28]	PRESRST3	Управление состоянием предделителя канала PWM_CH3 в момент простоя (PRESMODE3==0 и RUNSTS3==1'b0) 0 — счетчик предделителя сбрасывается в «0» 1 — счетчик предделителя сохраняет состояние на момент останова	R/W	0
[27:26]	-			
[25:24]	RUN3	Управление пуском/остановкой канала PWM_CH3 00b — остановка после следующего переключения счетчика CTRCNT 01b — остановка при совершении следующих событий: up-count режим: остановка при CTRCNT==CTRPRD down-count режим: остановка при CTRCNT==0 up-down-count режим: остановка при CTRCNT==0 1xb — пуск	R/W	0
[23:22]	-			
[21]	PRESMODE2	Управление режимом работы предделителя канала PWM_CH2 0 — предделитель формирует частоту только при включенном таймере (RUN2==2'b1x) 1 — предделитель формирует частоту не зависимо от RUN2. Генерация тактового сигнала начинается сразу после записи «1»	R/W	0
[20]	PRESRST2	Управление состоянием предделителя канала PWM_CH2 в момент простоя (PRESMODE2==0 и RUNSTS2==1'b0) 0 — счетчик предделителя сбрасывается в «0» 1 — счетчик предделителя сохраняет состояние на момент останова	R/W	0
[19:18]	-			

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[17:16]	RUN2	Управление пуском/остановкой канала PWM_CH2 00b — остановка после следующего переключения счетчика CTRCNT 01b — остановка при совершении следующих событий: up-count режим: остановка при CTRCNT==CTRPRD down-count режим: остановка при CTRCNT==0 up-down-count режим: остановка при CTRCNT==0 1xb — пуск	R/W	0
[15:14]	-			
[13]	PRESMODE1	Управление режимом работы предделителя канала PWM_CH1 0 — предделитель формирует частоту только при включенном таймере (RUN1==2'b1x) 1 — предделитель формирует частоту не зависимо от RUN1. Генерация тактового сигнала начинается сразу после записи «1»	R/W	0
[12]	PRESRST1	Управление состоянием предделителя канала PWM_CH1 в момент простоя (PRESMODE1==0 и RUNSTS1==1'b0) 0 — счетчик предделителя сбрасывается в «0» 1 — счетчик предделителя сохраняет состояние на момент останова	R/W	0
[11:10]	-			
[9:8]	RUN1	Управление пуском/остановкой канала PWM_CH1 00b — остановка после следующего переключения счетчика CTRCNT 01b — остановка при совершении следующих событий: up-count режим: остановка при CTRCNT==CTRPRD down-count режим: остановка при CTRCNT==0 up-down-count режим: остановка при CTRCNT==0 1xb — пуск	R/W	0
[7:6]	-			
[5]	PRESMODE0	Управление режимом работы предделителя канала PWM_CH0 0 — предделитель формирует частоту только при включенном таймере (RUN0==2'b1x) 1 — предделитель формирует частоту не зависимо от RUN0. Генерация тактового сигнала начинается сразу после записи «1»	R/W	0
[4]	PRESRST0	Управление состоянием предделителя канала PWM_CH0 в момент простоя (PRESMODE0==0 и RUNSTS0==1'b0) 0 — счетчик предделителя сбрасывается в «0» 1 — счетчик предделителя сохраняет состояние на момент останова	R/W	0
[3:2]	-			

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[1:0]	RUN0	Управление пуском/остановкой канала PWM_CH0 00b — остановка после следующего переключения счетчика CTRCNT 01b — остановка при совершении следующих событий: up-count режим: остановка при CTRCNT==CTRPRD down-count режим: остановка при CTRCNT==0 up-down-count режим: остановка при CTRCNT==0 1xb — пуск	R/W	0

### 17.2.1.2 Регистр статуса счетчиков CTRSTS

Регистр CTRSTS (Таблица 17.3) отображает текущее состояние подсистемы Count.

**Таблица 17.3. Регистр CTRSTS**

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:27]	-			
[26]	RUN_STS3	Бит отображающий состояние канала PWM_CH3 0 — счетчик остановлен 1 — счетчик работает	R	0
[25]	SYNC_STS3	Бит статуса входа синхронизации SYNCI канала PWM_CH3 0 — запись нуля не имеет эффекта, чтение нуля говорит об отсутствии внешней синхронизации 1 — чтение единицы говорит о том, что произошла синхронизация, запись единицы очищает данный бит	R/W	0
[24]	DIR_STS3	Отображает текущее направление счета CTRCNT канала PWM_CH3. Этот бит не несет никакой информации пока таймер не запущен. 0 — счет осуществляется вниз 1 — счет осуществляется вверх	R	1
[23:19]	-			
[18]	RUN_STS2	Бит отображающий состояние канала PWM_CH2 0 — счетчик остановлен 1 — счетчик работает	R	0
[17]	SYNC_STS2	Бит статуса входа синхронизации SYNCI канала PWM_CH2 0 — запись нуля не имеет эффекта, чтение нуля говорит об отсутствии внешней синхронизации 1 — чтение единицы говорит о том, что произошла синхронизация, запись единицы очищает данный бит	R/W	0
[16]	DIR_STS2	Отображает текущее направление счета CTRCNT канала PWM_CH2. Этот бит не несет никакой информации, пока таймер не запущен. 0 — счет осуществляется вниз 1 — счет осуществляется вверх	R	1
[15:11]	-			
[10]	RUN_STS1	Бит, отображающий состояние канала PWM_CH1 0 — счетчик остановлен 1 — счетчик работает	R	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[9]	SYNC_STS1	Бит статуса входа синхронизации SYNCI канала PWM_CH1 0 — запись нуля не имеет эффекта, чтение нуля говорит об отсутствии внешней синхронизации 1 — чтение единицы говорит о том, что произошла синхронизация, запись единицы очищает данный бит	R/W	0
[8]	DIR_STS1	Отображает текущее направление счета CTCNT канала PWM_CH1. Этот бит не несет никакой информации, пока таймер не запущен. 0 — счет осуществляется вниз 1 — счет осуществляется вверх	R	1
[7:3]	-			
[2]	RUN_STS0	Бит отображающий состояние канала PWM_CH0 0 — счетчик остановлен 1 — счетчик работает	R	0
[1]	SYNC_STS0	Бит статуса входа синхронизации SYNCI канала PWM_CH0 0 — запись нуля не имеет эффекта, чтение нуля говорит об отсутствии внешней синхронизации 1 — чтение единицы говорит о том, что произошла синхронизация, запись единицы очищает данный бит	R/W	0
[0]	DIR_STS0	Отображает текущее направление счета CTCNT канала PWM_CH0. Этот бит не несет никакой информации, пока таймер не запущен. 0 — счет осуществляется вниз 1 — счет осуществляется вверх	R	1

## 17.2.2 Регистры блока Count unit

### 17.2.2.1 Регистр управления подсистемой синхронизации CLKCTL (Таблица 17.4)

Таблица 17.4. Регистр CLKCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:20]	-			
[19:17]	DIVMUX	Управление мультиплексором делителя частоты 2: 0h — частота от предделителя делится на 1; 1h — частота от предделителя делится на 2; 2h — частота от предделителя делится на 4; 3h — частота от предделителя делится на 8; 4h — частота от предделителя делится на 16; 5h — используется внешняя частота PWM_CLK <b>Важно!</b> Менять значение возможно только при выключенном предделителе CLKCTL[PRESLEN] == 0	R/W	0
[16]	SYNCRST	Бит разрешения сброса предделителя при возникновении событий SYNCI или SWFSYNC 0 — сброс запрещен 1 — сброс разрешен	R/W	0
[15:8]	PRESPRD	Значение предделителя частоты CLK	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[7]	DIRSYNC	Направление счета после синхронизации. Этот бит используется, только когда счетчик работает в up-down режиме. 0 — после синхронизации счетчик декрементируется; 1 — после синхронизации счетчик инкрементируется	R/W	0
[6]	SWFSYNC	Программный импульс синхронизации. 0 — запись нуля не дает никакого эффекта, при чтении всегда возвращается 0; 1 — запись 1 провоцирует формирование импульса синхронизации, который по ИЛИ соединяется со входом SYNCI. SWFSYNC	R/W	0
[5:4]	SYNCOSEL	Выбор источника выходного сигнала SYNCO: 0h — SYNCI; 1h — счетчик равен нулю (CTRCNT==0); 2h — счетчик равен значению регистра сравнения В (CTRCNT==CMPB); 3h — SYNCO отключен	R/W	3h
[3]	LOADPRD	Управление моментом переписи данных из теневого регистра периода в активный 0 — регистр периода (CTRPRD) загружается из теневого регистра, когда счетчик (CTRCNT) равен нулю (запись или чтение CTRPRD осуществляется через теневой регистр); 1 — CTRPRD загружается напрямую без использования теневого регистра (запись или чтение CTRPRD осуществляется напрямую).	R/W	1h
[2]	SYNCPHSEN	Сигнал разрешения загрузки счетчика из регистра фазы 0 — загрузка CTRCNT из регистра фазы CTRPHS запрещена 1 — загрузка CTRCNT из регистра фазы CTRPHS во время синхронизации разрешена	R/W	0
[1:0]	CNTMODE	Режим работы счетчика CTRCNT. 0h — up-count режим; 1h — down-count режим; 2h — up-down-count режим; 3h — счет не осуществляется	R/W	2h

### 17.2.2.2 Регистр фазы синхронизации CTRPHS (Таблица 17.5)

Таблица 17.5. Регистр CTRPHS

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:0]	CTRPHS	Содержит значение, в которое переключится счетчик CTRCNT во время синхронизации. Если CLKCTL[SYNCPHSEN]=0, то при возникновении события синхронизации счетчик CTRCNT не переходит в значение, прописанное в регистре CTRPHS.	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
		Если CLKCTL[SYNCPHSEN]=1, то при возникновении события синхронизации счетчик CTRCNT загружает значение регистра CTRPHS.		

### 17.2.2.3 Регистр счетчика CTRCNT (Таблица 17.6)

Таблица 17.6. Регистр CTRCNT

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:0]	CTRCNT	При чтении данный регистр отображает текущее значение счетчика CTRCNT. При записи прописывается новое значение счетчика CTRCNT без синхронизации с частотой работы счетчика и без использования теневого регистра.	R/W	0

### 17.2.2.4 Регистр периода счета CTRPRD (Таблица 17.7)

Таблица 17.7. Регистр CTRPRD

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:0]	CTRPRD	Данное поле определяет период счетчика CTRCNT и, соответственно, устанавливает частоту PWM. Теневой регистр включается и выключается битом CLKCTL[LOADPRD]. Если CLKCTL[LOADPRD]=0, теневой регистр включен, и любая запись или чтение проходят через него. Активный регистр загружается из теневого в момент, когда CTRCNT=0 Если CLKCTL[LOADPRD]=1, теневой регистр выключен, и любая запись или чтение проходит напрямую без участия теневого регистра. Активный и теневой регистры имеют единый адрес.	R/W	0

## 17.2.3 Регистры блока Compare unit

### 17.2.3.1 Регистр управления параметрами блока сравнения Compare unit CMPCTL (Таблица 17.8)

Таблица 17.8. Регистр CMPCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:8]	-			
[7]	SCMPBFULL	Признак наличия данных в теневом регистре сравнения CMPB не загруженных в активный регистр сравнения CMPB. Бит сбрасывается во время загрузки данных в активный регистр. 0 — нет не загруженных данных 1 — есть не загруженные данные	R	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[6]	SCMPAFULL	Признак наличия данных в теневом регистре сравнения СМРА не загруженных в активный регистр сравнения СМРА. Бит сбрасывается во время загрузки данных в активный регистр. 0 — нет не загруженных данных 1 — есть не загруженные данные	R	0
[5]	SCMPBMODE	Режим работы регистра <i>СМРВ</i> 0 — работа с теневым регистром, все запросы <i>CPU</i> проходят через теневой регистр; 1 — прямой режим, используется только активный регистр <i>СМРВ</i>	R/W	0
[4]	SCMPAMODE	Режим работы регистра СМРА 0 — работа с теневым регистром, все запросы <i>CPU</i> проходят через теневой регистр; 1 — прямой режим, используется только активный регистр СМРА	R/W	0
[3:2]	LDBMODE	Выбор режима загрузки данных из теневого регистра в активный СМРВ. Данное поле не имеет значения при работе в режиме прямой загрузки $СМРCTL[SCMPBMODE]=1$ 0h — загрузка при $CTRCNT=0$ 1h — загрузка при $CTRCNT=CTRPRD$ 2h — загрузка при $CTRCNT=0$ или $CTRCNT=CTRPRD$ 3h — загрузка не осуществляется	R/W	0
[1:0]	LDAMODE	Выбор режима загрузки данных из теневого регистра в активный СМРА. Данное поле не имеет значения при работе в режиме прямой загрузки $СМРCTL[SCMPAMODE]=1$ 0h — загрузка при $CTRCNT=0$ 1h — загрузка при $CTRCNT=CTRPRD$ 2h — загрузка при $CTRCNT=0$ или $CTRCNT=CTRPRD$ 3h — загрузка не осуществляется	R/W	0

### 17.2.3.2 Регистр сравнения СМРА (Таблица 17.9)

Таблица 17.9. Регистр СМРА

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
---------------	----------------------	----------	--------	--------------------



[31:0]	СМРА	<p>Содержимое активного регистра СМРА сравнивается со значением счетчика CRTCNT. Если значения равны, то блок сравнения генерирует событие «СМРА==CRTCNT». По этому событию менеджер событий (Event manager) совершает одно или более действие. Эти действия могут быть направлены на формирование выходных сигналов OUTA или OUTB в зависимости от конфигурации регистров EMCTLA и EMCTLB. Реакция на событие может быть следующая:</p> <ul style="list-style-type: none"> <li>не производить никаких действий;</li> <li>зажать OUTA и/или OUTB в 0;</li> <li>зажать OUTA и/или OUTB в 1;</li> <li>инвертировать OUTA и/или OUTB</li> </ul> <p>Теневой регистр включен по умолчанию и управляется битом CMPCTL[SCMPAMODE]. Если CMPCTL[SCMPAMODE]=0, то теневой регистр включен, запись/чтение проходит через него. В этом случае CMPCTL[LDAMODE] определяет момент загрузки данных в активный регистр.</p> <p>Перед записью необходимо удостовериться, что в теневом регистре нет переданных в активный регистр данных проанализировав CMPCTL[SCMPAFULL] бит.</p> <p>Если CMPCTL[SCMPAMODE]=1, то теневой регистр не используется, запись/чтение осуществляется с активным регистром.</p> <p>В любом режиме работы активный и теневой регистры имеют один адрес.</p>	R/W	0
--------	------	--	-----	---

### 17.2.3.3 Регистр сравнения СМРВ (Таблица 17.10)

Таблица 17.10. Регистр СМРВ

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:0]	СМРВ	<p>Содержимое активного регистра СМРВ сравнивается со значением счетчика CRTCNT. Если значения равны, то блок сравнения генерирует событие «СМРВ==CRTCNT». По этому событию менеджер событий (Event manager) совершает одно или более действие. Эти действия могут быть направлены на формирование выходных сигналов OUTA или OUTB в зависимости от конфигурации регистров EMCTLA и EMCTLB. Реакция на событие может быть следующая:</p> <ul style="list-style-type: none"> <li>не производить никаких действий;</li> <li>зажать OUTA и/или OUTB в 0;</li> <li>зажать OUTA и/или OUTB в 1;</li> <li>инвертировать OUTA и/или OUTB</li> </ul> <p>Теневой регистр включен по умолчанию и управляется битом CMPCTL[SCMPBMODE]. Если CMPCTL[SCMPBMODE]=0, то теневой регистр включен, запись/чтение проходит через него. В этом случае CMPCTL[LDBMODE] определяет момент загрузки данных в активный регистр.</p>	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
		Перед записью необходимо удостовериться, что в теневом регистре нет не переданных в активный регистр данных проанализировав CMPCTL[SCMPBFULL] бит. Если CMPCTL[SCMPBMODE]=1, то теневой регистр не используется, запись/чтение осуществляется с активным регистром. В любом режиме работы активный и теневой регистры имеют один адрес.		

## 17.2.4 Регистры блока Event manager

### 17.2.4.1 Регистр управления выхода OUTA блока Event manager EMCTLA (Таблица 17.11)

Таблица 17.11. Регистр EMCTLA

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:12]	-			
[11:10]	ECMPBD	Действие, которое нужно выполнить при декременте счетчика и CTRCNT=CMPB 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0
[9:8]	ECMPBI	Действие, которое нужно выполнить при инкременте счетчика и CTRCNT=CMPB 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0
[7:6]	ECMPAD	Действие, которое нужно выполнить при декременте счетчика и CTRCNT=CMPA 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0
[5:4]	ECMPAI	Действие, которое нужно выполнить при инкременте счетчика и CTRCNT=CMPA 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0
[3:2]	EPRD	Действие, которое нужно выполнить при CTRCNT=CTRPRD 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0
[1:0]	EZRO	Действие, которое нужно выполнить при CTRCNT=0 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0

### 17.2.4.2 Регистр управления выхода OUTB блока Event manager EMCTLB (Таблица 17.12)

Таблица 17.12. Регистр EMCTLB

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:12]	-			
[11:10]	ECMPBD	Действие, которое нужно выполнить при декременте счетчика и CTRCNT=CMPB 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[9:8]	ECMPBI	Действие, которое нужно выполнить при инкременте счетчика и CTRCNT=CMPB 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[7:6]	ECMPAD	Действие, которое нужно выполнить при декременте счетчика и CTRCNT=CMPA 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[5:4]	ECMPAI	Действие, которое нужно выполнить при инкременте счетчика и CTRCNT=CMPA 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[3:2]	EPRD	Действие, которое нужно выполнить при CTRCNT=CTRPRD 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[1:0]	EZRO	Действие, которое нужно выполнить при CTRCNT=0 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0

### 17.2.4.3 Регистр программного управления выходами OUTA/OUTB с однократным действием блока Event manager EMSWFR (Таблица 17.13)

Таблица 17.13. Регистр EMSWFR

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:8]	-			
[7:6]	LDCSWRF	Определяет механизм загрузки активного регистра EMCSWFR из теневого регистра 0h — загрузка при CTRCNT=0 1h — загрузка при CTRCNT=CTRPRD 2h — загрузка при CTRCNT=0 или CTRCNT=CTRPRD	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
		3h — загружать напрямую при обращении CPU без использования теневого регистра.		
[5]	ONESFB	Программный вызов единичного переключения для выхода OUTB 0 — запись 0 не вызывает эффекта, читается всегда 0 1 — запись 1 генерирует однократное переключение выхода OUTB, затем данный бит сбрасывается в 0	R/W	0
[4:3]	ACTSFB	Действие, которое нужно выполнить при программном вызове переключения OUTB 0h — не выполнять действий; 1h — clear: сбросить OUTB в 0 2h — set: установить OUTB в 1 3h — toggle: инвертировать OUTB	R/W	0
[2]	ONESFA	Программный вызов единичного переключения для выхода OUTA 0 — запись 0 не вызывает эффекта, читается всегда 0 1 — запись 1 генерирует однократное переключение выхода OUTA, затем данный бит сбрасывается в 0	R/W	0
[1:0]	ACTSFA	Действие, которое нужно выполнить при программном вызове переключения OUTA 0h — не выполнять действий; 1h — clear: сбросить OUTA в 0 2h — set: установить OUTA в 1 3h — toggle: инвертировать OUTA	R/W	0

#### 17.2.4.4 Регистр программного управления выходами OUTA/OUTB с продолжительным действием блока Event manager EMCSWFR (Таблица 17.14)

Таблица 17.14. Регистр EMCSWFR

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:4]	-			
[3:2]	LONGSFB	Продолжительное программное воздействие на OUTB При использовании теневого регистра эффект от воздействия продолжительного зажима выхода наступает на следующий такт CTRCNT после загрузки данных в активный регистр Во время работы без теневого регистра эффект наступает на следующий такт CTRCNT Управление режимом работы осуществляется при помощи бита EMSWFR[LDCSWRF] 0h — нет воздействия 1h — OUTB продолжительно зажимается в 0 2h — OUTB продолжительно зажимается в 1 3h — программное воздействие запрещено	R/W	0
[1:0]	LONGSFA	Продолжительное программное воздействие на OUTA	R/W	0

		<p>При использовании теневого регистра эффект от воздействия продолжительного зажима выхода наступает на следующий такт CTRCNT после загрузки данных в активный регистр</p> <p>Во время работы без теневого регистра эффект наступает на следующий такт CTRCNT</p> <p>Управление режимом работы осуществляется при помощи бита EMSWFR[LDCSWRF]</p> <p>0h — нет воздействия</p> <p>1h — OUTA продолжительно зажимается в 0</p> <p>2h — OUTA продолжительно зажимается в 1</p> <p>3h — программное воздействие запрещено</p>		
--	--	--	--	--

## 17.2.5 Регистры блока Dead zone.

### 17.2.5.1 Регистр управления блока Dead zone DZCTL (Таблица 17.15)

Таблица 17.15. Регистр DZCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:4]	-			
[5]	INMUXF	Выбор источника сигнала для генерации запрещенной зоны после заднего фронта 0 — подключен сигнал OUTA in 1 — подключен сигнал OUTB in	R/W	0
[4]	INMUXR	Выбор источника сигнала для генерации запрещенной зоны после переднего фронта 0 — подключен сигнал OUTA in 1 — подключен сигнал OUTB in	R/W	0
[3]	INVMUXB	Управление полярностью OUTB после генерации запрещенной зоны 0 — нет инверсии 1 — есть инверсия	R/W	0
[2]	INVMUXA	Управление полярностью OUTA после генерации запрещенной зоны 0 — нет инверсии 1 — есть инверсия	R/W	0
[1]	OUTMUXB	Выбор режима работы блока dead zone при формировании OUTB 0 — генератор запрещенной зоны после заднего фронта выключен, OUTB транслируется без изменений, поля регистра DZCTL[INMUXF] и DZCTL[INVMUXB] ни на что не влияют 1 - генератор запрещенной зоны после заднего фронта включен, входной сигнал для формирования OUTB определяется битом DZCTL[INMUXF]	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[0]	OUTMUXA	Выбор режима работы блока dead zone при формирования OUTA 0 — генератор запрещенной зоны после переднего фронта выключен, OUTA транслируется без изменений, поля регистра DZCTL[INMUXR] и DZCTL[INVMUXA] ни на что не влияют 1 - генератор запрещенной зоны после заднего фронта включен, входной сигнал для формирования OUTA определяется битом DZCTL[INMUXR]	R/W	0

### 17.2.5.2 Регистр периода задержки после переднего фронта блока Dead zone DZRPER (Таблица 17.16)

Таблица 17.16. Регистр DZRPER

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:10]	-			
[9:0]	PER	Определяет длительность задержки для формирования запрещенной зоны после переднего фронта	R/W	0

### 17.2.5.3 Регистр периода задержки после заднего фронта блока Dead zone DZFPER (Таблица 17.17)

Таблица 17.17. Регистр DZFPER

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:10]	-			
[9:0]	PER	Определяет длительность задержки для формирования запрещенной зоны после заднего фронта	R/W	0

## 17.2.6 Регистры блока Chopper

### 17.2.6.1 Регистр управления блока Chopper CHCTL (Таблица 17.18)

Таблица 17.18. Регистр CHCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:11]	-			
[10:8]	CHDUTY	Скважность дробящего сигнала 0h — 1/8 1h — 2/8 2h — 3/8 3h — 4/8 4h — 5/8		

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
		5h — 6/8 6h — 7/8 7h — не используется		
[7:5]	CHCLKDIV	Выбор частоты дробящего сигнала 0h — CLK/8 1h — CLK/16 2h — CLK/24 3h — CLK/32 4h — CLK/40 5h — CLK/48 6h — CLK/56 7h — CLK/64		
[4:1]	FIRSTWTH	Ширина первого импульса 0h — 0xCLK/8 1h — 1xCLK/8 2h — 2xCLK/8 3h — 3xCLK/8 ... Fh — 15xCLK/8		
[0]	CHEN	Управление работой блока Chopper 0 — выключена функция дробления 1 — включена функция дробления		

## 17.2.7 Регистры блока Trip unit

### 17.2.7.1 Регистр выбора принципа работы блока trip unit TUSEL (Таблица 17.19)

Таблица 17.19. Регистр TUSEL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:16]	-			
[15:8]	ONE7 ... ONE0	Выбор используемых TU[7:0] сигналов для канала PWM работающих в режиме однократного срабатывания 0 — вход не используется 1 — вход используется в режиме однократного срабатывания	R/W	0
[7:0]	MULT7 ... MULT0	Выбор используемых TU[7:0] сигналов для канала PWM работающих в режиме многократного срабатывания 0 — вход не используется 1 — вход используется в режиме многократного срабатывания	R/W	0

### 17.2.7.2 Регистр определения реакции на событие блока trip unit TUCTL (Таблица 17.20)

Таблица 17.20. Регистр TUCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
---------------	----------------------	----------	--------	--------------------

[31:4]	-			
[3:2]	TUB	Когда происходит TRIP событие, к выходу OUTB могут быть применены следующие действия: 0h — OUTB переводится в high impedance state 1h — OUTB подтягивается к 1 2h — OUTB подтягивается к 0 3h — действий не производится	R/W	0
[1:0]	TUA	Когда происходит TRIP событие, к выходу OUTA могут быть применены следующие действия: 0h — OUTA переводится в high impedance state 1h — OUTA подтягивается к 1 2h — OUTA подтягивается к 0 3h — действий не производится	R/W	0

### 17.2.7.3 Регистр маски прерывания блока trip unit TUINTM (Таблица 17.21)

Таблица 17.21. Регистр TUINTM

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:2]	-			
[1]	ONE	Разрешение прерывания при возникновении события ONE TRIP 0 — прерывание запрещено 1 — прерывание разрешено, генерируется прерывание PWMТУ_INT	R/W	0
[0]	MULT	Разрешение прерывания при возникновении события MULT TRIP 0 — прерывание запрещено 1 — прерывание разрешено, генерируется прерывание PWMТУ_INT	R/W	0

### 17.2.7.4 Регистр статуса блока trip unit TUSTS (Таблица 17.22)

Таблица 17.22. Регистр TUSTS

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:3]	-			
[2]	TUINT	Признак возникновения прерывания 0 — прерывания не было 1 — было сформировано прерывание PWMТУ_INT Дальнейшие прерывания генерироваться не будут до тех пор, пока данный флаг не будет очищен. Если флаг прерывания будет очищен, а события его вызвавшие все еще установлены, то сформируется еще одно прерывание. Чтобы избежать дальнейшей генерации прерывания нужно очистить все флаги статуса.	R	0
[1]	ONE	Статус события ONE 0 — событие не произошло 1 — событие произошло Сброс данного бита осуществляется только программно записью «1» в регистр TUCLR[ONE]	R	0



Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[0]	MULT	Статус события MULT 0 — событие не произошло 1 — событие произошло Данный бит остается установленным до тех пор, пока не будет очищен. Очищается он программно (записью «1» в регистр TUCLR[MULT]) или при достижении основного счетчика нулевого значения (CTRCNT=0), но только в том случае, если в момент очистки бита не будет присутствовать событие MULT на входах TU (&TU == 1). Если событие вызвавшее возникновение статуса все еще присутствует, то сбрасываемый бит вновь возводится в «1»	R	0

### 17.2.7.5 Регистр сброса статуса блока trip unit TUCLR (Таблица 17.23)

Таблица 17.23. Регистр TUCLR

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:2]	-			
[2]	TUINT	Бит сброса статуса глобального прерывания PWM_TU_INT 0 — нет эффекта, читается всегда как 0 1 — очищает флаг прерывания TUSTS[TUINT] Никакие дальнейшие прерывания PWM_TU_INT не будут формироваться, пока флаг не очищен. Если TUSTS[INT], бит очищен, а какой-либо из других флагов будет установлен, то будет сформирован еще один импульс прерывания. Очистка всех флагов предотвратит дальнейшее формирование запросов на прерывание.	R/W	0
[1]	ONE	Бит сброса статуса события ONE 0 — нет эффекта, читается всегда как 0 1 — очищает этот флаг	R/W	0
[0]	MULT	Бит сброса статуса события MULT 0 — нет эффекта, читается всегда как 0 1 — очищает этот флаг	R/W	0

### 17.2.7.6 Регистр программного формирования событий блока Trip unit TUFRC (Таблица 17.24)

Таблица 17.24. Регистр TUFRC

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:2]	-			
[1]	ONE	Программная эмуляция возникновения события ONE TRIP 0 — запись 0 игнорируется, читается всегда 0 1 — запись 1 имитирует возникновение события ONE TRIP и провоцирует установку бита TUSTS[ONE]	R/W	0

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[0]	MULT	Программная эмуляция возникновения события MULT TRIP 0 — запись 0 игнорируется, читается всегда 0 1 — запись 1 имитирует возникновение события MULT TRIP и провоцирует установку бита TUSTS[MULT]	R/W	0

## 17.2.8 Регистры блока Interrupt control

### 17.2.8.1 Регистр выбора принципа работы блока Interrupt control ICSEL (Таблица 17.25)

Таблица 17.25. Регистр ICSEL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:4]	-			
[3]	INTEN	Бит разрешения формирования прерывания PWM_INT 0 — прерывание PWM_INT запрещено 1 — прерывание PWM_INT разрешено	R/W	0
[2:0]	INTSEL	Выбор источника прерывания 0h — 1h — резерв 2h — признак равенства счетчика нулю CTCNT=0 3h — признак равенства счетчика периоду счета CTCNT=CTRPRD 4h — признак равенства счетчика регистру сравнения CMPA во время инкремента 5h — признак равенства счетчика регистру сравнения CMPA во время декремента 6h — признак равенства счетчика регистру сравнения CMPB во время инкремента 7h — признак равенства счетчика регистру сравнения CMPB во время декремента	R/W	0

### 17.2.8.2 Регистр выбора принципа работы блока Interrupt control ICCTL (Таблица 17.26)

Таблица 17.26. Регистр ICCTL

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:4]	-			

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[3:2]	EVENTCNT	Счетчик событий вызывающих прерывание. Эти биты отображают количество произошедших событий выбранных регистром ICSEL[INTSEL]. Данный счетчик автоматически сбрасывается при формировании импульса прерывания. Если прерывание отключено ICSEL[INTEN]=0 или уже сформировано ICSTS[INT]=1, то подсчет событий не производится 0h — события не обнаружены 1h — обнаружено 1 событие 2h — обнаружено 2 события 3h — обнаружено 3 события	R	0
[1:0]	EVENTPRD	Выбор периода прерываний PWM_INT. Эти биты определяют, сколько необходимо зафиксировать событий, выбранных регистром ICSEL[INTSEL] для формирования прерывания. 0h — нет генерации прерывания 1h — генерация прерывания каждый раз при возникновении события 2h — генерация прерывания каждый второй раз при возникновении события 3h - генерация прерывания каждый третий раз при возникновении события	R/W	0

### 17.2.8.3 Регистр статуса блока Interrupt control ICSTS (Таблица 17.27)

Таблица 17.27. Регистр ICSTS

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:1]	-			
[0]	INT	Отображает состояние входа PWM_INT 0 — нет прерывания 1 — есть прерывание. Новые прерывания не формируются до тех пор, пока данный бит не будет очищен.	R	0

### 17.2.8.4 Регистр очистки битов статуса блока Interrupt control ICCLR (Таблица 17.28)

Таблица 17.28. Регистр ICCLR

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:1]	-			
[0]	INT	Бит сброса прерывания PWM_INT 0 — нет действия, читается всегда 0 1 — сброс прерывания PWM_INT и его статусного бита	R/W	0

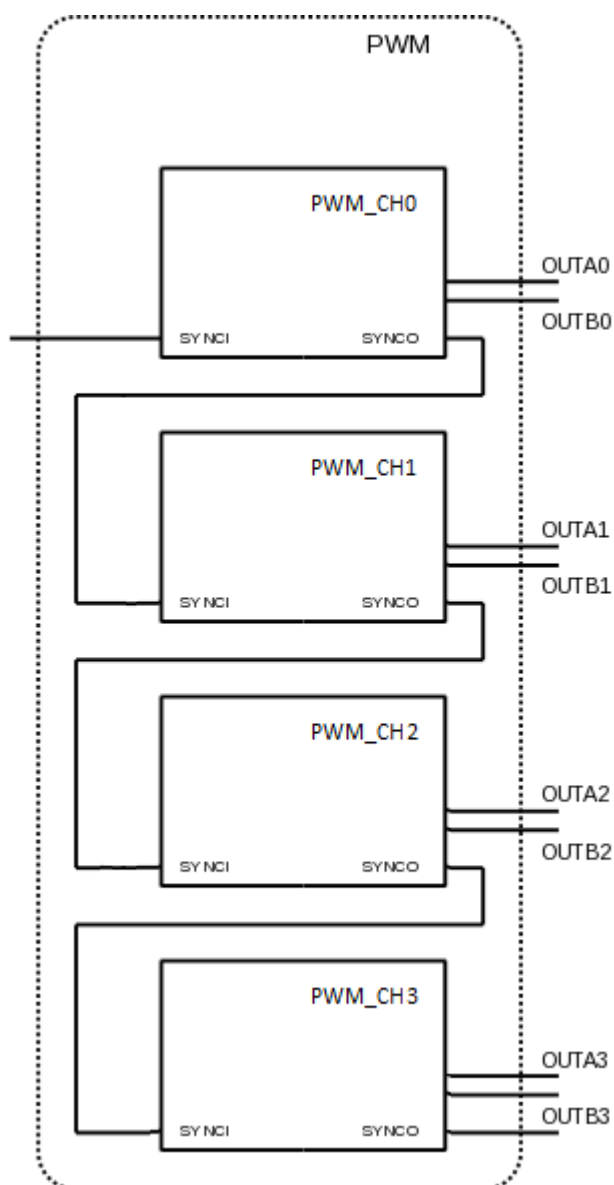
### 17.2.8.5 Регистр программного формирования событий блока Interrupt control ICFRC (Таблица 17.29)

Таблица 17.29. Регистр ICFRC

Номер разряда	Условное обозначение	Описание	Доступ	Исходное состояние
[31:1]	-			
[0]	INT	Бит программного формирования прерывания 0 — нет действия, читается всегда 0 1 — запись 1 ведет к формированию прерывания PWM_INT и установке его статусного бита	R/W	0

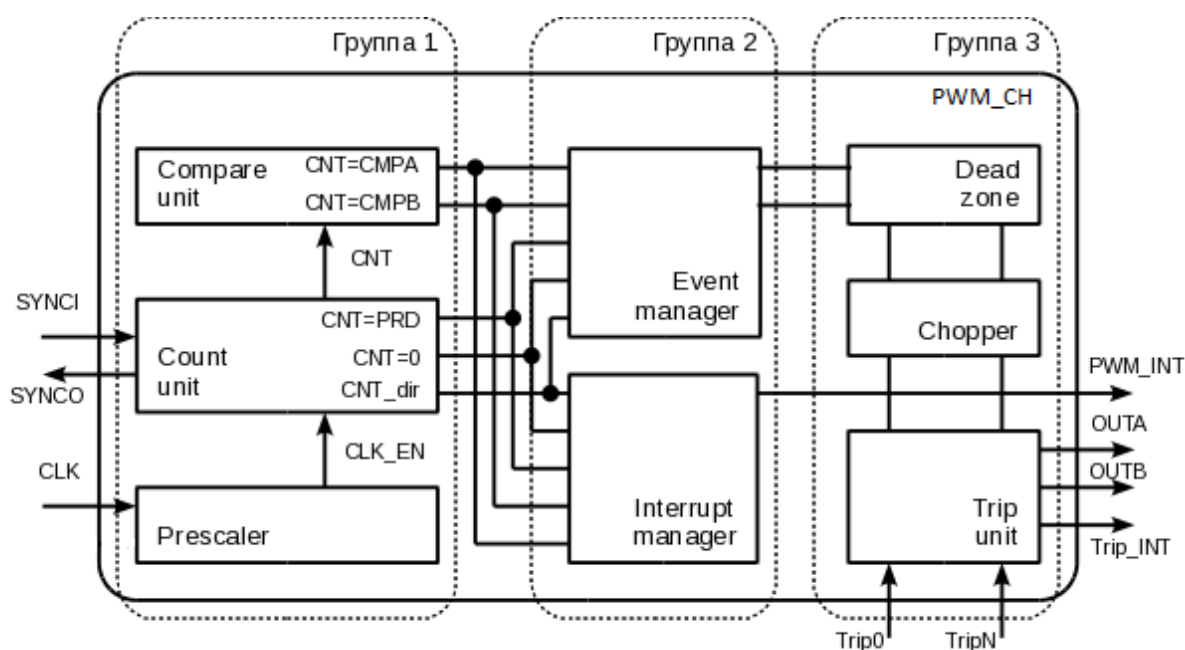
## 17.3 Функционирование

Структурная схема PWM представлена на Рисунок 17.1



**Рисунок 17.1. Структурная схема PWM**

PWM\_CH состоит из 8 блоков (Рисунок 17.2)



**Рисунок 17.2. Структурная схема PWM\_CH**

Условно их можно разделить на три группы:

Группа 1 — основные функциональные блоки (предделитель, счетчик, компаратор). Основная функция данной группы — формирование ряда периодичных событий, на основе которых осуществляется функционирование всего канала.

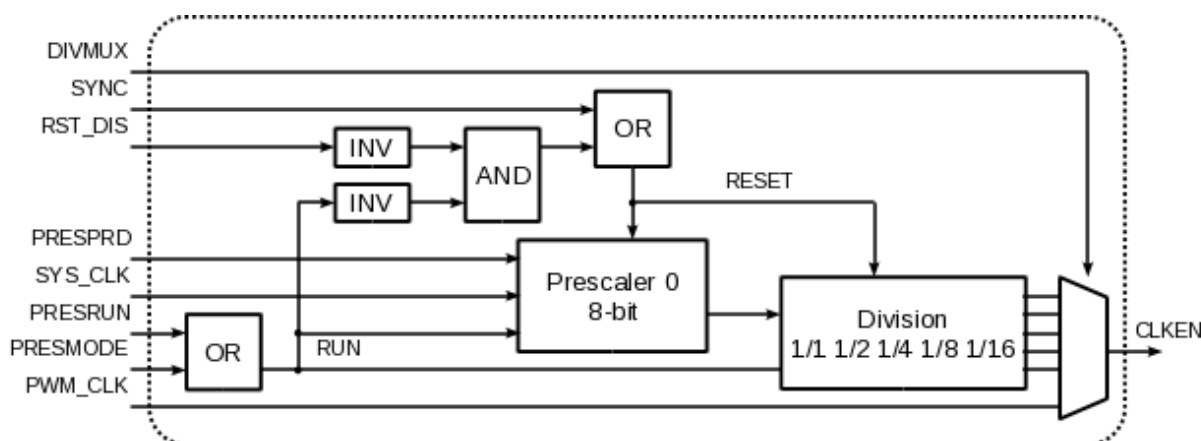
Группа 2 — блоки определяющие реакцию на события, поступающие из первой группы блоков и формирующие основные сигналы PWM\_CH OUTA, OUTB, PWM\_INT. Для широтно-импульсной модуляции достаточно задействовать первые две группы блоков.

Группа 3 — блоки, осуществляющие дополнительное форматирование сформированных на предыдущих стадиях сигналов. Данная группа блоков может использоваться по отдельности, все вместе, так и не использоваться вообще, в этом случае сигналы проходят на выход без изменения.

Для большей функциональной гибкости используется структура из четырех независимых PWM\_CH устройств с возможностью одновременного пуска.

### 17.3.1 Предделитель частоты (prescaler)

На Рисунок 17.3 представлена функциональная схема предделителя частоты.



**Рисунок 17.3. Функциональная схема делителя частоты**

Задача делителя — подготовить тактовый сигнал нужного периода (CLKEN). Генерация данного сигнала может осуществляться постоянно (в режиме  $\text{PRESMODE} == 1$ ) или только при включении PWM\_CH ( $\text{PRESMODE} == 0$  &  $\text{PRESRUN} = 1$ , где  $\text{PRESRUN}$  — признак включения PWM\_CH).

Во время остановки таймера делитель переходит в исходное состояние (при  $\text{RST\_DIS} == 0$ ) или сохраняет текущее (при  $\text{RST\_DIS} == 1$ )

Данная система пуска и останова позволяет при наращивании количества PWM\_CH в системе реализовать их синхронную работу.

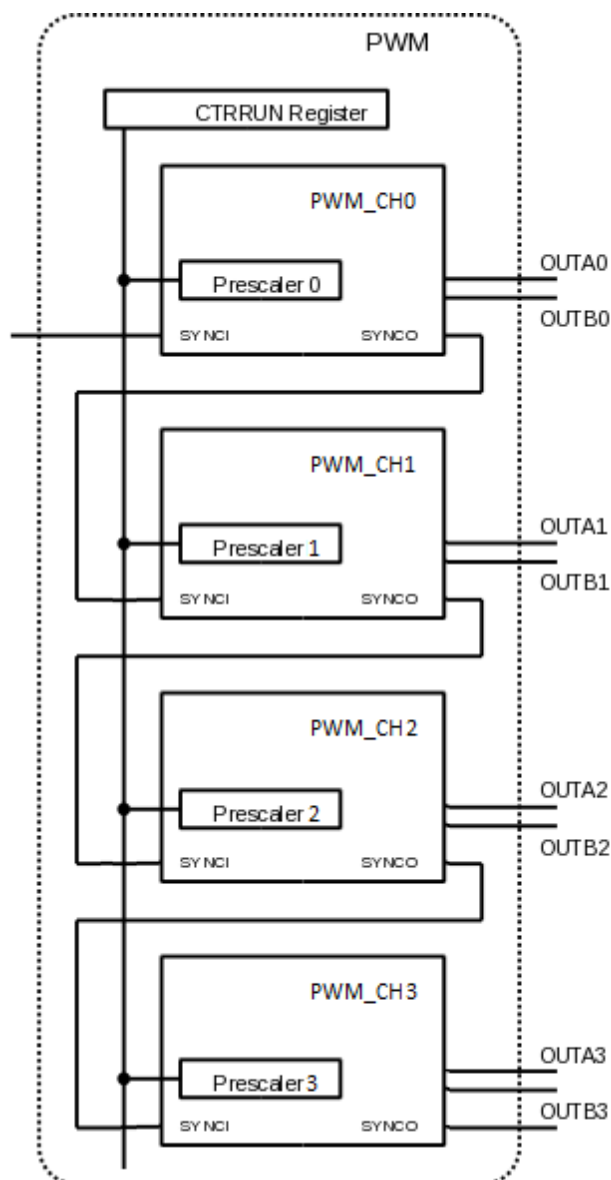
Перевод делителя в исходное состояние возможен также при возникновении внешнего сигнала SYNCI (или его программной эмуляции).

Для формирования рабочей частоты используются два уровня деления. На первом уровне (prescaler) реализован 8-разрядный счетчик-делитель способный уменьшить частоту в 256 раз (в соответствии со значением регистра  $\text{CLKCTL}[\text{PRESPRD}]$ ), на втором уровне частота делится в соотношении 1/1, 1/2, 1/4, 1/8, 1/16 либо переключается на внешний источник тактового сигнала PWM\_CLK (в соответствии со значением регистра  $\text{CLKCTL}[\text{DIVMUX}]$ ).

Для управления пуском делителей различных каналов PWM Рисунок 17.4 используется единый регистр CTRRUN позволяющий синхронизировать работу таймеров в момент пуска и останова.

В PWM реализовано 4-х независимых канала, каждый из которых может работать как асинхронно друг относительно друга, так и синхронно.

Настройка периода частоты синхронизации таймера осуществляется программированием регистра CLKCTL полей PRESPRD и DIVMUX.



**Рисунок 17.4. Управление пуском предделителей различных каналов PWM**

Период полученного сигнала CLKENx можно определить по формуле:

$$T_{CLKEN} = T_{CLK} * (CLKCTL[PRESPRD] + 1) * 2^{CLKCTL[DIVMUX]} \quad - \text{при } CLKCTL[DIVMUX] \leq 4$$

$$T_{CLKEN} = 2 * T_{PWM\_CLK} \quad - \text{при } CLKCTL[DIVMUX] > 4$$

### 17.3.2 Основной счетчик (Count unit)

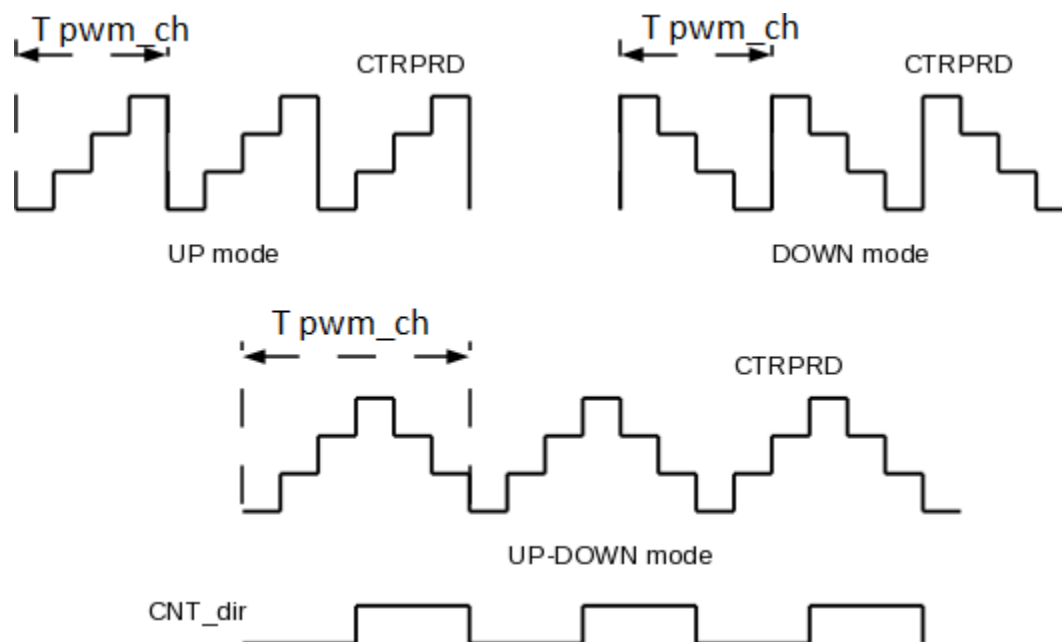
Функции:

- организует счет в трех режимах:
  - up;
  - down;
  - up-down;



- генерирует события:
- равенство счетчика нулю ( $CTRCNT=0$ );
- равенство счетчика периоду ( $CTRCNT=CTRPRD$ );
- синхронизация фазы с другими счетчиками.

Вычисление периода и частоты PWM\_CH (Рисунок 17.5)



**Рисунок 17.5. Вычисление периода и частоты PWM\_CH**

Таким образом период PWM\_CH определяем по формуле  $T_{PWM\_CH} = (CTRPRD + 1) * T_{CLKEN}$  - в режимах UP и DOWN,  $T_{PWM\_CH} = 2 * (CTRPRD + 1) * T_{CLKEN}$  - режиме UP-DOWN.

Теневой (буферный) регистр периода CTRPRD

Регистр периода CTRPRD имеет двойника, который используется для аппаратной синхронизации записи нового значения периода. Фактически имеется два регистра:

- активный регистр — регистр непосредственно участвующий в работе счетчика;
- теневой регистр — регистр, предназначенный для временного хранения данных предназначенных для активного регистра.

Адрес у этих регистров один и тот же, а с каким из них работает программист в данный момент определяет бит CLKCTL[LOADPRD]:

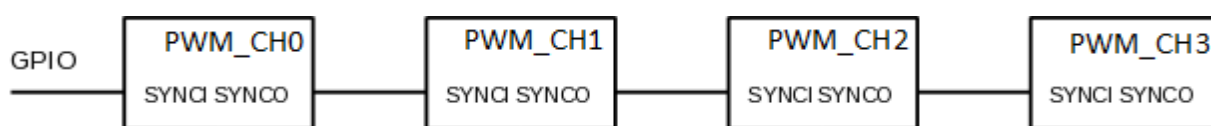
- режим работы с участием теневого регистра CLKCTL[LOADPRD]=0. Данные записанные программистом в регистр периода хранятся в теневом регистре до тех пор, пока счетчик не будет равен нулю ( $CTRCNT=0$ ). Когда счетчик обнулится данные из теневого регистра перепишутся в активный;

- режим работы без участия теневого регистра CLKCTL[LOADPRD]=1. Доступ к активному регистру осуществляется напрямую. Данный режим установлен по умолчанию.

Синхронизация каналов PMW при помощи сигналов SYNCI/O (Рисунок 17.6)

Под синхронизацией понимается перевод счетчика CTRCNT в значение, прописанное в регистре фазы синхронизации CTRPHS. Синхронизацию можно отключить при помощи бита CLKCTL[SYNCPHSEN]. Признаком к старту синхронизации может быть:

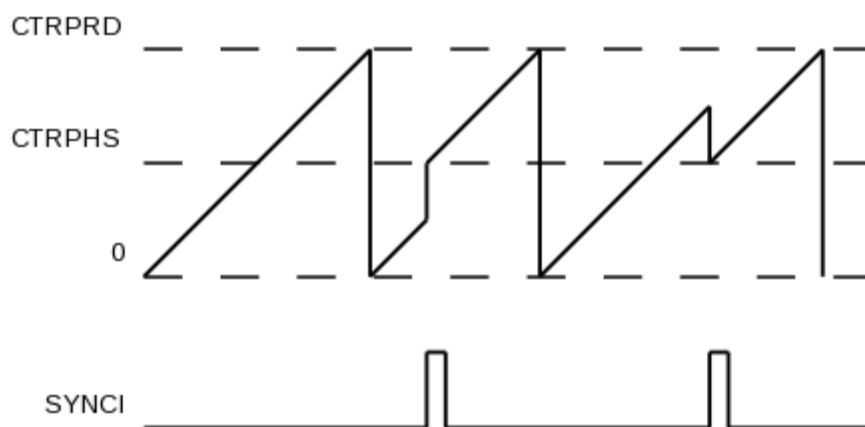
- импульс на входе SYNCI
- программная синхронизация (запись бита CLKCTL[SWFSYNC])



**Рисунок 17.6. Синхронизация каналов**

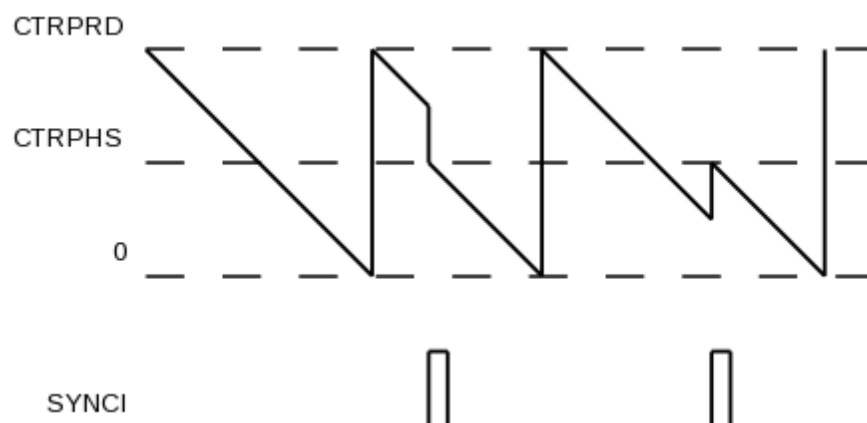
Примеры синхронизации.

Таймер работает в режиме UP-COUNT (Рисунок 17.7).



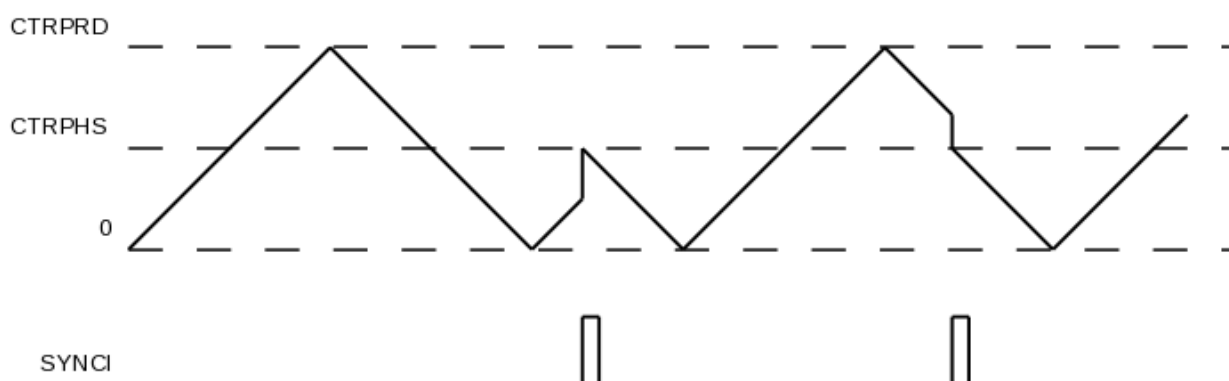
**Рисунок 17.7. Синхронизация при работе в режиме UP\_COUNT**

Таймер работает в режиме DOWN-COUNT (Рисунок 17.8).



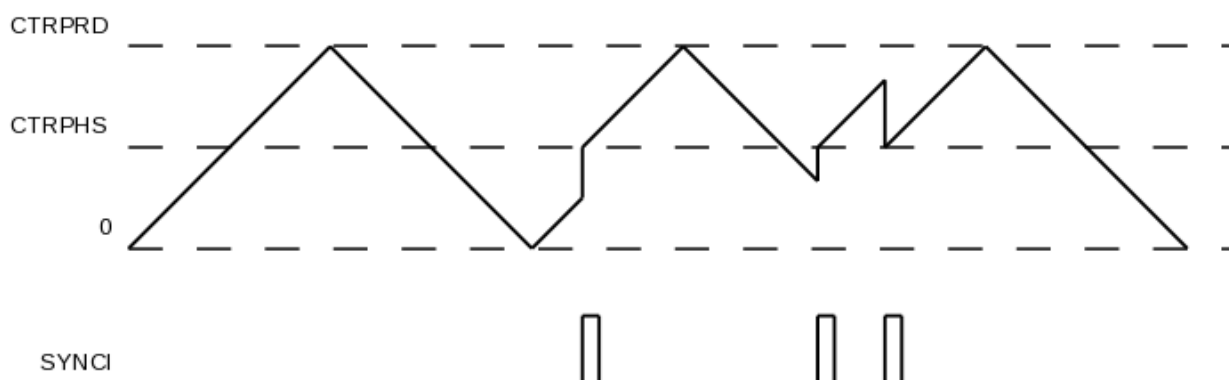
**Рисунок 17.8. Синхронизация при работе в режиме DOWN\_COUNT**

Таймер работает в режиме UP-DOWN-COUNT ( $\text{CLKCTL}[\text{DIRSYNC}]=0$  — после синхронизации счетчик декрементируется Рисунок 17.9).



**Рисунок 17.9. Синхронизация при работе в режиме UP-DOWN-COUNT**

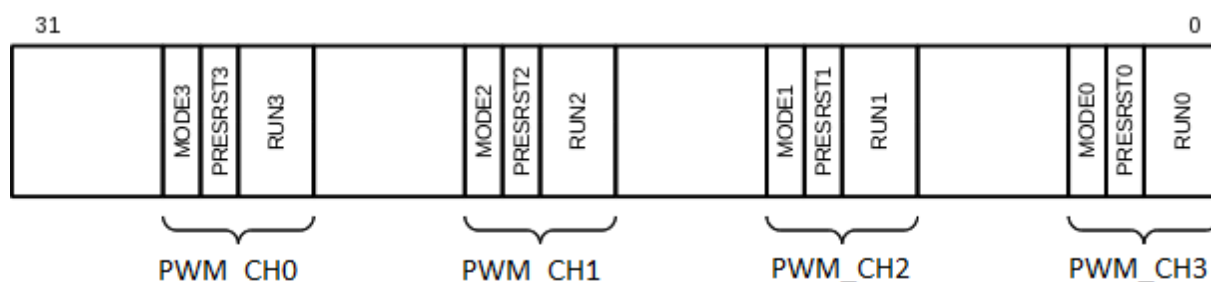
Таймер работает в режиме UP-DOWN-COUNT ( $\text{CLKCTL}[\text{DIRSYNC}]=0$  — после синхронизации счетчик инкрементируется Рисунок 17.10)



**Рисунок 17.10. Синхронизация при работе в режиме UP-DOWN-COUNT**

Пуск и остановка счетчиков осуществляется записью соответствующего значения в регистр  $\text{CTRRUN}[\text{RUN*}]$  (Рисунок 17.11). Для управления каналами PWM ( $\text{PWM\_CH0}$  —

PWM\_CH3) в этом регистре выделены по 3 бита. Наличие основных управляющих сигналов в одном регистре позволяет синхронно управлять всеми каналами.



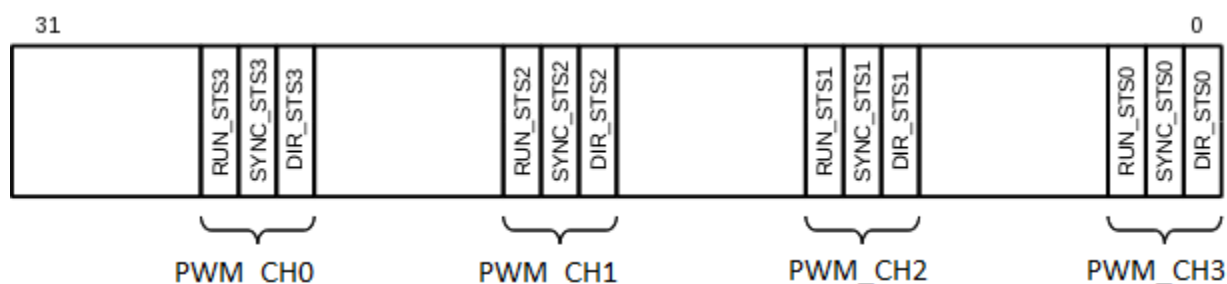
**Рисунок 17.11. Распределение управляющих битов в регистре CTRRUN**

При пуске таймера основной счетчик (CTRCNT) начинает счет с того состояния в котором он находится. В том случае, если счетчик запускается в режиме «DOWN-COUNT», а CTRCNT=0, то следующее значение CTRCNT=CTRPRD, при этом сформируется признак события равенства счетчика нулю ZERO\_EVENT.

Для отображения направления счета используется статусный бит CTRSTS[DIR\_STS].

При пуске канала в режиме «DOWN-COUNT» DIR\_STS устанавливается в 0, в остальных случаях устанавливается в 1. При работе в режиме «UP-DOWN-COUNT» DIR\_STS инвертируется при достижении счетчика нуля (CTRCNT=0) и периода счета (CTRCNT=CTRPRD).

Для удобства статусные биты от всех каналов PWM также сосредоточены в одном регистре CTRSTS (Рисунок 17.12).



**Рисунок 17.12. Распределение битов статуса в регистре CTRSTS**

### 17.3.3 Блок сравнения Compare unit

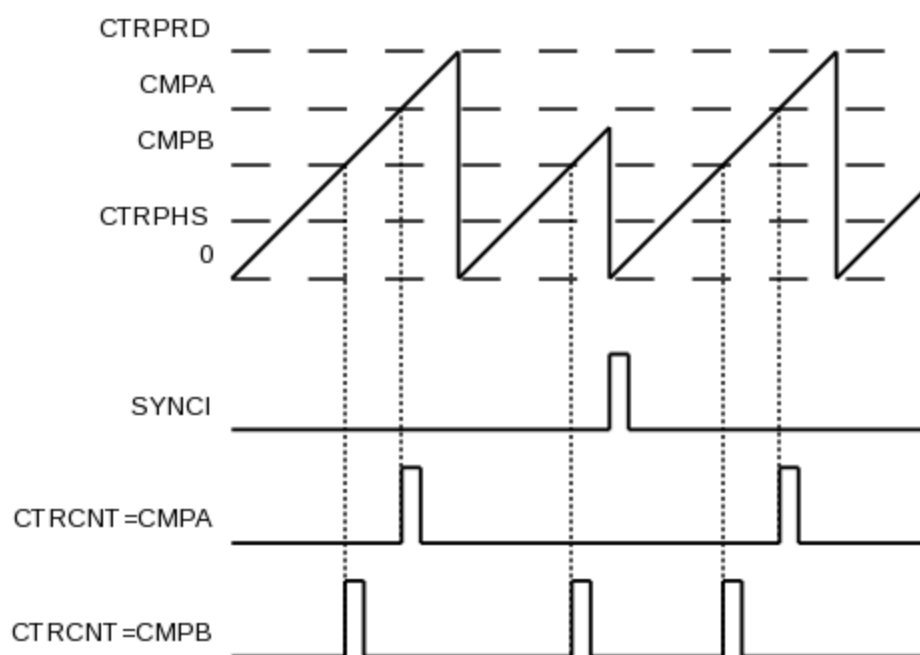
Основная задача блока — сравнение значения записанного в регистрах CMPA и CMPB со значением счетчика CTRCNT и формирование признаков равенства:

- CTRCNT=CMPA;
- CTRCNT=CNB.

Регистры СМРА и СМРВ имеют двойников (теневые регистры), и все операции чтения/записи по умолчанию осуществляются через них. Момент загрузки данных из теневых регистров в основные определяется в регистре управления СМРCTL[LDBMODE] для регистра СМРВ и СМРCTL[LDAMODE] для регистра СМРА. Также в регистре управления присутствуют флаги, определяющие были загружены данные из теневых регистров или нет (СМРCTL[SCMPBFULL] и СМРCTL[SCMPAFULL]). При необходимости теневые регистры можно отключить и работать напрямую с СМРА и СМРВ.

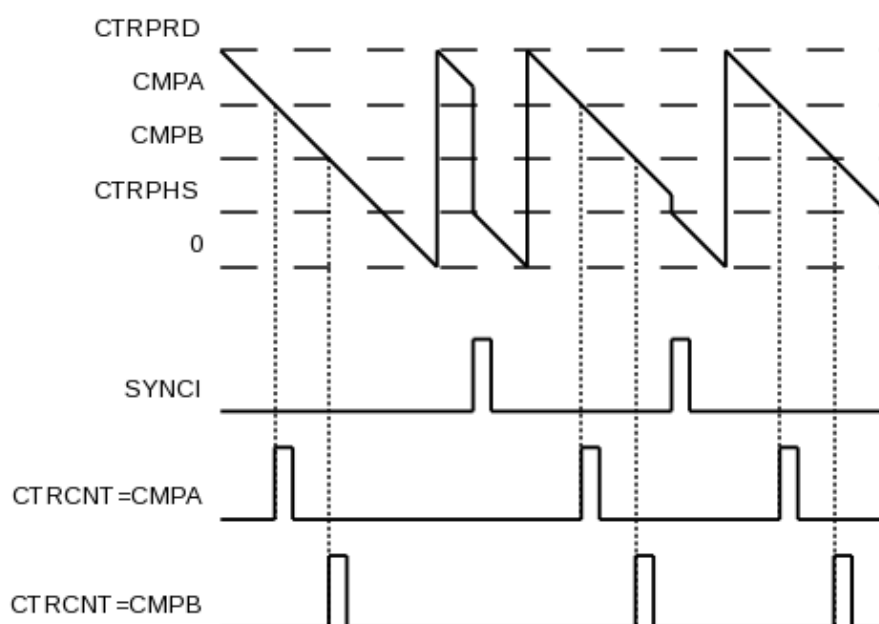
Примеры формирования событий совпадения при работе счетчика в различных режимах.

Таймер работает в режиме UP-COUNT (Рисунок 17.13)



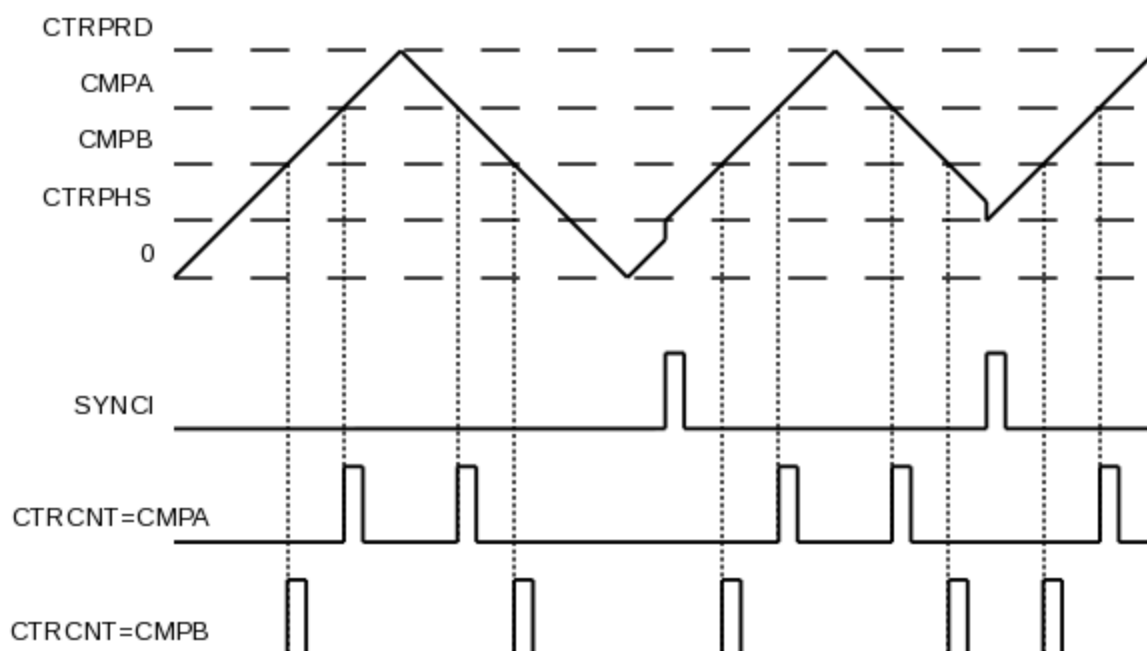
**Рисунок 17.13. Формирование событий при работе в режиме UP-COUNT**

Таймер работает в режиме DOWN-COUNT (Рисунок 17.14).



**Рисунок 17.14. Формирование событий при работе в режиме DOWN-COUNT**

Таймер работает в режиме UP-DOWN-COUNT ( $\text{CLKCTL}[\text{DIRSYNC}] = 0$ ) — после синхронизации счетчик декрементируется (Рисунок 17.15).



**Рисунок 17.15. Формирование событий при работе в режиме UP-DOWN-COUNT**

Таймер работает в режиме UP-DOWN-COUNT ( $\text{CLKCTL}[\text{DIRSYNC}] = 0$ ) — после синхронизации счетчик инкрементируется (Рисунок 17.16).

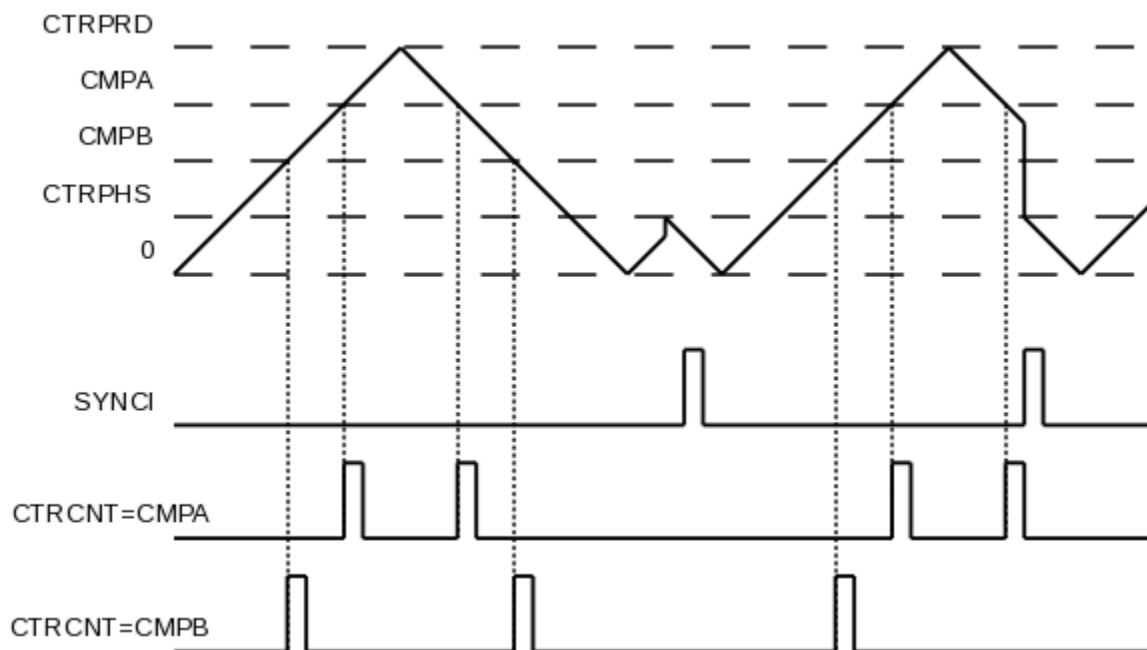


Рисунок 17.16. Формирование событий при работе в режиме UP-DOWN-COUNT

### 17.3.4 Блок реакции на событие Event manager

Задача данного блока — формирование выходных сигналов OUTA и OUTB на основе следующих событий:

- CTRCNT = PRD;
- CTRCNT = CMPA;
- CTRCNT = CMPB;
- CTRCNT = 0.
- программное переключение (SW forced)

При возникновении любого из этих событий возможна одна из следующих реакций:

- установить OUTA и/или OUTB в 1;
- установить OUTA и/или OUTB в 0;
- инвертировать OUTA и/или OUTB.
- OUTA и/или OUTB оставить без изменений.

Как реагировать на возникающие события определяется регистрами EMCTLA для OUTA и EMCTLB для OUTB.

Регистры EMSWFR и EMCSWFR определяют реакцию на программное переключение.

При одновременном возникновении событий отрабатывается событие с наивысшим приоритетом.

Для каждого из режимов работы счетчика определен порядок обработки событий.

Таймер работает в режиме UP-COUNT (перечислено в порядке убывания приоритета Таблица 17.30)

**Таблица 17.30. Порядок обработки событий при работе в UP-COUNT режиме**

1	SW forced
2	CTRCNT = PRD
3	CTRCNT = CMPB
4	CTRCNT = CMPA
5	CTRCNT = 0

Таймер работает в режиме DOWN-COUNT (Таблица 17.31).

**Таблица 17.31. Порядок обработки событий при работе в DOWN-COUNT режиме**

1	SW forced
2	CTRCNT = 0
3	CTRCNT = CMPB
4	CTRCNT = CMPA
5	CTRCNT = PRD

Таймер работает в режиме UP-DOWN-COUNT (Таблица 17.32).

**Таблица 17.32. Порядок обработки событий при работе в UP-DOWN-COUNT режиме**

	Счетчик инкрементируется	Счетчик декрементируется
1	SW forced	SW forced
2	CTRCNT = CMPB	CTRCNT = CMPB
3	CTRCNT = CMPA	CTRCNT = CMPA
4	CTRCNT = 0	CTRCNT = PRD

Пример формирования выходных сигналов OUTA и OUTB при работе счетчика в режиме UP-DOWN (Рисунок 17.17).



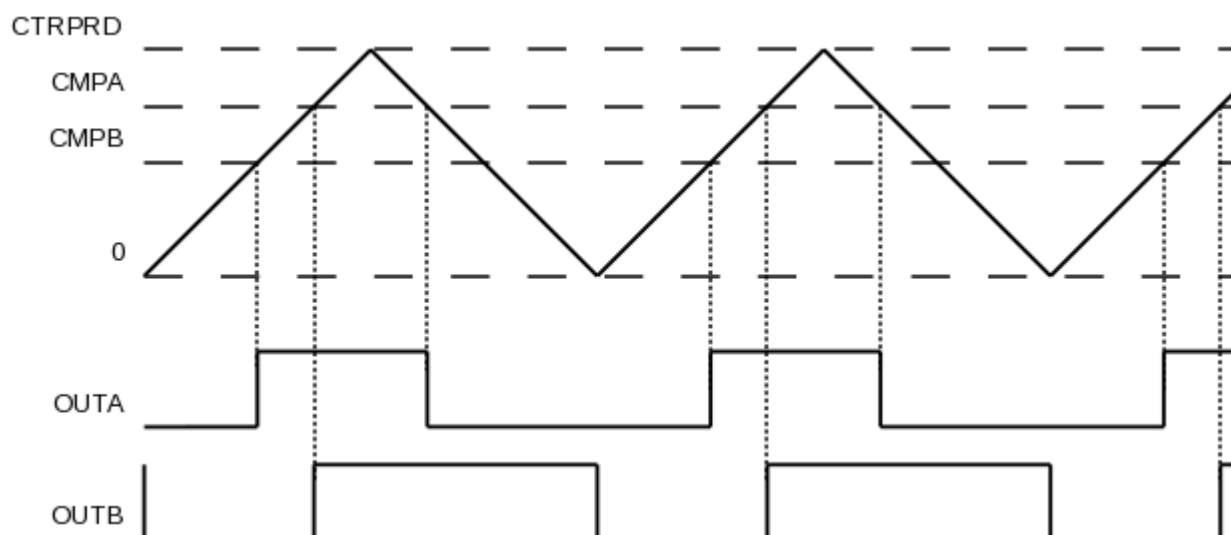


Рисунок 17.17. Пример формирования выходных сигналов OUTA и OUTB

### 17.3.5 Генератор запретной зоны (Dead zone)

Блок реакции на события позволяет достаточно гибко управлять формой выходного сигнала, но для формирования определенной задержки после фронта сигнала более удобно использовать генератор запретной зоны.

Функциональная схема блока представлена на Рисунок 17.18.

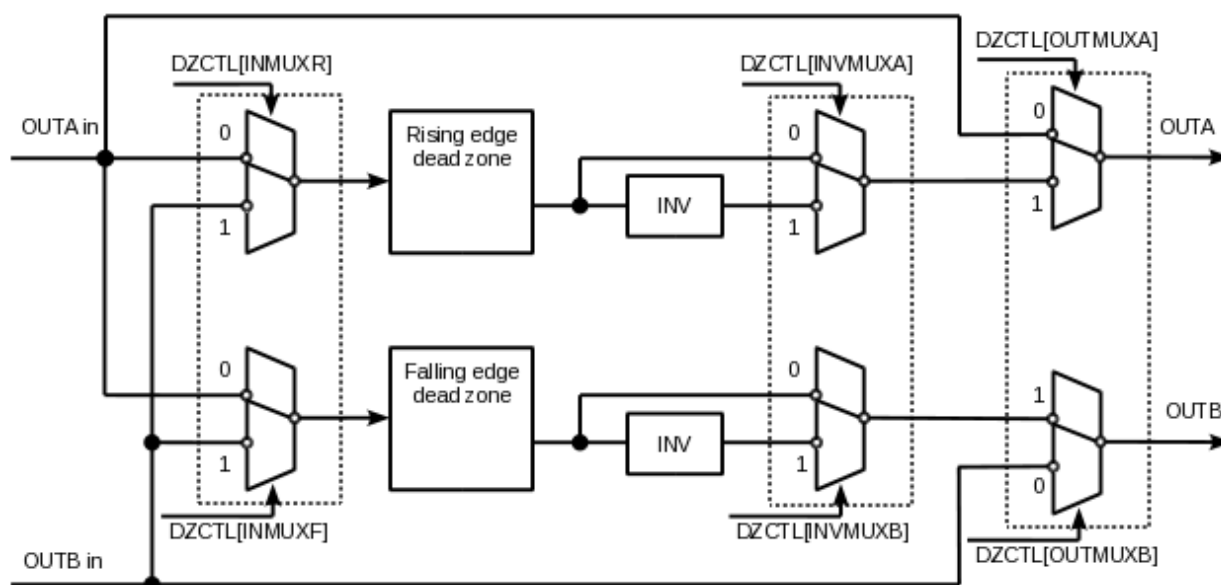
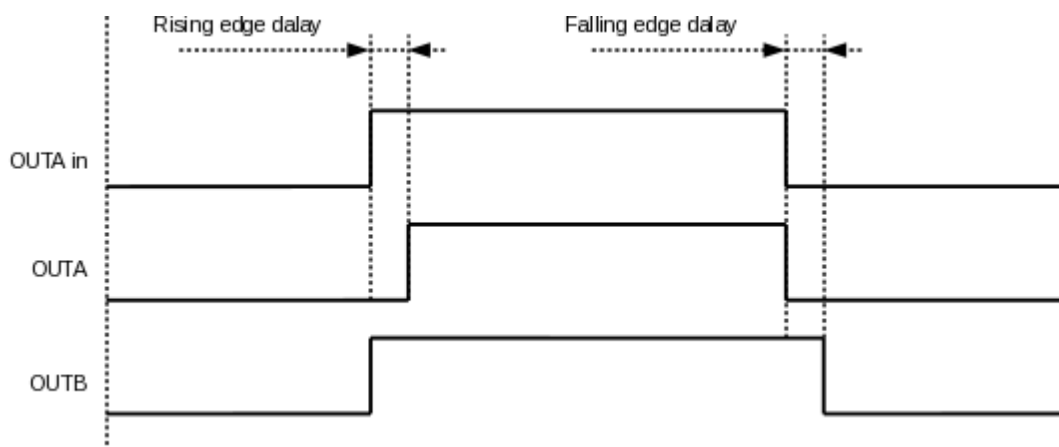


Рисунок 17.18. Функциональная схема блока генерации запретной зоны

Основная функция блока — формирование «мертвой» зоны (формирование выходного сигнала со сдвигом фронта относительно входного сигнала) после переднего, заднего или обоих фронтов.

Для управления режимом работы блока используется регистр DZCTL. Задержки после переднего и заднего фронтов задаются в регистрах DZRPER и DXFPER соответственно.

Пример генерации запретной зоны для сигнала «OUTA in» от переднего и заднего фронтов без выходной инверсии (Рисунок 17.19).



**Рисунок 17.19. Пример генерации запретной зоны**

Величину задержки после фронта можно определить по формуле:

$T_{CLKEN} * DZRPER$  — для переднего фронта (rising edge delay);

$T_{CLKEN} * DZFPER$  — для заднего фронта (falling edge delay)

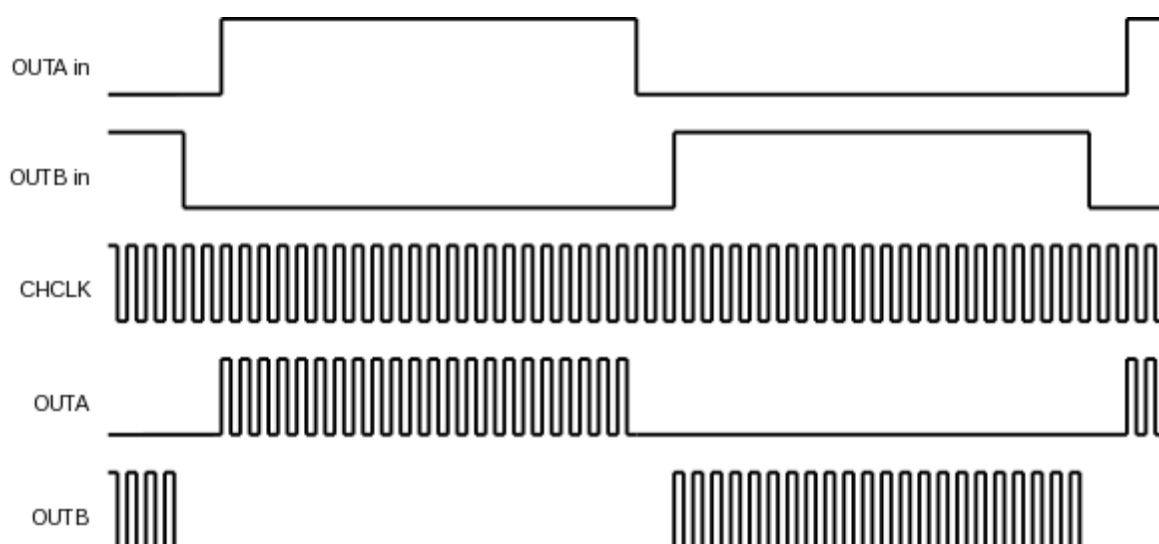
### 17.3.6 Блок дробления выходного сигнала Chopper

Основная функция блока — наложение на выходной сигнал дробящей частоты.

Поддерживаются следующие возможности:

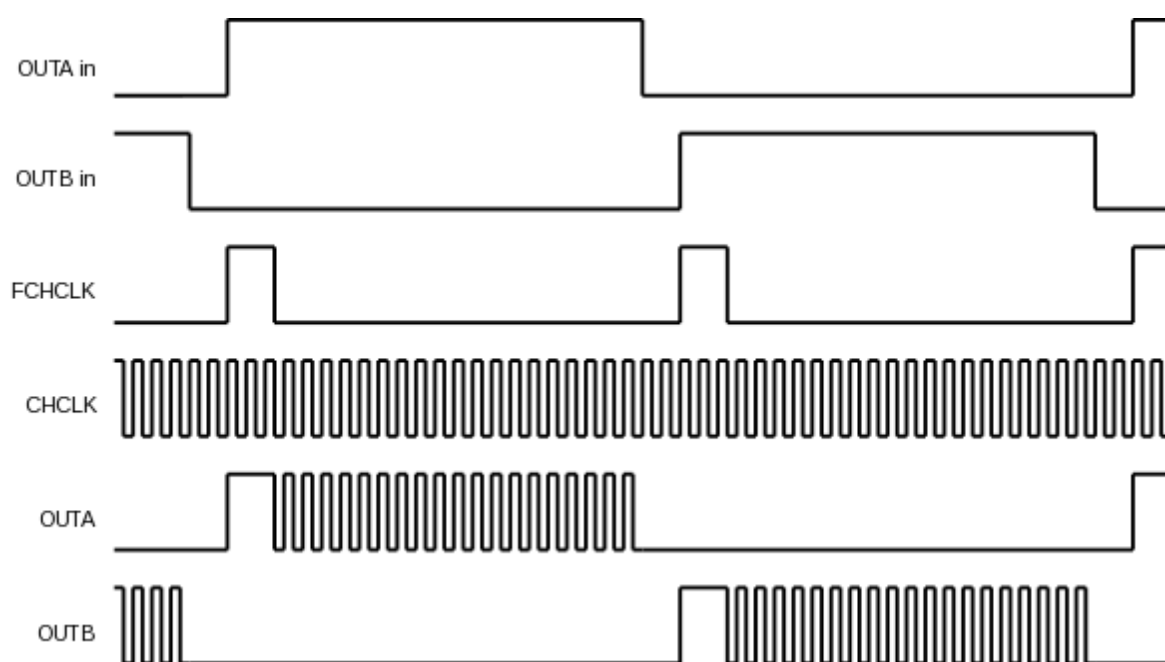
- управление длительностью первого импульса (CHCTL[FIRSTWTH]);
- управление периодом дробящей частоты (CHCTL[CHCLKDIV]);
- управление скважностью дробящей частоты (CHCTL[CHDUTY]);
- выключение данной функции.

Пример простого наложения дробящей частоты (Рисунок 17.20).



**Рисунок 17.20. Пример наложения дробящей частоты**

Пример наложения дробящей частоты и первого импульса (Рисунок 17.21).



**Рисунок 17.21. Пример наложения дробящей частоты и первого импульса**

### 17.3.7 Блок реакции на внешнее воздействие Trip unit

В следующем разделе описываются возможные операции и конфигурационные параметры блока Trip unit.

Основная функция блока — анализ входных воздействий по линиям TU и реакция на них путем изменения сигналов OUTA и OUTB, а также формирование запроса на прерывание процессора по каналу PWM\_TU\_INT.

Входные сигналы TU0 — TU7 активны низким уровнем. При установке на одном из данных входов «0» блок выполняет определенное действие. Каждый канал PWM может быть индивидуально настроен на каждый из TU\* сигналов. Какие линии TU будут использоваться конкретным каналом PWM, программируется в регистре TUSEL этого блока. TU сигнал может быть как синхронным, так и асинхронным относительно CLK.

TU\* вход может быть индивидуально сконфигурирован, чтобы обеспечить как циклическую реакцию на данное событие, так и единичную. Конфигурация определяется битами TUSEL[ONE] и TUSEL[MULT].

Независимо от того в какой конфигурации мы работаем реакция на событие производится в соответствии с данными в регистре TUCTL (Таблица 17.33).

**Таблица 17.33. Возможная реакция на внешнее воздействие**

TUCTL[TUA] и/или TUCTL[TUB]	OUTA и/или OUTB
0h	High-impedance
1h	Подтянуть к 1
2h	Подтянуть к 0
3h	Нет действия

В режиме реакции на каждое событие (MULT) устанавливается флаг данного события TUSTS[MULT], а также генерируется прерывание PWM\_TU\_INT если оно разрешено в регистре TUINTM[MULT].

Данное состояние автоматически сбрасывается при достижении счетчика нулевого значения CTRCNT=0 если на TU\* к этому времени восстановилось единичное значение. Таким образом, в данном режиме событие сброса очищается каждый цикл счетчика PWM. TUSTS[MULT] также может быть сброшен записью 1 в TUCLR[MULT], при условии, что на TU\* к этому времени восстановилось единичное значение. Если на входе TU присутствует 0, то очистить бит TUSTS[MULT] невозможно.

В режиме однократной реакции на событие (ONE) устанавливается флаг данного события TUSTS[ONE], а также генерируется прерывание PWM\_TU\_INT если оно разрешено в регистре TUINTM[ONE]. Реакция на данное событие может быть сброшена только вручную записью 1 в TUCLR[ONE].

Запись единичных значений в поля ONE и MULT регистра TUFRC позволяет имитировать возникновение событий ONE и MULT соответственно.

### 17.3.8 Блок формирования запроса прерывание процессора Interrupt control

Формирует сигнал прерывания PWN\_INT при возникновении различных событий. При помощи регистра TUSEL можно выбрать, на какое из следующих событий реагировать:

- $CTRCNT = 0$ ;
- $CTRCNT = CTRPRD$ ;
- $CTRCNT = CMPA$  при инкременте счетчика;
- $CTRCNT = CMPA$  при декременте счетчика;
- $CTRCNT = CMPB$  при инкременте счетчика;
- $CTRCNT = CMPB$  при декременте счетчика.

В зависимости от периода прерывания записанного в регистре ICCTL[EVENTPRD] происходит реакция:

- на каждое событие;
- на каждое второе событие;
- на каждое третье событие.

При необходимости возможно программное формирование прерывания при помощи регистра ICFRC.

## 18. QSPI: ИНТЕРФЕЙС ПАМЯТИ QUAD SPI

### 18.1 Основные особенности

- Режим работы Master
- Последовательный приём и передача от 4 до 32 бит
- Поддержка полнодуплексного режима работы
- Поддержка полудуплексного режима работы
- Программное управление частотой SCLK
- Quad – четырёхбитный режим работы
- Dual – двухбитный режим работы
- Буфер передачи FIFO объемом 32 слова
- Буфер приема FIFO объемом 32 слова
- Управление прерываниями
- Возможность выбора режимы LSB или MSB
- Интерфейс прямого доступа к памяти DMA
- Функционал исполнения на месте (XIP) под несколько общепринятых FLASH устройств

### 18.2 Функциональное описание

#### 18.2.1 Канал данных SPI

Блок QSPI поддерживает режим передачи данных как master-устройство.

Данные упорядочиваются и передаются (начиная либо с младшего, либо со старшего бита) через стандартный четырёхпроводной интерфейс шины SPI.

Данные передаются синхронно по протоколу MOSI (вывод Master-устройства, ввод Slave-устройства) в соответствии с тактовым сигналом SCLK, генерируемым master-устройством.

Master-устройство также в полнодуплексном режиме получает данные по сигналу MISO (ввод Master-устройства, вывод Slave-устройства).

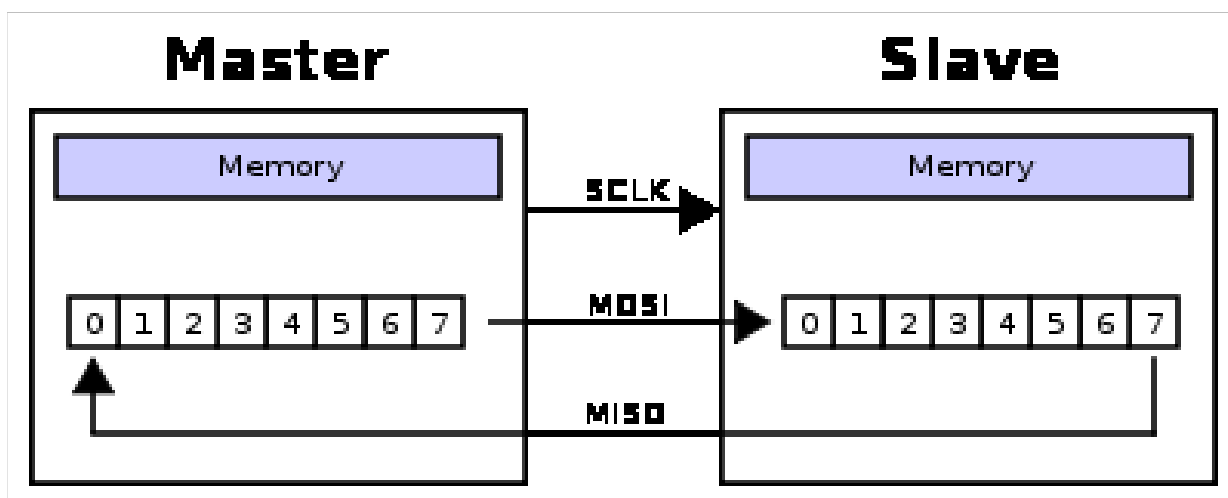


Рисунок 18.1. Передача данных между Master и Slave-устройствами

В Quad-режиме, четыре бита являются одновременно выводом и вводом, и старший бит - всегда QSPI\_IO3. В Dual (двухбитном) режиме, два бита являются одновременно выводом и вводом, и старший бит - всегда QSPI\_IO1. В однобитных операциях используются только QSPI\_IO0.

## 18.2.2 Работа DMA

Передача данных из/в FIFO приема/передачи возможна с помощью системного контрольного DMA, контроллер DMA действует как регулятор потока. Более подробно алгоритм работы с DMA описан в главе 23.

Загрузка буфера передач может начинаться, когда буфер передач пуст до или ниже своего предельного уровня (watermark level). Предельный уровень TX программируем и устанавливается через регистр управления. Контроллер DMA является тем устройством, которое отслеживает, чтобы записи пакета передачи не превышали объём буфера передачи.

Разгрузка буфера приема может начинаться, когда буфер приема заполнен до или выше своего предельного уровня (watermark level) или когда буфер приёма не пуст. Предельный уровень RX программируем и устанавливается через регистр управления.

Контроллер DMA может реализовать единичное чтение когда dmaSreq[1] имеет высокий уровень, или ожидать, когда dmaBreq[1] станет высоким, чтобы реализовать чтение пакета. Следует отметить что dmaSreq[1]=1'b1 указывает, что буфер приема не пуст, в то время как dmaBreq[1]=1'b1 указывает что буфер приёма заполнен выше запрограммированного предельного уровня. Контроллер DMA отвечает за реализацию соответствующей передачи чтения, которая не оставит RX FIFO недозаполненным.

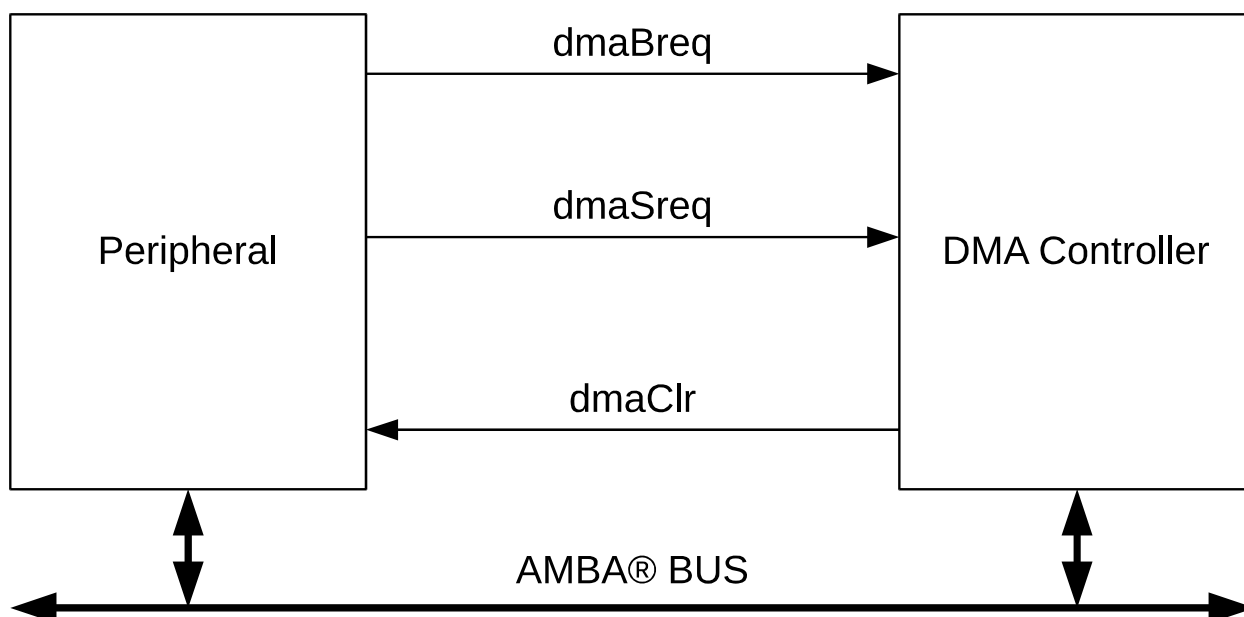


Рисунок 18.2. Работа DMA

### 18.2.3 Варианты подключения Slave устройств

К контроллеру QSPI может быть подключено одно slave-устройство SPI.

На Рисунок 18.3 показано взаимодействие устройств в режиме master/slave, где master инициирует передачу кадра данных.

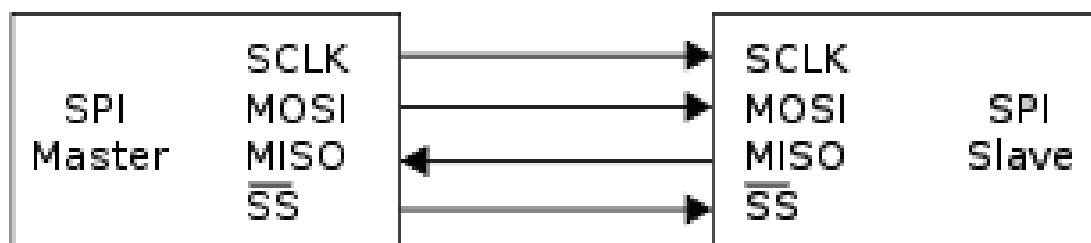


Рисунок 18.3. Коммутация Master - Slave устройств

### 18.2.4 XIP режим

Режим XIP - режим, который позволяет Master-устройству SoC напрямую считывать содержимое подключенной FLASH-памяти просто путём чтения из пространства адресов контроллера QSPI. Режим XIP следует задействовать только если контроллер QSPI работает с одним из стандартных поддерживаемых семейств FLASH-памяти QSPI, приведённых в Таблица 18.1.



**Таблица 18.1. Семейства поддерживаемых памятей**

Изготовитель	Семейство FLASH
Winbond	W25Q
Macronix	MX25L
Spansion	S25FL
Micron	N25Q
Adesto	AT25S
GigaDevice	GD25V

В режиме XIP контроллер QSPI функционирует как промежуточный преобразователь (мост) между мастером SoC к QSPI, переводя обращения на чтения по системной шине в одну из 6 поддерживаемых команд чтения SPI/Dual SPI/Quad SPI для FLASH-памяти. В таблице ниже перечислены поддерживаемые команды чтения и вариации.

**Таблица 18.2. Команды чтения в режиме XIP**

Название команды	Код команды	Поддержка высокопроизводительного режима
Read	0x03	Нет
Fast Read	0x0B	Нет
Fast Read Dual Out	0x3B	Нет
Fast Read Quad Out	0x6B	Нет
Fast Read Dual I/O	0xBB	Есть
Fast Read Quad I/O	0xEB	Есть

В режиме XIP встроенная FSM реализует запрошенную команду чтения через управление набором регистров и буферов FIFO контроллера QSPI. При подключении FSP XIP начнёт считывать содержимое устройства с FLASH-памятью, начиная с адреса 0. В отсутствие любых обращений на чтения по системной шине, FSM продолжит считывать содержимое последующих FLASH-адресов кусочками по 32-бита, до тех пор пока буфер RX FIFO не заполнится наполовину.

Если в режиме XIP приходит запрос на запись FLASH-памяти такая запись игнорируется и на системную шину возвращается признак ошибки.

### 18.2.5 Соотношение тактовых сигналов

Частота QSPI должна быть меньше или равна частоте SYSCLK.

### 18.2.6 Дополнительные возможности по настройке режима работы

Чтобы обеспечить гибкость при настройке режима работы в блоке реализован «Регистр настройки XIP режима»

XIP\_CFG\_HP\_MODE содержит описание командного байта режима XIP. В Dual или Quad режимах ввода/вывода, после того, как адрес передан, устройства flash-памяти QSPI

ожидают командного байта, который содержит указание, должно ли устройство дальше работать в стандартном или высокопроизводительном режиме. Командный байт разнится для разных устройств, таким образом, значение этого регистра используется чтобы передать командный байт, соответствующий устройству Flash-памяти, используемому в данный момент. Когда ядро QSPI выходит из XIP-режима, оно так же автоматически выходит из режима высокой производительности, так, чтобы в следующее обращение к устройству оно уже находилось в стандартном режиме работы.

XIP\_CFG\_DUMMY\_CYCLES определяет количество пустых тактов. В Dual или Quad режимах ввода/вывода быстрого чтения, на высоких частотах, устройствам flash-памяти QSPI требуется несколько пустых тактов для обращения к данным и возвращения их ядру QSPI. Программирование этого регистра позволяет согласовать количество пустых тактов контроллера и устройства flash-памяти QSPI.

В режимах быстрого чтения, быстрого чтения с Dual выводом и быстрого чтения с Quad выводом, ядро QSPI может быть сконфигурировано на 8, 16 тактов сигнала.

В Dual режиме ввода/вывода быстрого чтения, ядро QSPI может быть сконфигурировано на 4, 8, 12 или 16 тактов сигнала.

В Quad режиме ввода/вывода быстрого чтения, ядро QSPI может быть сконфигурировано на 2, 4, 6, 8, 12, 14 или 16 тактов сигнала.

XIP\_CFG\_END\_DUMMY определяет количество пустых тактов, необходимых для прекращения высокопроизводительного режима. Чтобы определить количество необходимых пустых тактов для выхода из высокопроизводительного состояния для передачи (0xEB) в Quad режиме быстрого чтения, применяется следующее уравнение:

пустые такты сигнала SPI =  $2 * (4 + \text{XIP\_CFG\_ADDR4} + \text{XIP\_CFG\_HP\_END\_DUMMY})$

### 18.2.7 Полудуплексный режим

Следует отметить что буфер передачи TX FIFO не должен быть пустым одновременно с тем, как Master-устройство запускает тактовый сигнал, выбор slave-устройства и т.д. Это выполняется всегда, даже когда шину SISO запускает внешнее slave-устройство. В случае, когда Master-устройство исполняет чтение с внешнего slave-устройства, система Master ответственна за запись пустых данных в буфер передачи TX FIFO на протяжении "чтения" в дополнение к чтению валидных данных, полученных от slave-устройства. Если буфер TX FIFO пуст, Master-устройство не переключит QSPI\_SCK и не установит QSPI\_SS.

## 18.3 Описание регистров

Перечень регистров приведен в Таблица 18.3.

**Таблица 18.3. Перечень регистров QSPI.**

Условное обозначение	Название регистра	Исходное состояние	Тип доступа	Смещение
TX_DATA	Регистр последовательной передачи данных (в Tx FIFO)	0	WO	0x00
RX_DATA	Регистр последовательного приема данных (из Rx FIFO)	0	RO	0x04
-	Зарезервировано	-	-	0x08
CTRL	Регистр управления	-	RW	0x0C
CTRL_AUX	Вспомогательный регистр управления	0	RW	0x10
STAT	Регистр статуса	-	RW	0x14
SS	Регистр переключения slave-устройства	0	RW	0x18
SS_POL	Регистр полярности переключения Slave-устройства	0	RW	0x1C
INTR_EN	Регистр подключения прерывания	0	RW	0x20
INTR_STAT	Регистр статуса прерывания	0	RO	0x24
INTR_CLR	Регистр очистки прерывания	0	Wclr	0x28
TX_FIFO_LVL	Регистр уровня буфера передачи TX FIFO	0	RO	0x2C
RX_FIFO_LVL	Регистр уровня буфера приема RX FIFO	0	RO	0x30
-	Зарезервировано	-	-	0x34
TRANS_DELAY	Регистр задержки между передачами Master-устройства	0	RW	0x38
ENABLE	Регистр подключения	0	RW	0x3C
-	Зарезервировано	-	-	0x40 – 0x7C
XIP_CFG	Регистр настройки XIP режима	0	RW	0x80

### 18.3.1 Регистр последовательной передачи данных (в Tx FIFO)

Смещение 0x00.

**Таблица 18.4. Регистр последовательной передачи данных (в Tx FIFO)**

Номер бита	Условное обозначение	Назначение
[31:0]	data reg	Регистр последовательной передачи данных. Попытки упаковки передаваемых данных не предпринимаются. Таким образом, количество переданных на элемент FIFO битов определяется значением “bitsize” вспомогательного регистра управления. К тому же, значение “msb1st” в регистре управления определяет, какие биты элемента FIFO используются: старшие или младшие.

### 18.3.2 Регистр последовательного приема данных (из Rx FIFO)

Смещение 0x04.

**Таблица 18.5. Регистр последовательного приема данных (из Rx FIFO)**

Номер бита	Условное обозначение	Назначение
[31:0]	data reg	Регистр последовательного приема данных. Попытки упаковки принимаемых данных не предпринимаются. Таким образом, количество принятых на элемент FIFO битов определяется значением “bitsize” вспомогательного регистра управления. (например, bitsize = 011 означает, что используется только 16 бит).

### 18.3.3 Регистр управления

Смещение 0x0C.

**Таблица 18.6. Регистр управления**

Номер бита	Условное обозначение	Назначение
[15:14]	txWmarkSet	Задаёт предельный уровень для работы TX FIFO 00 = 4 01 = 8 10 = 16 11 = 24
[13:12]	rxWmarkSet	Задаёт предельный уровень для работы RX FIFO 00 = 4 01 = 8 10 = 16 11 = 24
[11]	mWaitEn	Задержка между передачами. 0=отключение задержки (полная скорость), 1=подключение задержки. См. mWait.
[10]	Dma	Установить 1 для подключения режима DMA.
[9:6]	Зарезервировано	Зарезервировано
[5]	Master	Master-режим работы шины порта SPI; 0 = резерв; 1 = master-режим.
[4]	Cpol	0 = тактовый сигнал SPI остается низким, 1 = тактовый сигнал SPI остается высоким.
[3]	Cpha	0 = первые биты QSPI_IO запускаются при установке QSPI_SS. 1 = QSPI_IO запускается на первом фронте тактового сигнала QSPI_SCK после установки QSPI_SS.

Номер бита	Условное обозначение	Назначение
[2]	msb1st	<p>MSB/LSB</p> <p>1=старший бит (MSB) первый, 0=младший бит (LSB) первый.</p> <p>Замечание: в Quad или Dual режиме, эта установка влияет только на порядок (4-битного или 2-битного) фрагмента передачи, а не на порядок битов всего пакета;</p> <p>QSPI_IO3 всегда являются старшим битом (MSB) фрагмента данных в Quad режиме,</p> <p>QSPI_IO1 всегда являются старшим битом (MSB) фрагмента данных в Dual режиме</p>
[1]	Зарезервировано	Зарезервировано
[0]	contXfer	<p>Бит непрерывной передачи.</p> <p>0 = между последовательными передачами бит QSPI_SS становится неактивным.</p> <p>1 = бит QSPI_SS остается активен до тех пор, пока TX FIFO не опустеет и пока бит contXferExtend (во вспомогательном регистре управления) не будет установлен на низкий уровень.</p> <p>Значение по сбросу: 1</p>

### 18.3.4 Вспомогательный регистр управления

Смещение 0x10.

Вспомогательный регистр управления может быть безопасно переконфигурирован только когда TX FIFO пуст и никаких передач не происходит.

**Таблица 18.7. Вспомогательный регистр управления**

Номер бита	Условное обозначение	Назначение
[7]	contXferExtend	<p>Бит продления непрерывной передачи.</p> <p>Не имеет эффекта если бит contXfer в регистре управления равен 0.</p> <p>1 = QSPI_SS остается активен даже когда TX FIFO пуст.</p> <p>0 = QSPI_SS становится неактивен когда TX FIFO пуст.</p>
[6:4]	Bitsize	<p>Bitsize = (#количество битов в передаче+1)*4. Например:</p> <p>000 = 4-битный последовательный режим</p> <p>001 = 8-битный последовательный режим</p> <p>010 = 12-битный последовательный режим</p> <p>...</p> <p>111 = 32-битный последовательный режим</p> <p>Значение по сбросу: 111</p>

Номер бита	Условное обозначение	Назначение
[3]	inhibitDin	Установка 1 запрещает сериализаторам SPI запись в FIFO данных чтения.
[2]	Зарезервировано	Зарезервировано
[1:0]	Qmode	Режимы Spi: 11 = Quad SPI, 10 = Dual SPI, 01 = зарезервировано, 00 = стандартный SPI.

### 18.3.5 Регистр статуса

Смещение 0x14

**Таблица 18.8. Регистр статуса**

Номер бита	Условное обозначение	Назначение
[7]	rxFull	Флаг полного FIFO приема
[6]	rxWmark	Флаг предельного уровня заполнения FIFO приема
[5]	rxEmpty	Флаг пустого FIFO приема Значение по сбросу: 1
[4]	txFull	Флаг полного FIFO передачи
[3]	txWmark	Флаг предельного уровня заполнения FIFO передачи Значение по сбросу: 1
[2]	txEmpty	Флаг пустого FIFO передачи Значение по сбросу: 1
[1]	Зарезервировано	Зарезервировано
[0]	xferIP	Передача в процессе исполнения

### 18.3.6 Регистр переключения slave-устройства

Смещение 0x18.

**Таблица 18.9. Регистр переключения slave-устройства**

Номер бита	Условное обозначение	Назначение
[0]	QSPI_SS	0 = не выбирать slave-устройство для TX/RX. 1 = выбрать Slave-устройства для TX/RX.

Замечание: сигнал вывода QSPI\_SS переключится в активное состояние когда этот модуль подключен и его TX FIFO не пуст. В бите contXfer регистра управления и бите contXferExtend вспомогательного регистра управления приведено описание того, как и когда QSPI\_SS возвращается в неактивное состояние. Наконец, регистр полярности переключения Slave-устройства задаёт значение “активного” состояния - 1 или 0.

### 18.3.7 Регистр полярности переключения Slave-устройства

Смещение 0x1C.

**Таблица 18.10. Регистр полярности переключения Slave-устройства**

Номер бита	Условное обозначение	Назначение
[0]	Sspol0	0 = QSPI_SS – активный низкий сигнал. 1 = QSPI_SS – активный высокий сигнал.

### 18.3.8 Регистр подключения прерывания

Смещение 0x20.

Запись 1 в этот регистр подключает отдельные прерывания.

Чтение из этого регистра возвращает статус подключенных прерываний.

**Таблица 18.11. Регистр подключения прерывания**

Номер бита	Условное обозначение	Назначение
[4]	XFERDONEPULSE	Переключение QSPI_SS из активного в неактивное
[3]	RXFULLPULSE	RX FIFO полон
[2]	RXWMARKPULSE	RX FIFO заполнен до предельного уровня
[1]	TXWMARKPULSE	TX FIFO заполнен до предельного уровня
[0]	TXEMPTYPULSE	TX FIFO пуст

Замечание: актуальный статус флагов FIFO следует смотреть в регистре статусов.

### 18.3.9 Регистр статуса прерывания

Смещение 0x24.

1 = источник прерываний активен,

0 = прерываний нет.

**Таблица 18.12. Регистр статуса прерывания**

Номер бита	Условное обозначение	Назначение
[4]	XFERDONEPULSE	Переключение QSPI_SS из активного в неактивное
[3]	RXFULLPULSE	RX FIFO полон
[2]	RXWMARKPULSE	RX FIFO заполнен до предельного уровня
[1]	TXWMARKPULSE	TX FIFO заполнен до предельного уровня
[0]	TXEMPTYPULSE	TX FIFO пуст

Замечание: актуальный статус флагов FIFO следует смотреть в регистре статусов.

### 18.3.10 Регистр очистки прерывания

Смещение 0x28.

Очистка прерывания:

- запись 1 очищает бит прерывания.

- запись 0 не оказывает эффекта.

**Таблица 18.13. Регистр очистки прерывания**

Номер бита	Условное обозначение	Назначение
[4]	XFERDONEPULSE	Переключение QSPI_SS из активного в неактивное
[3]	RXFULLPULSE	RX FIFO полон
[2]	RXWMARKPULSE	RX FIFO заполнен до предельного уровня
[1]	TXWMARKPULSE	TX FIFO заполнен до предельного уровня
[0]	TXEMPTYPULSE	TX FIFO пуст

Замечание: актуальный статус флагов FIFO следует смотреть в регистре статусов.

### 18.3.11 Регистр уровня буфера передачи TX FIFO

Смещение 0x2C.

**Таблица 18.14. Регистр уровня буфера передачи TX FIFO**

Номер бита	Условное обозначение	Назначение
[5:0]	tx_FIFO_level	Чтение текущего уровня заполнения буфера передачи FIFO. (0-32).

### 18.3.12 Регистр уровня буфера приема RX FIFO

Смещение 0x30.

**Таблица 18.15. Регистр уровня буфера приема RX FIFO**

Номер бита	Условное обозначение	Назначение
[5:0]	rx_FIFO_level	Чтение текущего уровня заполнения буфера приема FIFO. (0-32).

### 18.3.13 Регистр задержки между передачами Master-устройства

Смещение 0x38.

**Таблица 18.16. Регистр задержки между передачами Master-устройства**

Номер бита	Условное обозначение	Назначение
[7:0]	mWait	Если master (в регистре управления) имеет значение 1, этот регистр задает задержку между передачами для сериализатора master-устройства. Задержка равна (mwait+1) циклам QSPI_SCK.

### 18.3.14 Регистр подключения

Смещение 0x3C.



**Таблица 18.17. Регистр подключения**

Номер бита	Условное обозначение	Назначение
[0]	enableReq	<p>Подключение SPI.</p> <p>0 = отключение приема и передачи SPI; сброс буферов FIFO RX и TX при переключении этого бита от 1 к 0. Хотя SPI может быть отключен в любое время, рекомендуется делать это, когда буферы FIFO пусты. После записи 0 в бит подключения, следует перепроверить бит подключения; только после того, как получено чтение того, что бит подключения равен 0, можно производить любые дальнейшие конфигурирования регистров.</p> <p>1 = подключение приема и передачи SPI.</p>

### 18.3.15 Регистр настройки XIP режима

Смещение 0x80.

**Таблица 18.18. Регистр настройки XIP режима**

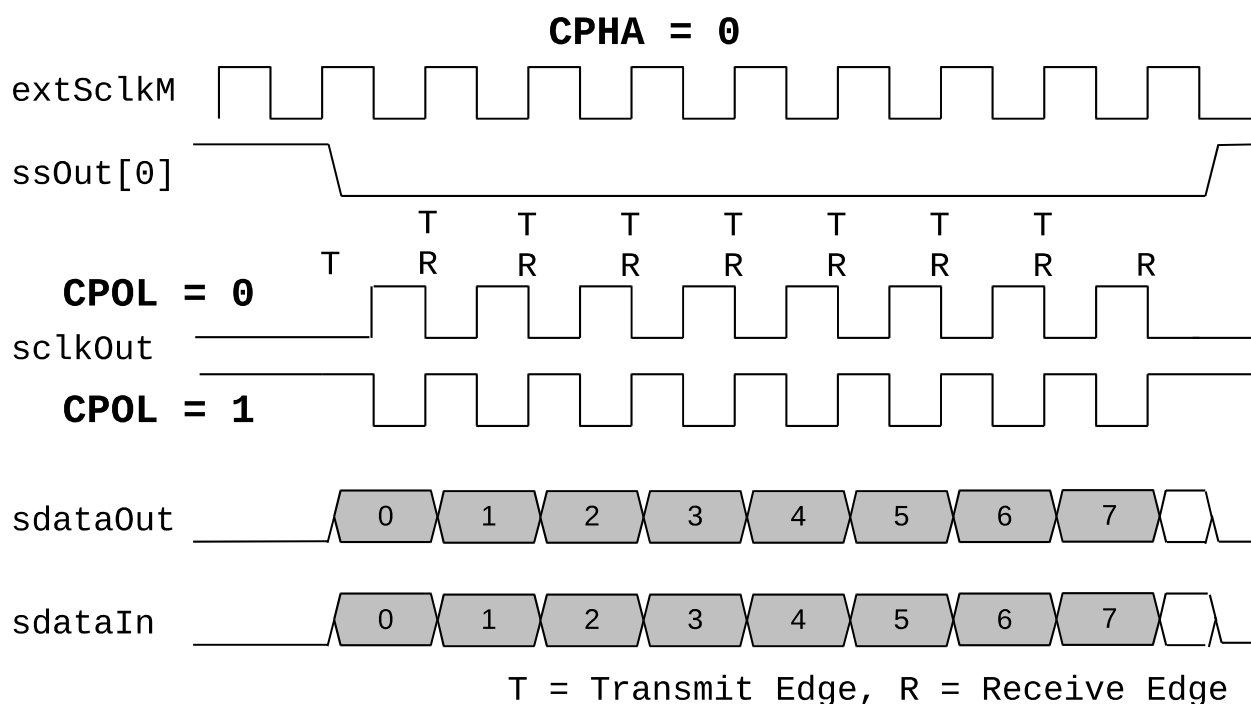
Номер бита	Условное обозначение	Назначение
[7:0]	XIP_CFG_CMD	<p>Тип команды чтения конфигурации XIP.</p> <p>8'h03=чтение,  8'h0B=быстрое чтение,  8'h3B=быстрое чтение, Dual вывод,  8'h6B=быстрое чтение, Quad вывод,  8'hBB=быстрое чтение, Dual ввод/вывод,  8'hEB=быстрое чтение, Quad ввод/вывод,  все другие зарезервированы.</p>
[8]	XIP_CFG_HPEN	<p>Режим высокой производительности конфигурации XIP.</p> <p>1'b0=высокопроизводительный режим отключен,  1'b1=высокопроизводительный режим подключен.</p>
[9]	XIP_CFG_CPNA	<p>Режим CPNA конфигурации XIP</p> <p>1'b0=режим CPNA 0,  1'b1=режим CPNA 1</p>
[10]	XIP_CFG_CPOL	<p>Режим CPOL конфигурации XIP</p> <p>1'b0=режим CPOL 0,  1'b1=режим CPOL 1</p>
[11]	XIP_CFG_ADDR4	<p>4-байтовый режим адресов конфигурации XIP.</p> <p>1'b0=XIP выпускает 3-байтовый адрес SPI,  1'b1=XIP выпускает 4-байтовый адрес SPI.</p>
[12]	XIP_CFG_LE32	<p>32-битная организация данных чтения в конфигурации XIP от младшего к старшему.</p> <p>1'b0=данные чтения организованы от старшего к младшему,  1'b1=данные чтения – 32-битные, от младшего к старшему.</p>
[15:13]	-	Резерв
[23:16]	XIP_CFG_HP_MOD E	<p>Командный байт режима XIP.</p> <p>Этот байт передается от ядра QSPI к флеш-модулю QSPI. Это значение, заданное разными флеш модулями QSPI, которое следует вводить в их регистры управления для активации режима оптимизации производительности в режимах Dual и Quad ввода/вывода.</p>

Номер бита	Условное обозначение	Назначение
[27:24]	XIP_CFG_DUMMY_CYCLES	Используется для конфигурирования количества пустых тактов, используемых ядром Qspi. Для режимов быстрого чтения, быстрого чтения с Dual и с Quad выводом, количество пустых тактов равно данному значению, умноженному на 8. Быстрое чтение с Dual вводом/выводом: количество пустых тактов равно данному значению, умноженному на 4 плюс еще 4. Быстрое чтение с Quad вводом/выводом: количество пустых тактов равно данному значению, умноженному на 2 плюс еще 2.
[29:28]	XIP_CFG_HP_END_DUMMY	Это значение используется для конфигурирования числа пустых тактов, необходимых для прекращения высокопроизводительного режима. Оно используется ядром при выходе из режима XIP после исполнения высокопроизводительных передач, чтобы удостовериться, что устройство FLASH-памяти не находится всё ещё в высокопроизводительном режиме. Для Dual режима, это значение умножается на 4, чтобы получить нужное реализованное количество пустых тактов. Для Quad режима это значение умножается на 2, чтобы получить нужное реализованное количество пустых тактов.
[30]	XIP_EN_REQ	Запрос на подключение XIP. Для подключения режима XIP: установить 1'b1, не делать обращений в область адресов QSPI, не менять сигналы XIP_CFG_X, дожидаться, пока XIP_EN_OUT примет значение 1'b1. Для отключения режима XIP: установить 1'b0, не делать обращений в область адресов QSPI, не менять сигналы XIP_CFG_X, дожидаться, пока XIP_EN_OUT примет значение 1'b0.
[31]	XIP_EN_OUT	Статус подключения XIP 1'b0=Режим XIP отключен, 1'b1=Режим XIP подключен.

## 18.4 Тайминги

### 18.4.1 Передача данных Master-устройством SPI

В большинстве случаев различные устройства SPI могут иметь очень разные оценки для таймингов. По этой причине контроллер QSPI поддерживает стандартные условия для таймингов, ими управляют два бита регистра управления: `srpha` и `сроl`. Эти два бита определяют какие фронты тактового сигнала SPI на каких данных будут переданы и взяты за образец. На Рисунок 18.4 показана работа этих битов при 8-битной передаче данных.



**Рисунок 18.4. Передача данных Master-устройства SPI при CPHA = 0, CPOL = 0, 1**

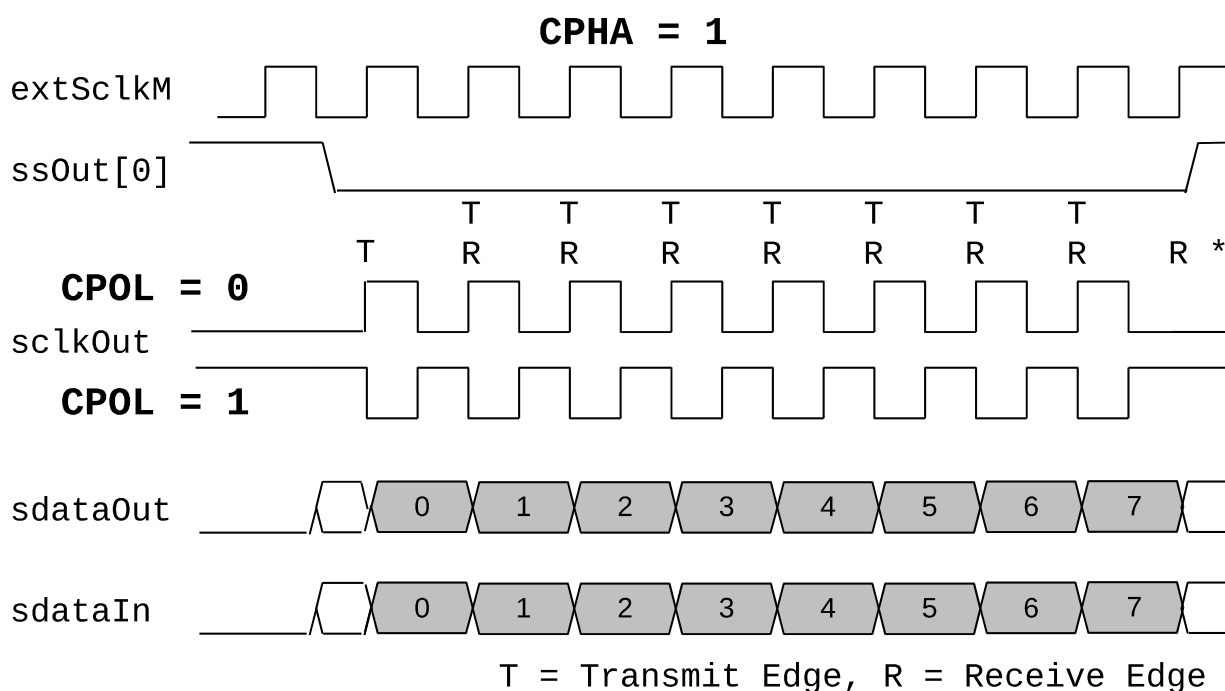
На Рисунок 18.4 входной тактовый сигнал extSclkM внешнего Master-устройства SPI используется для производства sclkOut (QSPI\_SCK). В регистр управления была произведена запись для выбора Master-режима, и вместе с этим сконфигурированы биты cpol и cpha. Регистр полярности выбора Slave-устройства был очищен для обеспечения сигнала выбора Slave-устройства с активным низким уровнем, а в регистр выбора Slave-устройства была установлена 1 для подключения взаимодействия со Slave-устройством 0 на шине SPI. В регистр передачи было записано слово данных. Наконец, для обеспечения работы SPI был установлен бит подключения. Данные передаются на нисходящем фронте sclkOut (CPOL = 0), или на восходящем фронте sclkOut (CPOL = 1). Следует отметить, что для CPHA=0, первый бит данных передается как только (или до того как) ssOut (QSPI\_SS) становится активен.

Для работы Master-устройства из-за неизбежной задержки распространения возвращенных данных от slave-устройства, лучшим местом для взятия образца входящих данных является полный такт sclkOut (QSPI\_SCK) после того, как передача состоялась. sclkOut (QSPI\_SCK) уже является немного запаздывающей версией внутреннего тактового сигнала сериализатора Master-устройства; sclkOut (QSPI\_SCK) также будет иметь добавочную задержку распространения сигнала из-за выхода за пределы микросхемы к slave-устройству.

Slave-устройство использует этот задержанный тактовый сигнал для производства данных, что накапливает дальнейшую задержку при возвращении обратно в логику master-устройства, которая берёт образец этого сигнала.

Данные должны прибыть и остаться стабильными обратно в логику master-устройства за менее чем один полный такт sclkOut (QSPI\_SCK). Такая схема работает куда лучше, чем привязка задержки с круговым маршрутом к половине такта sclkOut (QSPI\_SCK). Следует

отметить, что этот вариант реализуется, если бы master-устройство брало образец данных на восходящем фронте sclkOut (CPOL=0), или на нисходящем фронте sclkOut (CPOL=1).



**Рисунок 18.5. Передача данных Master-устройства SPI при CPHA = 1, CPOL = 0, 1**

На Рисунок 18.5 входной тактовый сигнал extSclkM внешнего Master-устройства SPI используется для производства sclkOut (QSPI\_SCK). В регистр управления была произведена запись для выбора Master-режима, и вместе с этим сконфигурированы биты cpol и cpha. Регистр полярности выбора Slave-устройства был очищен для обеспечения сигнала выбора Slave-устройства с активным низким уровнем, а в регистр выбора Slave-устройства была установлена 1 для подключения взаимодействия со Slave-устройством 0 на шине SPI. В регистр передачи было записано слово данных. Наконец, для обеспечения работы SPI был установлен бит подключения. Данные передаются на восходящем фронте sclkOut (CPOL = 0), или на нисходящем фронте sclkOut (CPOL = 1). Следует отметить, что для CPHA=1, первый бит данных передается на фронте первого такта после того как ssOut (QSPI\_SS) стал активен.

Для работы Master-устройства из-за неизбежной задержки распространения возвращенных данных от slave-устройства, лучшим местом для взятия образца входящих данных является полный такт sclkOut (QSPI\_SCK) после того, как передача состоялась. sclkOut (QSPI\_SCK) уже является немного запаздывающей версией внутреннего тактового сигнала сериализатора Master-устройства; sclkOut (QSPI\_SCK) также будет иметь добавочную задержку распространения сигнала из-за выхода за пределы микросхемы к slave-устройству.

Slave-устройство использует этот задержанный тактовый сигнал для производства данных, что накапливает дальнейшую задержку при возвращении обратно в логику master-устройства, которая берёт образец этого сигнала.

Данные должны прибыть и остаться стабильными обратно в логику master-устройства за менее чем один полный такт `sclkOut` (QSPI\_SCK). Такая схема работает куда лучше, чем привязка задержки с круговым маршрутом к половине такта `sclkOut` (QSPI\_SCK). Следует отметить, что этот вариант реализуется, если бы master-устройство брало образец данных на восходящем фронте `sclkOut` (CPOL=0), или на нисходящем фронте `sclkOut` (CPOL=1).

## 18.4.2 Взаимодействие с DMA

Контроллер QSPI устанавливает запрос `dmaBreq[0]`, когда его буфер передачи FIFO опустошается до предельного уровня, устанавливает `dmaSreq[1]`, когда его буфер приема FIFO не пуст, и `dmaBreq[1]` когда буфер приема FIFO заполняется до своего предельного уровня. Эти сигналы остаются установленными до тех пор, пока контроллер DMA не совершит достаточного количества чтений или записей (на это указывает предельный уровень буферов приема/передачи) и не установит сигнал `dmaClr[x]`. Когда контроллер DMA завершает записи в буфер передачи TX FIFO, он устанавливает `dmaClr[0]`; когда чтения из RX FIFO завершены, контроллер DMA устанавливает `dmaClr[1]`. Когда QSPI обнаруживает `dmaClr`, он деактивирует соответствующие сигналы `dmaBreq` и `dmaSreq`. Процедура подтверждения связи завершается когда контроллер DMA сэмплирует низкий уровень `dmaBreq/dmaSreq` и деактивирует `dmaClr`.

Ниже приведены некоторые дополнительные замечания о сигналах DMA.

QSPI устанавливает `dmaBreq[x]/dmaSreq[x]` только если `dmaClr[x]` имеет низкий уровень, и деактивирует `dmaBreq[x]/dmaSreq[x]`, только если `dmaClr[x]` имеет высокий уровень.

В зависимости от уровня заполненности FIFO приема к моменту процедуры подтверждения связи, `dmaSreq[1]` и `dmaBreq[1]` могут быть установлены одновременно, или `dmaSreq[1]` может быть установлен первым. В стандартном сценарии, в котором FIFO приема начинает работу пустым, `dmaSreq[1]` первым перейдет в высокий активный уровень, что укажет на то, что в FIFO приёма появились какие-то данные. Позднее, `dmaBreq[1]` тоже может перейти в высокий активный уровень, что укажет на то, что FIFO приёма достиг своего предельного уровня заполнения.

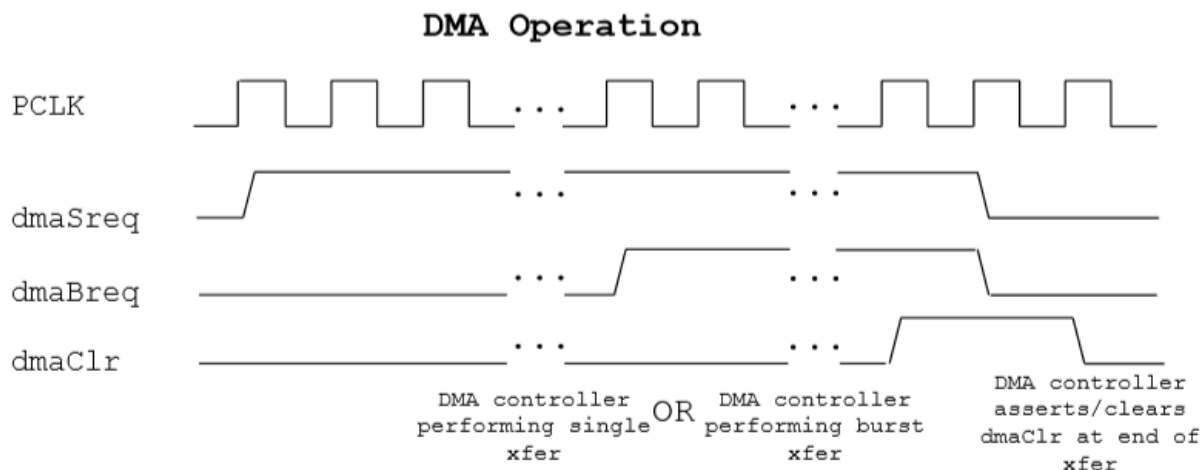


Рисунок 18.6. Сигналы DMA

## 18.5 Программирование

### 18.5.1 Сброс

Запись регистра `ENABLE[enableReq] = 0` сбрасывает порт SPI, а также провоцирует синхронную очистку буферов FIFO передачи и приёма.

### 18.5.2 Режим XIP

Вход и выход в режим XIP должен осуществляться следующим образом:

- 1) В идеале, никакие активные команды на чтение/запись не должны выходить в пространство адресов QSPI в процессе входа или выхода из режима XIP.
- 2) Если режим XIP неактивен, `XIP_EN_REQ` и `XIP_EN_OUT` оба имеют значение 0.
- 3) В регистре `XIP_CFG` производятся любые изменения конфигурации.
- 4) Для входа в режим XIP следует запустить `XIP_EN_REQ = 1`, затем дождаться, когда `XIP_EN_OUT = 1`.
- 5) После того, как установится `XIP_EN_OUT = 1`, обращения на чтение от пространства адресов QSPI будут реализованы в формате одной из нескольких поддерживаемых общепринятых транзакций чтения к устройству FLASH-памяти (Q)SPI. Записи в пространство адресов QSPI в режиме XIP будут возвращать `HRESP = ERROR`.
- 6) Для выхода из режима XIP, следует задать `XIP_EN_REQ = 0`, затем дождаться, когда установится `XIP_EN_OUT = 0`, прежде чем предпринимать любые дальнейшие обращения к пространству адресов QSPI.

### 18.5.3 Последовательность запуска

- Произвести запись в регистр подключения для отключения порта SPI.
- Произвести чтение регистра подключения – он должен возвращать 0, прежде чем можно будет выполнить следующий шаг.
- Обнулить любые прерывания путём записи 'hFF в регистр очистки прерываний.
- Если прерывания используются, подключить прерывания путём записи в регистр подключения прерываний.
- Задать полярность шин переключения путём записи в регистр переключения полярности slave-устройства.
- Выбрать какое slave-устройство должно быть выбрано путём записи в регистр выбора slave-устройства.
- Запрограммировать регистр задержек между передачами.
- Запись в регистр управления:
  - Подключение задержки между передачами.
  - Подключение DMA (dmaEnable).
  - Установка бита cpol.
  - Установка бита cpha.
  - Установка бита msb1st.
  - Подключение непрерывной передачи.
- Запись во вспомогательный регистр управления:
  - Расширение непрерывной передачи.
  - Установка бита bitsize (4, 8, 12, 16, 20, 24, 28, 32 бита).
  - Запрет записей в RX FIFO.
  - Режим Spi (Quad, Dual, Spi).
- Запись в регистр подключения для подключения порта SPI.

Следует отметить, что допускается производить предварительные записи в TX FIFO перед подключением модуля в последнем шаге.

Следует также отметить что дальнейшие записи во вспомогательный регистр управления после подключения этого модуля допускаются, если одновременно выполняется следующее: буфер TX FIFO пуст и процесс передачи не происходит.

### 18.5.4 Последовательность передачи

Загрузка регистра передачи может быть осуществлена разными способами: использование процессора для опроса регистра статуса о состоянии буфера FIFO, использование возможностей прерываний процессора и кода ISR для прерывания когда буфер передачи TX FIFO пуст или заполнен ниже предельного уровня, или использование контроллера DMA (и связанных с ним сигналов) для выполнения необходимых записей в буфер передачи TX FIFO.

Для функционала опроса:

- Следует прочесть регистр уровня FIFO, чтобы определить число байтов/слов, которые возможно записать.
- Записать байты/слова в регистр последовательной передачи данных.
- Прочесть регистр статуса чтобы проверить флаг заполненности буфера передачи TX FIFO перед записью добавочных байтов/слов.

Для функционала прерывания:

- Использовать служебную программу прерывания для того чтобы отправить прерывание когда TX FIFO пуст или заполнен ниже предельного уровня.
- Записать подходящее число слов данных в регистр последовательной передачи данных в служебной программе прерывания.

Для функционала DMA:

- Подключить функционал DMA и позволить подключенному механизму DMA передавать данные в TX FIFO когда dmaBreq[0] установлен в логике DMA.

Для функционала Master-устройства выбор другого slave-устройства осуществляется следующим образом:

- Прочесть регистр статуса чтобы удостовериться что TX FIFO пуст и не занят процессом передачи.



- Сделать запись в регистр выбора Slave-устройства (и регистр выбора полярности Slave-устройства, если необходимо) для выбора другого slave-устройства под следующие транзакции.

### 18.5.5 Последовательность приема

При сигнале о полном FIFO RX/получении прерывания о достижении предельного уровня заполнения FIFO RX или чтении регистра статуса:

- если установлен флаг `gx_not_empty`, тогда следует чтение слов(а) данных из регистра последовательного приема данных.
- если применяются прерывания, следует очистить прерывания путем записи “1” в регистр очистки прерываний.
- если применяется режим DMA, нет необходимости читать регистр статуса. Вместо этого логика DMA установит `dmaSreq[1]` если RX FIFO не пуст, и установит `dmaBreq[1]` когда RX FIFO заполнится до своего предельного уровня. Если только `dmaSreq[1]` будет иметь высокий уровень, то внешний контроллер DMA может реализовать единичные чтения, или может дожидаться, пока `dmaBreq[1]` станет высоким, чтобы реализовать пакетные чтения (пакетные чтения должны соответствовать предельному уровню RX FIFO)

### 18.5.6 Замечания по полудуплексному режиму

Менять направление шины данных QSPI\_IO можно только когда буфер передачи TX FIFO пуст. Следует отметить что `ssOut[x]` заведомо переключается в неактивное состояние между успешными передачами (`contXfer = 0`) или когда уровень TX FIFO пуст (`contXfer = 1` и `contXferExtend = 0`). Для неограниченного расширения периода активности `ssOut[x]`, следует установить `contXfer = 1` и `contXferExtend = 1`. Бит `contXferExtend` во вспомогательном регистре управления может быть переконфигурирован в любое время под планирование окончания активной передачи. Однако, следует соблюдать осторожность и не модифицировать никакие другие биты во вспомогательном регистре управления во время активной передачи.

При «передаче» данных в полудуплексном режиме, желательно запретить записи в буфера приема RX FIFO. И хотя это обычно связано только с полудуплексным режимом, в полнодуплексном режиме тоже могут быть моменты, когда записи в RX FIFO желательно запретить. По этой причине, во вспомогательном регистре управления существует отдельный бит под это (`inhibitDin`).

При “приеме” данных в полудуплексном режиме, необходимо записывать пустые данные в TX FIFO для получения реальных данных от slave-устройства. Пустые данные не передаются на шину SPI (поскольку Master является принимающим устройством), но

используются для отслеживания того, сколько передач “приема” необходимо исполнить. Для каждой записи в TX FIFO, Master-устройство генерирует единичную транзакцию на шину SPI. Система может читать RX FIFO Master-устройства независимо для обращения к полученным данным.

## 19. SMC: КОНТРОЛЛЕР ВНЕШНЕЙ СТАТИЧЕСКОЙ ПАМЯТИ

### 19.1 Основные особенности

SMC (Static Memory Controller) – контроллер внешней статической памяти.

Поддерживаются следующие типы памяти:

- Синхронная или асинхронная SRAM
- Псевдостатическая память (PSRAM)
- NOR Flash память
- NAND Flash память с интерфейсом SRAM

Также SMC поддерживает модули LCD с параллельными интерфейсами типа Intel 8080 и Motorola 6800.

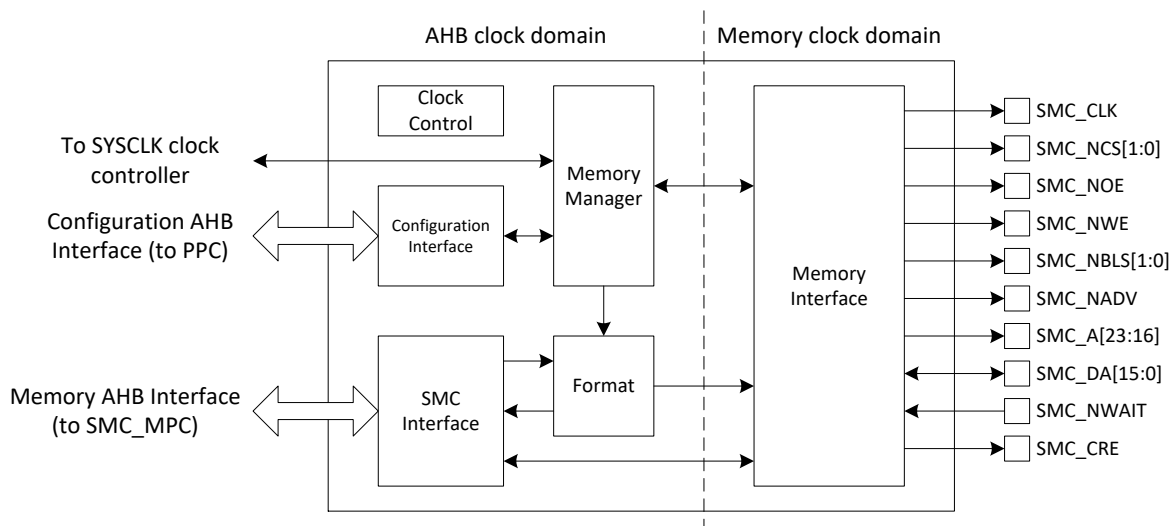
Контроллер SMC имеет следующие характеристики:

- разрядность шины данных – 16 бит
- максимальная разрядность адреса – 24 бита
- мультиплексирование линий адреса и данных
- поддержка подключения до двух микросхем памяти
- емкость каждой микросхемы памяти – до 32 Мбайт
- интерфейс работает на тактовой частоте SYSCLK с возможностью деления на коэффициент 1 – 32
- независимая настройка временных характеристик интерфейса для каждой микросхемы
- поддержка пакетного режима передачи данных
- режим пониженного энергопотребления

## 19.2 Описание функционирования

### 19.2.1 Структурная схема SMC

На следующем рисунке показана упрощенная структурная схема контроллера SMC.



**Рисунок 19.1. Структурная схема SMC**

Контроллер имеет следующие интерфейсы для взаимодействия с системой:

- Конфигурационный АНВ-интерфейс предоставляет доступ к регистрам SMC. Подключается к основной матрице коммутации через блок PPC.
- АНВ-интерфейс памяти для прямого доступа к внешним микросхемам памяти. Подключается к основной матрице коммутации через блок MPC.
- Low-power интерфейс для управления тактовым сигналом АНВ-домена.

Основные подблоки SMC и их назначение:

- Clock Control – формирование и управление тактовым сигналом интерфейса памяти.
- Configuration Interface – блок обработки запросов по конфигурационному АНВ-интерфейсу.
- SMC Interface – блок обработки запросов по АНВ-интерфейсу памяти.
- Memory Manager – обновление регистров настроек интерфейса, управление командами к памяти, управление режимом энергосбережения.
- Блок Format – получает запросы от АНВ-интерфейса памяти и блока Memory Manager (запросы от блока Memory Manager имеют более высокий приоритет), выполняет преобразование в запросы памяти и передает их в блок Memory Interface.
- Memory Interface – содержит буферы FIFO данных и команд, регистры ввода-вывода и управляющую логику.

## 19.2.2 Адресация внешней памяти

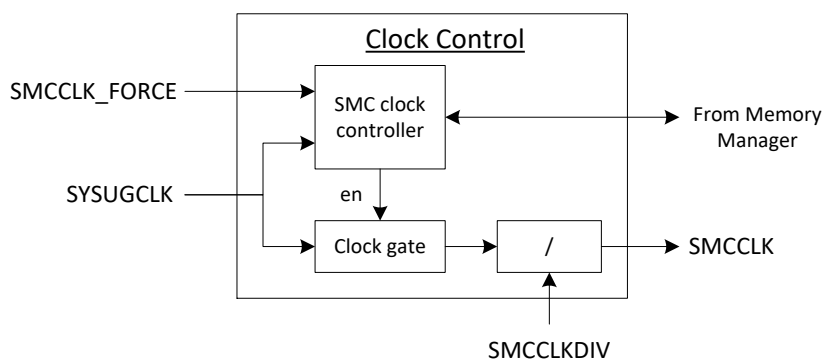
Контроллер SMC поддерживает подключение двух внешних микросхем памяти объемом до 32 Мбайт каждая. Доступ к внешней памяти осуществляется через АНВ-интерфейс памяти по следующим адресам:

- Банк памяти 0 (SMC\_NCS0):
  - 0x6000\_0000 – 0x61FF\_FFFF (Non-secure);
  - 0x7000\_0000 – 0x71FF\_FFFF (Secure).
- Банк памяти 1 (SMC\_NCS1):
  - 0x6200\_0000 – 0x63FF\_FFFF (Non-secure);
  - 0x7200\_0000 – 0x73FF\_FFFF (Secure).

## 19.2.3 Управление тактированием

Контроллер SMC имеет два тактовых домена:

- Тактирование АНВ-интерфейсов и блоков SMC Interface, Memory Manager, Format осуществляется тактовым сигналом SYSCLK. Для запроса включения тактирования SMC имеет low-power интерфейс, подключенный к системному контроллеру тактового сигнала SYSCLK.
- Тактирование интерфейса внешней памяти и блока Memory Interface выполняется с помощью сигнала SMCCLK, схема формирования которого показана на следующем рисунке.



**Рисунок 19.2. Схема формирования сигнала SMCCLK**

Сигнал SMCCLK формируется из тактового сигнала SYSUGCLK (free-running версия SYSCLK) путем деления. Коэффициент деления SMCCLKDIV задается в регистре SMC\_USER\_CONFIG и может принимать значения от 1 до 32.

После изменения значения SMCCLKDIV необходимо дождаться обновления текущего значения делителя SMCCLKDIV\_CUR в регистре SMC\_USER\_CONFIG.

Дополнительно, тактовый сигнал SMCCLK может быть автоматически отключен при отсутствии активных транзакций на интерфейсе памяти. Эта функция

управляется битом SMCCLK\_FORCE регистра CMU\_CLKFORCE (по умолчанию выключена).

## 19.2.4 Функционирование АНВ-интерфейса памяти

### 19.2.4.1 АНВ-пакеты фиксированной длины

Контроллер SMC предназначен для передачи пакетов данных в память/из памяти. Чем больше размер пакета, тем эффективнее выполняется передача данных и тем выше пропускная способность интерфейса.

Все АНВ-пакеты с фиксированной длиной (INCR4, INCR8, INCR16) используются контроллером без изменений.

### 19.2.4.2 АНВ-пакеты неопределенной длины

АНВ-пакеты типа INCR по умолчанию преобразуются контроллером в пакеты длиной 4 (INCR4).

При поступлении запроса чтения типа INCR контроллер SMC выполняет в 4 раза больше запросов чтения из памяти. При этом считанные данные помещаются в буфер FIFO. Если мастер-устройство продолжает чтение INCR-пакета, то данные могут быть отданы контроллером быстрее, т.к. они уже были запрошены из памяти. При завершении INCR-пакета лишние считанные данные сбрасываются контроллером.

При поступлении запроса записи типа INCR контроллер SMC формирует в 4 раза больше транзакций записи в память. Если мастер-устройство завершает INCR-пакет, то лишние посылки в память выполняются при неактивных сигналах SMC\_NBLS[1:0].

INCR-пакеты длиной более четырех посылок преобразуются в соответствующее количество INCR4-пакетов.

Преобразование INCR-пакетов в INCR4-пакеты может быть отключено с помощью установки битов CONV\_INCR\_DIS0 и CONV\_INCR\_DIS1 регистра SMC\_USER\_CONFIG для банков памяти 0 и 1 соответственно. В этом случае INCR-пакеты обрабатываются контроллером как одиночные пересылки (SINGLE).

### 19.2.4.3 Буферизация данных

Сигнал HPROT[2] интерфейса АНВ обозначает буферизацию данных при записи. Контроллер SMC поддерживает буферизируемые обращения. При этом подтверждение записи устанавливается контроллером сразу при получении данных во внутренний буфер и не дожидаясь фактической записи в память.

Если АНВ-запрос помечен как небуферизируемый, то подтверждение запроса установится только после завершения записи в память.

#### **19.2.4.4 Обработка ситуаций Read After Write**

При выполнении буферизируемой записи и последующего чтения внешней памяти по одному и тому же адресу может возникнуть конфликт типа Read After Write, когда запрос чтения опережает запрос записи. Для предотвращения этой ситуации контроллер SMC содержит логику для отслеживания адресов до 4-х запросов записи. Если запрос чтения попадает в область 4 Кбайт, которая в данный момент записывается, то этот запрос приостанавливается.

### **19.2.5 Форматирование запросов в память**

#### **19.2.5.1 Выравнивание адреса памяти**

Для формирования адреса памяти адрес запросов по АНВ-интерфейсу выравнивается по ширине памяти (16 бит).

#### **19.2.5.2 Выравнивание пакетной передачи**

Контроллер SMC предоставляет возможность форматирования передаваемых данных с учётом границ пакетов памяти. Для этого предназначено поле BURST\_ALIGN регистров SMC\_OPMODE\*.

При значении поля BURST\_ALIGN > 0 пакетные обращения к памяти выравниваются по заданной границе пакета. Если передаваемый пакет пересекает заданную границу памяти, то SMC разделяет передачу на несколько пакетов. Эта опция предназначена для использования с микросхемами памяти, имеющими внутренние страницы. Это может быть, например, асинхронная память со страничным доступом или синхронная псевдостатическая память (PSRAM).

При использовании данной опции следует убедиться, что размер страницы кратен заданной длине пакета, чтобы избежать ситуации, когда передаваемый пакет памяти пересекает границу страницы памяти.

При BURST\_ALIGN = 0 контроллер SMC игнорирует ограничение на размер пакета памяти. Это может быть использовано при работе с микросхемами памяти, не имеющими деления на страницы.

#### **19.2.5.3 Длина пакета памяти**

SMC позволяет программировать длину пакета памяти индивидуально для каждой микросхемы: фиксированная длина от 1 до 8 посылок по 16 бит или непрерывный пакет. Длина пакета задается полями RD\_BL и WR\_BL регистров SMC\_OPMODE\* для операций чтения и записи соответственно.

## 19.2.6 Энергосберегающий режим

Контроллер SMC имеет режим энергосбережения, при переходе в который выключается тактовый сигнал SMCCLK.

Переход в режим энергосбережения возможен по запросу через конфигурационный АНВ-интерфейс либо автоматически при отсутствии активности в контроллере (см. пункт 19.2.3).

SMC входит в состояние энергосбережения только при выполнении следующих условий:

- в блоке Format нет активных запросов;
- в блоке SMC Interface нет активных запросов;
- все буферы FIFO пусты;
- блок Memory Interface находится в неактивном состоянии.

После перехода в режим энергосбережения новые запросы к памяти не принимаются до перехода в рабочий режим.

При переходе в режим энергосбережения по запросу через конфигурационный АНВ-интерфейс выход из него выполняется также по запросу.

## 19.2.7 Регистры конфигурации микросхемы

Регистры SMC\_SET\_CYCLES и SMC\_SET\_OPMODE служат для хранения новых рабочих параметров интерфейса памяти.

Набор регистров SMC\_CYCLES\* и SMC\_OPMODE\* для каждой микросхемы содержит текущие параметры интерфейса памяти. Данные регистры доступны только для чтения и обновляются значениями из регистров SMC\_SET\_CYCLES и SMC\_SET\_OPMODE в следующих случаях:

- при выполнении команды UpdateReg;
- после выполнения команды ModeReg и завершения записи конфигурационных регистров внешней памяти.

SMC обеспечивает два механизма синхронизации параметров интерфейса с параметрами выбранной микросхемы памяти:

- Аппаратный механизм с помощью вывода SMC\_CRE. Применяется, например, в некоторых микросхемах PSRAM.

Для синхронной смены настроек необходимо:

- записать новые временные параметры и режим работы в регистры SMC\_SET\_CYCLES и SMC\_SET\_OPMODE;
- в регистре SMC\_DIRECT\_CMD выбрать нужную микросхему памяти, установить бит SET\_CRE, записать поле ADDR в соответствии с настройками микросхемы и записать тип команды в поле CMD\_TYPE – ModeReg + UpdateRegs;



- с момента записи регистра SMC\_DIRECT\_CMD контроллер SMC приостанавливает обработку обращений к памяти и новых конфигурационных команд;
  - после завершения записи регистра внешней памяти SMC обновляет регистры SMC\_CYCLES\* и SMC\_OPMODE\* для выбранного банка памяти и возобновляет обработку обращений к памяти и конфигурационных команд.
- Программный механизм. Применяется с микросхемами памяти, для записи регистров которых необходимо выполнить последовательность команд чтения и записи. При этом определенное записываемое значение на шине данных обозначает завершение конфигурации микросхемы. К таким микросхемам относится большинство микросхем типа NOR Flash.

Для синхронной смены настроек необходимо:

- записать новые временные параметры и режим работы в регистры SMC\_SET\_CYCLES и SMC\_SET\_OPMODE;
- в регистре SMC\_DIRECT\_CMD выбрать нужную микросхему памяти, записать в поле ADDR[15:0] значение команды, завершающей последовательность конфигурации микросхемы, и записать тип команды в поле CMD\_TYPE –UpdateRegs + AHB command;
- с момента записи регистра SMC\_DIRECT\_CMD контроллер SMC приостанавливает обработку новых конфигурационных команд и ожидает появления записи значения ADDR[15:0] на интерфейсе памяти;
- отправить необходимую последовательность команд чтения и записи через АHB-интерфейс памяти;
- после обнаружения значения ADDR[15:0] на шине данных SMC приостанавливает обработку обращений к памяти, обновляет регистры SMC\_CYCLES\* и SMC\_OPMODE\* для выбранного банка памяти и возобновляет обработку запросов к памяти.

## 19.2.8 Работа интерфейса памяти

Блок Memory Interface Интерфейс отправляет команды в память из соответствующего буфера FIFO и контролирует временные параметры этих команд. Новая команда отправляется только тогда, когда предыдущая команда завершилась и выдержана пауза в tTR тактов. Вдобавок, команда на чтение будет отправлена, только если в приемном буфере FIFO достаточно места для всех запрашиваемых данных.

### 19.2.8.1 Асинхронное чтение

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

Таблица 19.1. Настройка SMC\_OPMODE для асинхронного чтения

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	0	0	-	-	-	-	-

Таблица 19.2. Настройка SMC\_CYCLES для асинхронного чтения

Поле	tRC	tWC	tCEOЕ	tWP	tPC	tTR
Значение	3	-	1	-	-	-

На следующем рисунке показана одиночная транзакция асинхронного чтения с временем доступа, tRC, равным 3 тактам и задержкой разрешения выхода, tCEOЕ, 1 такт.

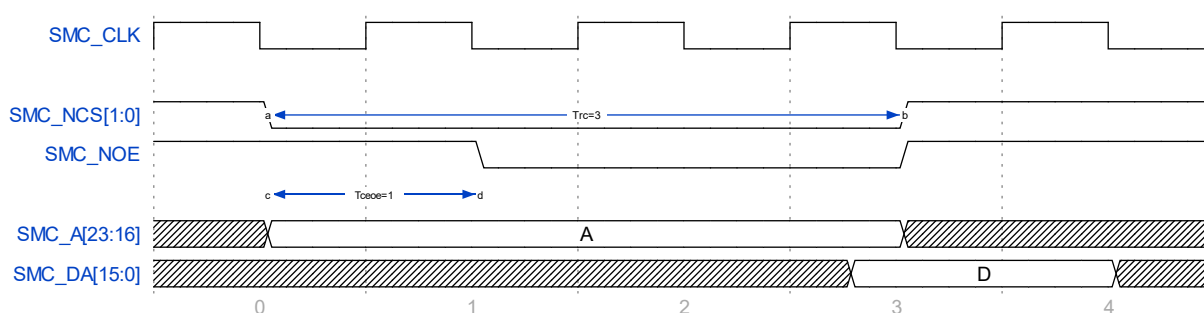


Рисунок 19.3. Временная диаграмма асинхронного чтения

#### 19.2.8.2 Асинхронное чтение в режиме мультиплексирования

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

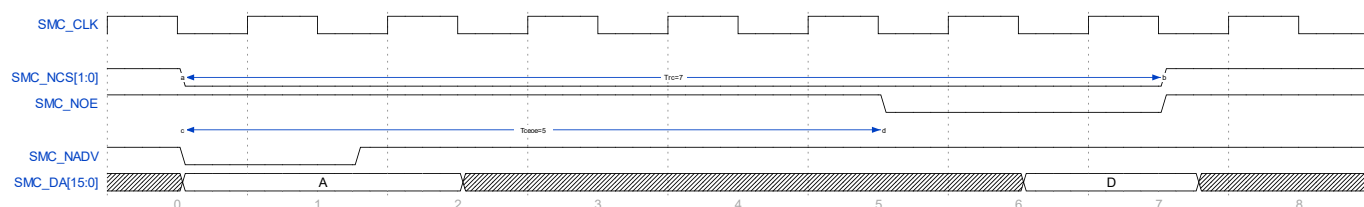
Таблица 19.3. Настройка SMC\_OPMODE для асинхронного чтения в режиме мультиплексирования

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	0	0	-	-	1	-	-

Таблица 19.4. Настройка SMC\_CYCLES для асинхронного чтения в режиме мультиплексирования

Поле	tRC	tWC	tCEOЕ	tWP	tPC	tTR
Значение	7	-	5	-	-	-

На следующем рисунке показана одиночная транзакция асинхронного чтения в режиме мультиплексирования с временем доступа,  $t_{RC}$ , равным 7 тактам и задержкой разрешения выхода,  $t_{CEOE}$ , 5 тактам.



**Рисунок 19.4. Временная диаграмма асинхронного чтения в режиме мультиплексирования**

### 19.2.8.3 Асинхронная запись

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

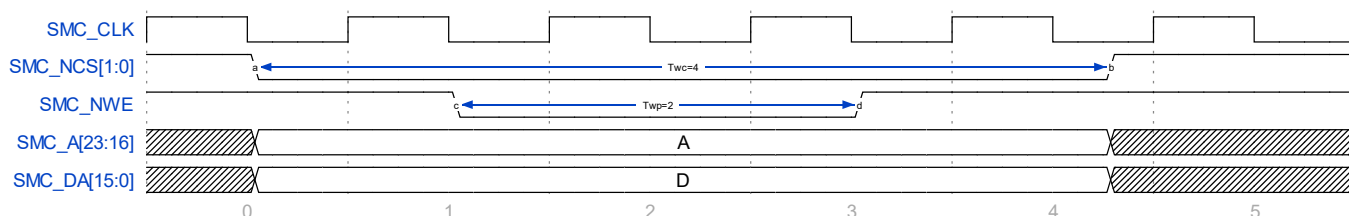
**Таблица 19.5. Настройка SMC\_OPMODE для асинхронной записи**

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	-	-	0	0	-	-	-

**Таблица 19.6. Настройка SMC\_CYCLES для асинхронной записи**

Поле	$t_{RC}$	$t_{WC}$	$t_{CEOE}$	$t_{WP}$	$t_{PC}$	$t_{TR}$
Значение	-	4	0	2	-	-

На следующем рисунке показана одиночная транзакция асинхронной записи с временем доступа,  $t_{WC}$ , равным 4 тактам и задержкой разрешения входа,  $t_{WP}$ , 2 тактам.



**Рисунок 19.5. Временная диаграмма асинхронной записи**

#### 19.2.8.4 Асинхронная запись в режиме мультиплексирования

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

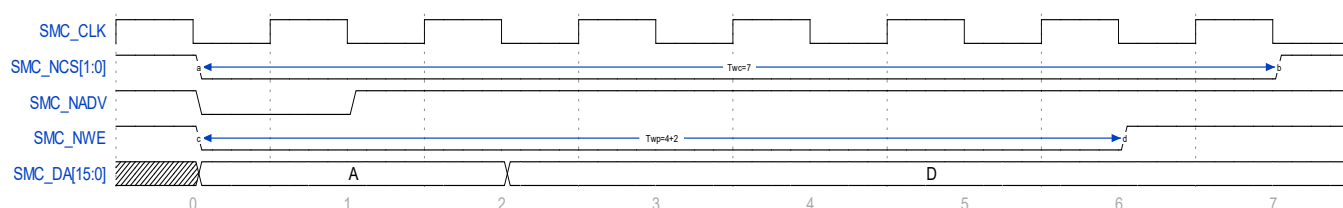
**Таблица 19.7. Настройка SMC\_OPMODE для асинхронной записи в режиме мультиплексирования**

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	-	-	0	0	0	-	-

**Таблица 19.8. Настройка SMC\_CYCLES для асинхронной записи в режиме мультиплексирования**

Поле	tRC	tWC	tCEOЕ	tWP	tPC	tTR
Значение	-	7	-	4	-	-

На следующем рисунке показана одиночная транзакция асинхронной записи с временем доступа, tWC, равным 7 тактам и задержкой разрешения входа, tWP, 4 тактам и 2 тактам для адресной фазы.



**Рисунок 19.6. Временная диаграмма асинхронной записи в режиме мультиплексирования**

#### 19.2.8.5 Синхронное пакетное чтение в режиме мультиплексирования

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

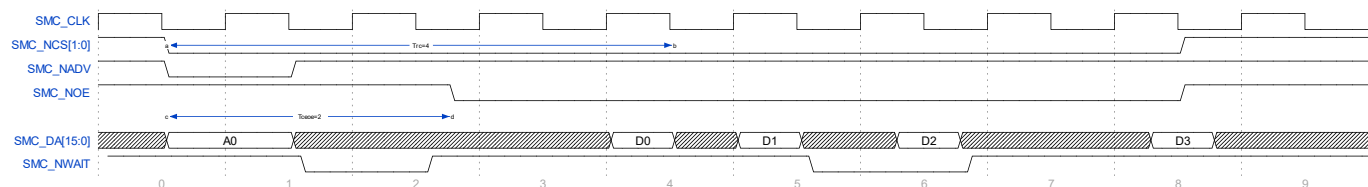
**Таблица 19.9. Настройка SMC\_OPMODE для синхронного пакетного чтения в режиме мультиплексирования**

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	1	<burst length>	-	-	-	-	-

**Таблица 19.10. Настройка SMC\_CYCLES для синхронного пакетного чтения в режиме мультиплексирования**

Поле	tRC	tWC	tCEOЕ	tWP	tPC	tTR
Значение	4	-	2	-	-	-

На следующем рисунке показано синхронное пакетное чтение в режиме мультиплексирования.



**Рисунок 19.7. Временная диаграмма синхронного пакетного чтения в режиме мультиплексирования**

#### 19.2.8.6 Синхронная пакетная запись в режиме мультиплексирования

В следующих таблицах показаны настройки регистров SMC\_OPMODE и SMC\_CYCLES.

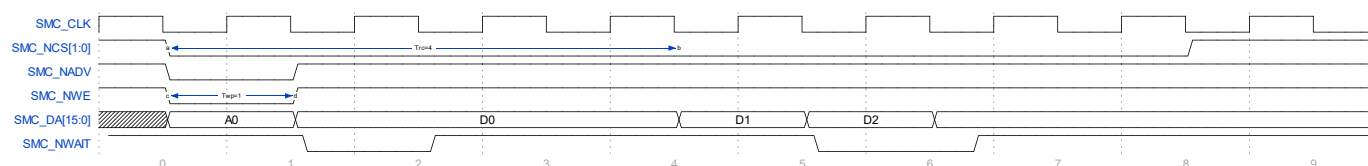
**Таблица 19.11. Настройка SMC\_OPMODE для синхронного пакетного чтения в режиме мультиплексирования**

Поле	RD_SYNC	RD_BL	WR_SYNC	WR_BL	ADV	BLS	BURST_ALIGN
Значение	-	-	1	<burst length>	-	-	-

**Таблица 19.12. Настройка SMC\_CYCLES для синхронного пакетного чтения в режиме мультиплексирования**

Поле	tRC	tWC	tCEOE	tWP	tPC	tTR
Значение	-	4	-	1	-	-

На следующем рисунке показано синхронная пакетная запись в режиме мультиплексирования.



**Рисунок 19.8. Временная диаграмма синхронной пакетной записи в режиме мультиплексирования**

## 19.2.9 Подключение LCD дисплеев

Контроллер SMC поддерживает подключение LCD дисплеев с параллельным интерфейсом типа Intel 8080 и Motorola 6800.

Интерфейс таких дисплеев состоит из шины данных D[15:0] (разрядность может быть меньше) и управляющих сигналов  $\overline{CS}$ , RS,  $\overline{RD}$ ,  $\overline{RW}$ , E, R/  $\overline{W}$ , показанных в следующей таблице.

**Таблица 19.13. Управляющие сигналы дисплеев**

RS	Intel 8080		Motorola 6800		Функция
	$\overline{RD}$	$\overline{RW}$	E	R/ $\overline{W}$	
0	0	1	1	1	Чтение регистров
0	1	0	1	0	Запись регистров
1	0	1	1	1	Чтение памяти
1	1	0	1	0	Запись памяти

Для подключения дисплея к интерфейсу SMC используются выходы SMC\_NCS, SMC\_DA, SMC\_NOE, SMC\_NWE и один из выводов SMC\_A[23:16]. Дополнительно может потребоваться один GPIO вывод или внешний инвертор. Один из выводов адреса SMC\_A используется для выбора обращения к регистрам дисплея или к его памяти. Подключение показано в следующих таблицах.

**Таблица 19.14. Подключение сигналов дисплея типа Intel 8080**

Вывод дисплея	Подключение
$\overline{CS}$	Один из выводов SMC_NCS[1:0]
$\overline{RD}$	SMC_NOE
$\overline{RW}$	SMC_NWE
RS	Один из выводов SMC_A[23:16]
D[15:0]	SMC_DA[15:0]

**Таблица 19.15. Подключение сигналов дисплея типа Motorola 6800**

Вывод дисплея	Подключение
$\overline{CS}$	Один из выводов SMC_NCS[1:0]
E	Вывод GPIO либо инвертированный сигнал SMC_NCS
R/ $\overline{W}$	SMC_NWE
RS	Один из выводов SMC_A[23:16]
D[15:0]	SMC_DA[15:0]

Для работы с дисплеями функция мультиплексирования адреса и данных должна быть отключена в контроллере (бит ADV = 0 в регистре SMC\_OPMODEx).

## 19.3 Регистры

### 19.3.1 Перечень регистров SMC

Таблица 19.16. Регистры SMC

Условное Обозначение	Название регистра	Исходное состояние	Смещение
-	Резерв	0x0	0x0000 – 0x0FFF
SMC_MEMC_STATUS	Регистр статуса контроллера памяти	0x0	0x1000
SMC_MEMIF_CFG	Регистр конфигурации интерфейса памяти	0x15	0x1004
SMC_MEMC_CFG_SET	Регистр установки конфигурации контроллера памяти	0x0	0x1008
SMC_MEMC_CFG_CLR	Регистр сброса конфигурации контроллера памяти	0x0	0x100C
SMC_DIRECT_CMD	Командный регистр	0x0	0x1010
SMC_SET_CYCLES	Регистр установки временных параметров интерфейса	0x0	0x1014
SMC_SET_OPMODE	Регистр установки режима работы интерфейса	0x0	0x1018
SMC_REFRESH_PERIOD		0x0	0x1020
SMC_CYCLES0	Регистр настройки временных параметров для банка памяти 0	0x0002B3CC	0x1100
SMC_OPMODE0	Регистр настройки режима работы для банка памяти 0	0x02000801	0x1104
SMC_CYCLES1	Регистр настройки временных параметров для банка памяти 1	0x0002B3CC	0x1120
SMC_OPMODE1	Регистр настройки режима работы для банка памяти 1	0x22000801	0x1124
SMC_USER_CONFIG	Регистр дополнительной конфигурации контроллера	0x0	0x1204

### 19.3.2 Регистр SMC\_MEMC\_STATUS

Таблица 19.17. Регистр SMC\_MEMC\_STATUS

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	STATE	RO	0	Состояние контроллера памяти: 0x0: контроллер готов к работе; 0x1: режим энергосбережения.

### 19.3.3 Регистр SMC\_MEMIF\_CFG

Таблица 19.18. Регистр SMC\_MEMIF\_CFG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17:16	EXCL_MON	RO	0	Количество реализованных мониторов эксклюзивного доступа – 0.
15:7	-	RO	0	Резерв
6	REMAP	RO	0	Функция REMAP не используется

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
5:4	MEM_WIDTH	RO	0x1	Максимальная разрядность интерфейса памяти – 16 бит
3:2	MEM_CHIPS	RO	0x1	Количество поддерживаемых микросхем памяти – 2
1:0	MEM_TYPE	RO	0x1	Тип памяти – SRAM

### 19.3.4 Регистр SMC\_MEMC\_CFG\_SET

Таблица 19.19. Регистр SMC\_MEMC\_CFG\_SET

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0	Резерв
2	LPREQ	WO	0	Запись 1 инициирует запрос на переход контроллера в энергосберегающее состояние. Запись 0 игнорируется.
1:0	-	RO	0	Резерв

### 19.3.5 Регистр SMC\_MEMC\_CFG\_CLR

Таблица 19.20. Регистр SMC\_MEMC\_CFG\_CLR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0	Резерв
2	LPEXIT	WO	0	Запись 1 инициирует запрос на выход контроллера из энергосберегающего состояния. Запись 0 игнорируется.
1:0	-	RO	0	Резерв

### 19.3.6 Регистр SMC\_DIRECT\_CMD

Регистр SMC\_DIRECT\_CMD предназначен для отправки конфигурационных команд во внешнюю память и для управления обновлением конфигурационных регистров контроллера значениями из регистров SMC\_SET\_OPMODE и SMC\_SET\_CYCLES.

Таблица 19.21. Регистр SMC\_DIRECT\_CMD

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:26	-	RO	0	Резерв
25:23	CHIP_SELECT	WO	0	Выбор банка памяти для обновления конфигурационных регистров контроллера, а также для доступа к регистрам памяти: 0x0: банк 0; 0x1: банк 1; 0x2 – 0x7: резерв
22:21	CMD_TYPE	WO	0	Определяет тип конфигурационной команды: 0x0: UpdateRegs + AHB command; 0x1: ModeReg 0x2: UpdateRegs; 0x3: ModeReg + UpdateRegs



Номер бита	Название	Тип доступа	Исходное состояние	Назначение
20	SET_CRE	WO	0	При выполнении команды ModeReg задает значение выхода SMC_CRE: 0x0: выход SMC_CRE сброшен; 0x1: выход SMC_CRE установлен
19:0	ADDR	WO	0	При выполнении команды ModeReg поле используется в качестве разрядов [19:0] адреса внешней памяти. При выполнении команды UpdateRegs+AHB command поле ADDR[15:0] используется для сопоставления со значением на шине SMC_DA[15:0] для синхронного обновления конфигурации.

### 19.3.7 Регистр SMC\_SET\_CYCLES

Регистр предназначен для хранения новой конфигурации временных параметров интерфейса. Значение этого регистра переписывается в регистр SMC\_CYCLES\* выбранного банка при выполнении команды UpdateRegs.

**Таблица 19.22. Регистр SMC\_SET\_CYCLES**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:20	-	RO	0	Резерв
19:17	tTR	WO	0	Задержка между последовательными пакетами (turnaround)
16:14	tPC	WO	0	Длительность цикла доступа к странице
13:11	tWP	WO	0	Задержка активации вывода CMS_NWE
10:8	tCEOE	WO	0	Задержка активации вывода SMC_NOE
7:4	tWC	WO	0	Длительность цикла записи
3:0	tRC	WO	0	Длительность цикла чтения

### 19.3.8 Регистр SMC\_SET\_OPMODE

Регистр предназначен для хранения новой конфигурации режима работы интерфейса. Значение этого регистра переписывается в регистр SMC\_OPMODE\* выбранного банка при выполнении команды UpdateRegs.

**Таблица 19.23. Регистр SMC\_SET\_OPMODE**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0	Резерв
15:13	SET_BURST_ALIGN	WO		Задаёт границу пакета памяти, при пересечении которой пакет разбивается на несколько: 0x0: нет границы; 0x1: 32 слова (по 16 бит); 0x2: 64 слова; 0x3: 128 слов; 0x4: 256 слов; 0x5 – 0x7: резерв.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
12	SET_BLS	WO		Бит задает поведение выводов SMC_NBLS[1:0]: 0x0: выходы SMC_NBLS переключаются так же, как выходы SMC_NCS; 0x1: выходы SMC_NBLS переключаются так же, как вывод SMC_NWE.
11	SET_ADV	WO		Задаёт использование сигнала SMC_NADV: 0x0: сигнал не используется; 0x1: сигнал используется, шины адреса и данных мультиплексируются. Бит SET_ADV должен быть установлен при работе с микросхемами памяти. При работе с LCD дисплеем бит должен быть сброшен.
10	SET_BAA	WO	0	Не используется, резерв.
9:7	SET_WR_BL	WO	0	Задаёт длину пакета данных при записи: 0x0: 1 слово (16 бит); 0x1: 4 слова; 0x2: 8 слов; 0x3, 0x4: резерв; 0x5: непрерывный пакет; 0x6, 0x7: резерв
6	SET_WR_SYNC	WO	0	Задаёт тип интерфейса при записи: 0x0: асинхронный; 0x1: синхронный
5:3	SET_RD_BL	WO	0	Задаёт длину пакета данных при чтении: 0x0: 1 слово (16 бит); 0x1: 4 слова; 0x2: 8 слов; 0x3, 0x4: резерв; 0x5: непрерывный пакет; 0x6, 0x7: резерв
2	SET_RD_SYNC	WO	0	Задаёт тип интерфейса при чтении: 0x0: асинхронный; 0x1: синхронный
1:0	SET_MW	WO	0	Задаёт разрядность интерфейса памяти: 0x0: резерв; 0x1: 16 бит; 0x2, 0x3: резерв

### 19.3.9 Регистр SMC\_REFRESH\_PERIOD

Регистр позволяет настроить паузу между обращениями, необходимую для обновления микросхем памяти типа PSRAM.

Таблица 19.24. Регистр SMC\_REFRESH\_PERIOD

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:4	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
3:0	PERIOD	RW	0	<p>Задаёт количество последовательных пакетов перед деактивацией микросхемы памяти для возможности ее обновления:</p> <p>0x0: деактивация не выполняется;</p> <p>0x1: деактивация после каждого пакета;</p> <p>0x2: деактивация после 2-х последовательных пакетов;</p> <p>0x3: деактивация после 3-х последовательных пакетов;</p> <p>...</p> <p>0xF: деактивация после 15-ти последовательных пакетов.</p>

### 19.3.10 Регистр SMC\_CYCLES<0-1>

Регистр хранит текущие временные параметры для соответствующего банка памяти.

Регистр доступен только для чтения.

**Таблица 19.25. Регистр SMC\_CYCLES<0-1>**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:20	-	RO	0	Резерв
19:17	tTR	RO	0x1	Задержка между последовательными пакетами (turnaround)
16:14	tPC	RO	0x2	Длительность цикла доступа к странице
13:11	tWP	RO	0x6	Задержка активации вывода CMS NWE
10:8	tCEOE	RO	0x3	Задержка активации вывода SMC NOE
7:4	tWC	RO	0xC	Длительность цикла записи
3:0	tRC	RO	0xC	Длительность цикла чтения

### 19.3.11 Регистр SMC\_OPMODE<0-1>

Регистр хранит текущий режим работы для соответствующего банка памяти.

Регистр доступен только для чтения.

**Таблица 19.26. Регистр SMC\_OPMODE<0-1>**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:14	ADDRESS_MATCH	RO	*	<p>Сравниваемое значение для разрядов [31:24] адреса на АНВ-интерфейсе для определения номера банка памяти:</p> <p>банк 0: 0x00;</p> <p>банк 1: 0x02.</p>
23:16	ADDRESS_MASK	RO	0x2	Маска для разрядов [31:24] адреса на АНВ-интерфейсе для определения номера банка памяти

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
15:13	BURST_ALIGN	RO		Граница пакета памяти, при пересечении которой пакет разбивается на несколько: 0x0: нет границы; 0x1: 32 слова (по 16 бит); 0x2: 64 слова; 0x3: 128 слов; 0x4: 256 слов; 0x5 – 0x7: резерв.
12	BLS	RO		Поведение выводов SMC_NBLS[1:0]: 0x0: выводы SMC_NBLS переключаются так же, как выводы SMC_NCS; 0x1: выводы SMC_NBLS переключаются так же, как вывод SMC_NWE.
11	ADV	RO	1	Использование сигнала SMC_NADV: 0x0: сигнал не используется; 0x1: сигнал используется, шины адреса и данных мультиплексируются. Бит SET_ADV должен быть установлен при работе с микросхемами памяти. При работе с LCD дисплеем бит должен быть сброшен.
10	BAA	RO	0	Не используется, резерв.
9:7	WR_BL	RO	0	Длина пакета данных при записи: 0x0: 1 слово (16 бит); 0x1: 4 слова; 0x2: 8 слов; 0x3, 0x4: резерв; 0x5: непрерывный пакет; 0x6, 0x7: резерв
6	WR_SYNC	RO	0	Тип интерфейса при записи: 0x0: асинхронный; 0x1: синхронный
5:3	RD_BL	RO	0	Длина пакета данных при чтении: 0x0: 1 слово (16 бит); 0x1: 4 слова; 0x2: 8 слов; 0x3, 0x4: резерв; 0x5: непрерывный пакет; 0x6, 0x7: резерв
2	RD_SYNC	RO	0	Тип интерфейса при чтении: 0x0: асинхронный; 0x1: синхронный
1:0	MW	RO	0x1	Разрядность интерфейса памяти: 0x0: резерв; 0x1: 16 бит; 0x2, 0x3: резерв

### 19.3.12 Регистр SMC\_USER\_CONFIG

Таблица 19.27. Регистр SMC\_USER\_CONFIG

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:21	-	RO	0	Резерв
20:16	SMCCLKDIV_CUR	RO	0	Текущее значение поля SMCCLKDIV регистра SMC_USER_CONFIG.
15:8	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
7	CONV_INCR_DIS1	RW	0	Управление преобразованием АНВ-пакетов типа INCR в пакеты типа INCR4 для банка памяти 1: 0x0: пакеты типа INCR преобразуются в пакеты типа INCR4; 0x1: пакеты типа INCR обрабатываются как одиночные обращения типа SINGLE
6	CONV_INCR_DIS0	RW	0	Управление преобразованием АНВ-пакетов типа INCR в пакеты типа INCR4 для банка памяти 0: 0x0: пакеты типа INCR преобразуются в пакеты типа INCR4; 0x1: пакеты типа INCR обрабатываются как одиночные обращения типа SINGLE
5	NADV_LEVEL	RW	0	Управление активным уровнем сигнала SMC_NADV: 0x0: активный уровень 0; 0x1: активный уровень 1.
4:0	SMCCLKDIV	RW	0	Коэффициент деления для формирования тактового сигнала SMCCLK. В поле записывается значение минус 1.

## 20. GNSS: контроллер цифровой части приемника навигационного сигнала

В микросхеме реализован контроллер цифровой части приемника навигационного сигнала позволяющий обеспечить следующий функционал и характеристики:

- Прием навигационного сигнала ГЛОНАСС в полосах L1 и L2;
- Прием навигационного сигнала GPS в полосах L1 и L2;
- IQ интерфейс с внешней RF и антенной частью приемника;

Контроллер позволяет получать навигационное решение вместе с прошивкой навигационного ПО, поставляемого компанией ЭЛВИС.

### 20.1 Navicore S: введение

Приемник поддерживает сигналы следующих GNSS.

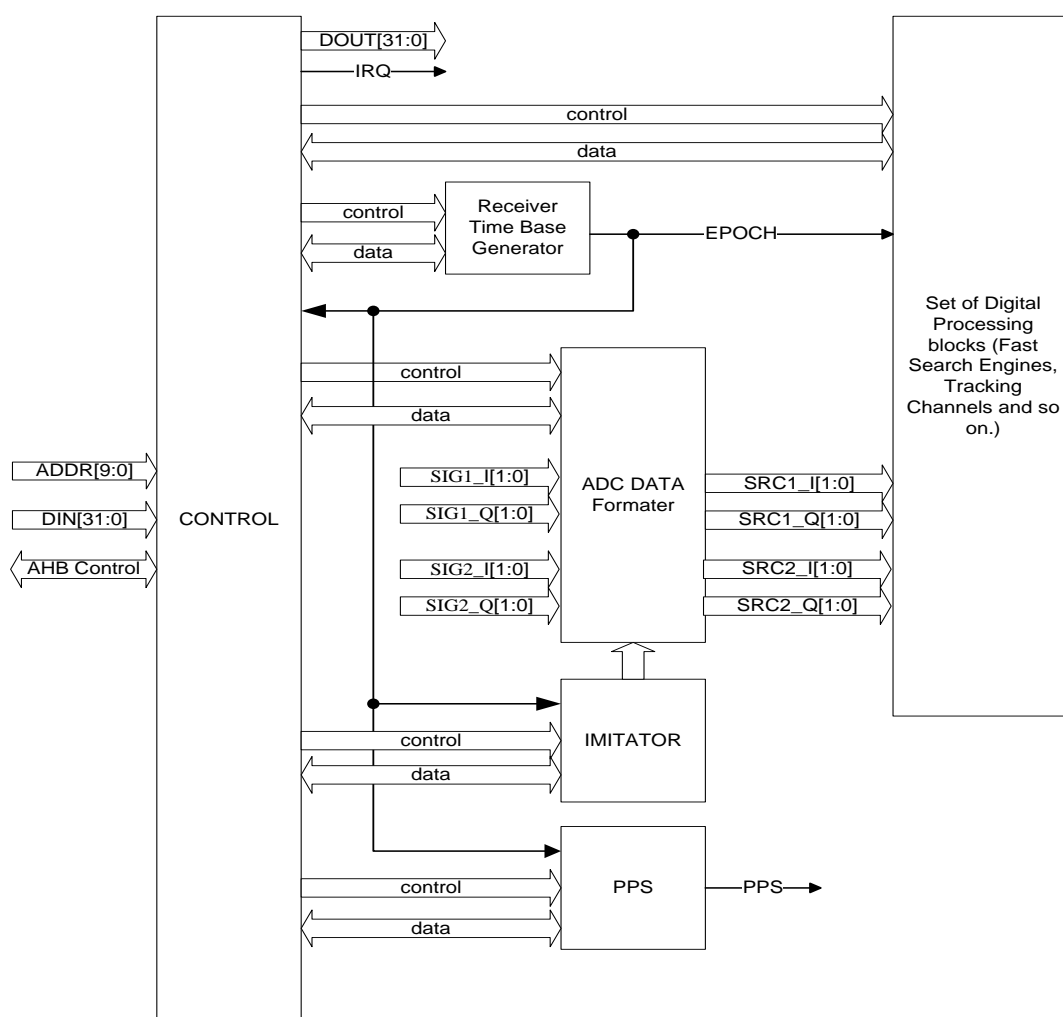
Открытые коды GPS C/A в полосах L1 (1575.42) и L2 (1227.6).

Открытые коды GLONASS C/A в полосах L1( $1602+0.5625*k$ ) и L2 ( $1246+0.4375*k$ ).

Navicore S содержит следующие компоненты:

- 20 следящих каналов (CA5)
- 1 устройство быстрого поиска (FSE)
- Коллектор
- Генератор временных интервалов
- Модуль PPS
- Интерфейс ADC
- Имитатор - симулятор сигнала спутника

Блок-схема компонентов Navicore S, подключенных к SoC хоста, показана на рисунке ниже.



**Рисунок 20.1. Блок-схема Navicore S**

Основные компоненты Navicore S даны в подразделах главы “Navicore S: введение” ниже с подробным описанием блоков.

### 20.1.1 Следящие каналы CA5

Двадцать корреляционных следящих каналов повышенной точности предназначены для слежения за сигналами GNSS при помощи 5 подканалов с независимо конфигурируемыми интервалами задержки. Входные данные представляют собой 2 квадратурных компонента 2 бита каждый. Продолжительность накопления устанавливается программно от 1 до 16 мс. Корреляционные отклики, вычисленные в подканалах CA5, обеспечивают следящее ПО информацией по задержке и частоте.

### 20.1.2 FSE (устройство быстрого поиска)

Устройство FSE предназначено для ускорения поиска сигналов GNSS. Поиск основывается на соответствующих принципах фильтрации. FSE каждый раз конфигурируется и запускается программным обеспечением; проверяет одну гипотезу Доплеровской частоты в одном данном PN-коде. Выход представляет собой наиболее вероятную задержку сигнала. Кроме того, все результаты некогерентного накопления доступны через АНВ.

### 20.1.3 Генератор временных интервалов

Таймер обеспечивает локальный опорный 1 мс сигнал. Отсчет выводится из частоты генератора RFFE, которая делится на программируемый делитель. Это основной отсчет синхронизации (локальный сигнал EPOCH) для всех задач обработки.

### 20.1.4 Модуль PPS

Высокоточный импульс в секунду точности следящий генератор секундного импульса. PPS выводится из локального сигнала EPOCH (1 мс). Слежение выполняется программным обеспечением в соответствии с фактическим координатно-временным решением с целью достигнуть глобальной синхронизации PPS.

### 20.1.5 Интерфейс ADC

Модуль интерфейса ADC предназначен для создания условий для входного потока данных и преобразования формата данных. Дополнительной задачей ставится оценка параметров сигнала (например, амплитуды сигнала) для реализации алгоритма AGC.

### 20.1.6 IMITATOR (симулятор сигнала спутника)

Отладочный симулятор сигнала спутников GNSS предназначен для проверки основной функциональности Navicore. Симулятор позволяет установить спутниковый PN-код, значение мощности шума, который будет смешан с сигналом, а также Доплеровский сдвиг сигнала.

## 20.2 Назначение регистров Navicore S

Некоторые регистры определены для конфигурации и управления модулями Navicore S. Регистры выделены для Шины АНВ. В таблице ниже указаны диапазоны адресов для различных модулей.

**Таблица 20.1. Диапазоны регистров Navicore S**

Смещение	Название блока
0x0000 – 0x00FF	Регистры общего управления Navicore
0x0100 – 0x01FF	Регистры устройства быстрого поиска (FSE)



0x0200 – 0x0340	Регистры управления следящими каналами (CA5)
-----------------	--

Общая карта регистров Navicore S приведена в Таблице ниже.

**Таблица 20.2. Карта регистров Navicore S**

Смещение	Обозначение	Доступ	Значение сброса	Описание
<b>Регистры общего контроля и состояния Navicore S: 0x0000–0x00FC</b>				
0x0000-0x0004	<b>Тестовые регистры версии Navicore S</b>			
0x0000	VERSION	R	0x00000053	Версия Navicore
0x0004	TEST_RW	RW	0x00001234	Для испытаний доступа к записи/чтению
0x0008-0x000C	<b>Установка генератора локальной эпохи</b>			
0x0008	TB_ECNT	RW	0x00000000	Счетчик EPOCH
0x000C	TB_MS_CODE	RW	0x00003FF0	Количество отсчетов ADC в 1 мс
0x0010-0x0014	<b>Установка генератора локальной частоты микросхемы</b>			
0x0010	FCHIP_CA5_GPS	RW	0x20000000	Коэффициент чиповой скорости GPS
0x0014	FCHIP_CA5_GLO	RW	0x20000000	Коэффициент чиповой скорости GLONASS
0x0018-0x001C	<b>Управление генератора PPS</b>			
0x0018	PPS_MS	RW	0x00013FF0	Общая синхронизация генератора PPS
0x001C	PPS_OFS	RW	0x00000000	Начальное состояние генератора PPS
0x0020-0x0030	<b>Регистры управления и состояния интерфейса ADC</b>			
0x0020	ADC_SIG1_I	R	0x00000000	Статистика состояний SIG1_I
0x0024	ADC_SIG1_Q	R	0x00000000	Статистика состояний SIG1_Q
0x0028	ADC_SIG2_I	R	0x00000000	Статистика состояний SIG2_I
0x002C	ADC_SIG2_Q	R	0x00000000	Статистика состояний SIG2_Q
0x0030	ADC_CTRL	RW	0x00000053	Представление входных данных и режим преобразования
0x0034-0x0044	<b>IMIT – Регистры управления сигналом локального имитатора</b>			
0x0034	IMIT_PCHIP	RW	0x00000000	Коэффициент фазы чипа Имитатора
0x0038	IMIT_FDOP	RW	0x00000000	Коэффициент частоты гетеродина Имитатора
0x003C	IMIT_G1G2	RW	0x00000000	Начальное состояние PN-генераторов Имитатора
0x0040	IMIT_CFG	RW	0x00000018	Конфигурация Имитатора
0x0044	IMIT_NPF	RW	0x00000000	Коэффициент мощности шума Имитатора
0x0048-0x004C	N/A			Зарезервировано
0x0050-0x0058	<b>ISG – Генератор начального состояния для GPS/GLONASS</b>			
0x0050	ISG_CHIP	RW,R	0x80000000	Задержка чипа G1/G2/G
0x0054	ISG_GPS	R	0x00000000	Состояние G1/G2 для GPS
0x0058	ISG_GLO	R	0x00000000	Состояние G для GLONASS
0x005C	N/A			Зарезервировано
0x0060-0x0068	<b>IRQ – Регистры управления прерываниями</b>			
0x0060	IRQ_CFG	RW	0x00000000	Регистр конфигурации IRQ
0x0064	IRQ_STAT	RW	0x00000000	Регистр состояния IRQ
0x0068	IRQ_ACK	W	0x00000000	Конфигурация IRQ

Смещение	Обозначение	Доступ	Значение сброса	Описание
0x006C	N/A			Зарезервировано
0x0070-0x0074	<b>COL – Регистры управления устройства сбора данных CA5</b>			
0x0070	COL_CFG	RW,W,R	0x00000004	Конфигурация устройства сбора данных
0x0074	COL_DATA	R	0x00000000	Регистр данных устройства сбора данных
0x0078-0x00FC	N/A			Зарезервировано
<b>Адреса регистров устройства быстрого поиска (FSE): 0x0100-0x01FC</b>				
0x0100	FSE_FHET	RW	0x00000000	Коэффициент частоты гетеродина
0x0104	FSE_CFG	RW	0x850000FF	Режим FSE
0x0108	FSE_ACC	RW	0x00000000	Режим аккумулятора FSE
0x010C	FSE_PR SIN	WO	0x00000000	Регистр загрузки PRS
0x0110	FSE_MAX	R	0x00000000	Результат поиска
0x0114	FSE_NCA_RES	R	0x00000000	Результаты некогерентного накопления
0x0118	FSE_FCHIP	RW	0x20000000	Коэффициент чиповой скорости SE
0x011C-0x01FC	N/A			Зарезервировано
<b>CA5 tracking channels control registers address range: 0x0200-0x02FC</b>				
0x0200-0x020C	<b>CA5_0 control registers</b>			
0x0200	CA5_0_PCHIP	RW	0x00000000	Chip initial phase
0x0204	CA5_0_FHET	RW	0x00000000	Коэффициент частоты гетеродина
0x0208	CA5_0_CFG1	RW	0x00000000	Регистр установки канала 1
0x020C	CA5_0_CFG2	RW	0x00000000	Регистр установки канала 2
0x0210-0x021C	<b>Регистры управления CA5_1</b>			
0x0220-0x022C	<b>Регистры управления CA5_2</b>			
0x0230-0x023C	<b>Регистры управления CA5_3</b>			
0x0240-0x024C	<b>Регистры управления CA5_4</b>			
0x0250-0x025C	<b>Регистры управления CA5_5</b>			
0x0260-0x026C	<b>Регистры управления CA5_6</b>			
0x0270-0x027C	<b>Регистры управления CA5_7</b>			
0x0280-0x028C	<b>Регистры управления CA5_8</b>			
0x0290-0x029C	<b>Регистры управления CA5_9</b>			
0x02A0-0x02AC	<b>Регистры управления CA5_10</b>			
0x02B0-0x02BC	<b>Регистры управления CA5_11</b>			
0x02C0-0x02CC	<b>Регистры управления CA5_12</b>			
0x02D0-0x02DC	<b>Регистры управления CA5_13</b>			
0x02E0-0x02EC	<b>Регистры управления CA5_14</b>			
0x02F0-0x02FC	<b>Регистры управления CA5_15</b>			
0x0300-0x030C	<b>Регистры управления CA5_16</b>			
0x0310-0x031C	<b>Регистры управления CA5_17</b>			
0x0320-0x032C	<b>Регистры управления CA5_18</b>			
0x0330-0x033C	<b>Регистры управления CA5_19</b>			

## 20.3 Следящий канал CA5

### 20.3.1 Общее описание следящего канала

Следящий канал CA5 отличается увеличенной точностью отслеживания спутникового сигнала. Канал состоит из пяти подканалов с корректируемым сдвигом времени поступления входного сигнала от 0 до 15 циклов частоты дискретизации. Канал компенсирует Доплеровскую частоту IF сигнала, и затем вычисляет свою свертку с опорным PN-кодом и накапливает результаты в течение предопределенного периода времени (с 1 до 16 эпох).

Операции, выполняемые в канале, описываются формулой:

$$Y_n(\tau) = \int_0^T S(t) c(t - \tau) e^{-j\omega t} dt$$

где:

$n = 1 \dots 5$  - номер подканала,

$t$  – задержка опорного PN-кода в отсчетах,

$T$  – время накопления.

Формулу выше можно переписать в дискретной форме:

$$ACC_d = \sum_{i=1}^N (S_i \cdot H_i) \cdot Cp_{i+d},$$

где:

$ACC_d$  - накопление в подканале с задержкой  $d$ ,

$N$  - количество отсчетов в эпохе,

$S_i$  - комплексные (квадратурные отсчеты) входного сигнала,

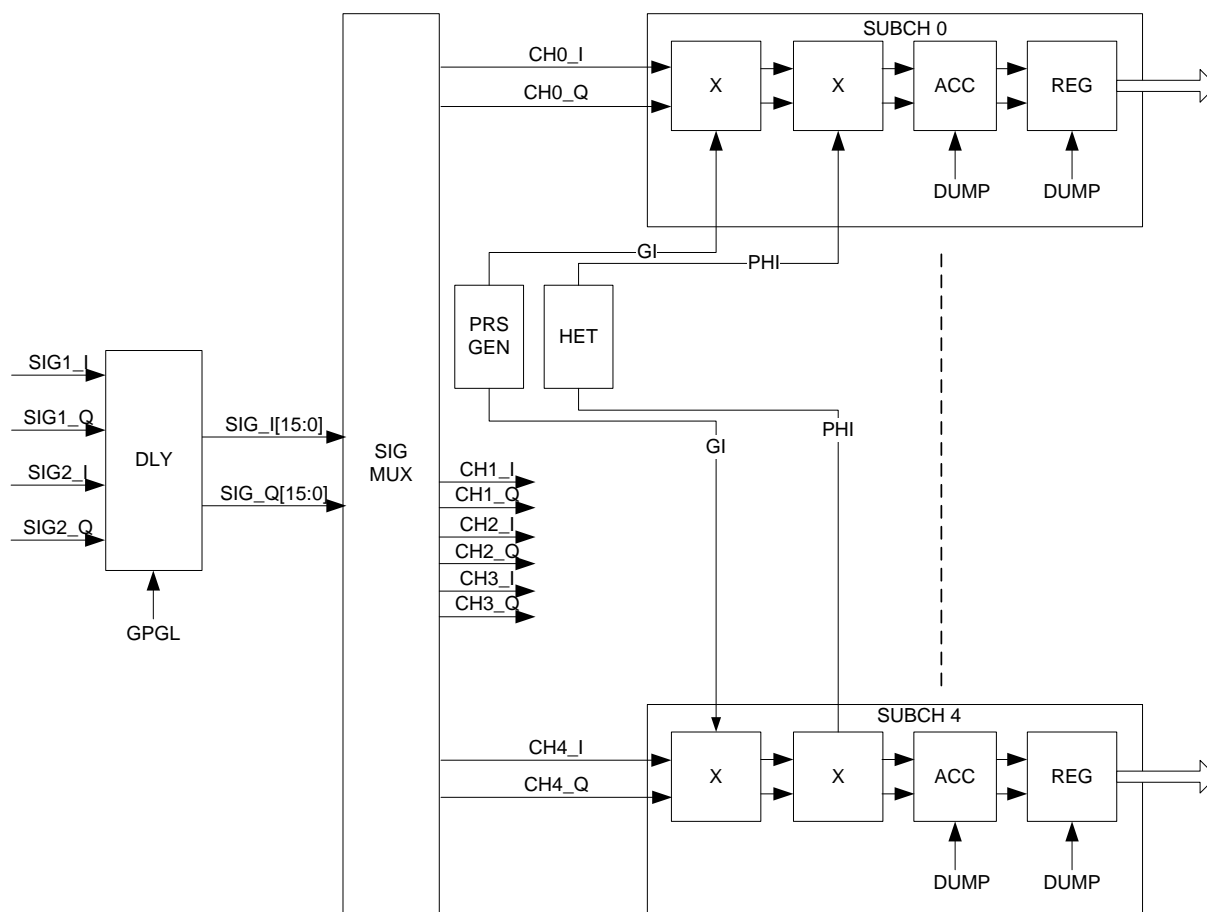
$H_i$  - комплексный сигнал гетеродина,

$Cp_{i+d}$  - задержанный PN-код.

В канале предусмотрены два режима: одиночный и непрерывный. В одиночном режиме канал находится в состоянии ожидания и начинает работать с предварительно установленным числом ЕРОСН. Достигнув заранее заданного количества когерентных накоплений, канал устанавливает флаг готовности и входит в режим ожидания. В непрерывном режиме канал постоянно вычисляет когерентное накопление и устанавливает флаг готовности после каждого накопления. В этом режиме команда старт используется только для пересинхронизации канала в заданное время и для применения новых настроек канала.

Регистры управления канала, а также стартовый сигнал канала, являются конвейерными. Если канал вычисляет когерентное накопление, а также когда в контрольные регистры записываются другие значения, то изменения будут применены только после завершения накопления. В состоянии ожидания (idle) изменения применяются сразу.

Блок-схема следящего канала CA5 показана ниже:



**Рисунок 20.2. Блок-схема канала CA5**

Тип входного сигнала (GPS или GLONASS) выбирается в поле CA5\_CFG1\_GPGL регистра CA5\_CFG1. Чиповая частота определяется регистрами FCHIP\_CA5\_GPS и FCHIP\_CA5\_GLONASS, которые определяют значения чиповой частоты (см. 10.3).

Кроме установки чиповой частоты также можно установить начальное значение фазового аккумулятора, используемое для генерации чиповой скорости, путем определения желательного числа регистра CA5\_PCHIP. Значение 0x7FFFFFFF соответствует максимальному сдвигу одного чипа минус один отсчет CLK, значение ноль соответствует нулевому смещению. 31-й бит регистра игнорируется. Это значение будет применено после завершения текущего накопления.

Частота гетеродина устанавливается в регистре CA5\_FHET.

Фаза гетеродина сбрасывается в регистр накопления фазы местного гетеродина каждый раз, когда запускается новое накопление.

Настройки канала устанавливаются в регистрах CA5\_CFG1 и CA5\_CFG2.

Задержка PN-кода регулируется указанием начального состояния генераторов кода. Начальные значения для регистров G1 и G2 генератора определяются в полях CA5\_CFG1\_G1 и CA5\_CFG1\_G2 регистра CA5\_CFG1. Эти начальные значения загружаются в конце текущего периода накопления. В режиме GPS биты [9: 0] и [25:16] должны использоваться для битов сдвиговых регистров G1 и G2, соответственно. В режиме GLONASS используется только бит G1 [8:0].

Аккумуляторы имеют ширину 23 бит. 16-битный диапазон аккумулятора картируется (отображается) на выходе канала. Выходной диапазон картирования (коэффициент весов), определяется полем CA5\_CFG1\_BITSEL регистра CA5\_CFG1 (см. описание CA5\_xx\_CFG1).

Работа подканалов (от 1 до 4) CA5 включается записью ненулевого значения в соответствующем поле CA5\_SPC1... CA5\_SPC4 регистра CA5\_CFG2, 0 подканалов включены если по крайней мере один из CA5\_SPC1... CA5\_SPC4 ненулевой. Если все подканалы отключены, то CA5 переходит в режим низкого потребления мощности. Таким образом, запрещение всех пяти подканалов отключает весь канал.

Режим работы устанавливается битом CA5\_CFG1\_SINGLE в регистре CA5\_CFG1; 0 соответствует непрерывной работе, 1 - одиночному накоплению. В зависимости от режима работы канала, запись '1' в поле CA5\_CFG1\_START регистра CA5\_CFG1 вызывает запуск одиночного или непрерывного накопления с номера эпохи, заданного в поле CA5\_CFG2\_ESTART регистра CA5\_CFG2.

Состоянием канала можно управлять чтением бита CA5\_CFG1\_START, который останется в состоянии '1', пока не запустится накопление с новыми параметрами.

Номер эпохи, которая должна запустить работу канала, задается полем CA5\_CFG2\_ESTART в регистре CA5\_CFG2. Это 8-разрядное число в конце каждой эпохи сравнивается с восемью младшими битами счетчика эпохи. В случае совпадения канал

запускается. Таким образом, чтобы запустить канал в следующую эпоху, текущий номер эпохи должно быть указано в CA5\_CFG2\_ESTART.

Продолжительность накопления устанавливается в регистре CA5\_CFG2\_ACCTIME значением от 0 до 15, которое определяет количество эпох (от 1 до 16) для непрерывного когерентного накопления. Минимальное значение поля CA5\_CFG1\_BITSEL должно быть установлено в регистре CA5\_CFG1 так, чтобы избежать возможности переполнения.

Результаты обработки канала доступны при использовании коллектора (см. п. 6).

## 20.3.2 Регистры управления CA5

Адресное пространство управляющих регистров CA5 запускается при 0x0200 с блока регистров CA5\_0. Адреса блоков CA5\_хх выравниваются при помощи 0x0004, где 'хх' в имени регистра обозначает номер канала от 0 до 19. Абсолютный адрес регистра получается добавлением базового адреса нужного блока канала и определенного ниже смещения регистра к основе пула.

Например, адрес регистра CA5\_05\_FHET (канал CA5\_5) определяется:

$$\&\text{CA5\_05\_FHET} = 0\text{x}0200 + 5 * 0\text{x}10 + 0\text{x}04 = 0\text{x}0254$$

Все 20 блоков регистров идентичны первому (обозначен 0).

Все регистры являются регистрами чтения-записи, если не указано иначе. В поля, отмеченные как 'зарезервированные', следует писать только значение '0', считываемое значение не определяется и им нужно пренебречь.

Любой неопределенный бит является зарезервированным.

Карта регистров одного CA5\_0 представлена в таблице ниже:

**Таблица 20.3. Регистры CA5\_0**

Адрес	Обозначение	Доступ	Сброс	Описание
0x0200	CA5_0_PCHIP	RW	0x00000000	Начальная фаза чипа
0x0204	CA5_0_FHET	RW	0x00000000	Коэффициент частоты гетеродина
0x0208	CA5_0_CFG1	RW	0x00000000	Установки регистра 1 канала
0x020C	CA5_0_CFG2	RW	0x00000000	Установки регистра 2 канала

### 20.3.2.1 CA5\_хх\_PCHIP

Смещение: 0x00

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.4. Регистр CA5\_xx\_PCHIP**

Биты	Обозначение	Доступ	Сброс	Описание
31	-	-	-	Зарезервировано
30:0	CA5_PCHIP	RW	0x0	Начальное значение аккумулятора фазы чипа

Регистр используется, для установки начального значения аккумулятора, которое используется для формирования чиповой частоты канала. Таким образом это значение определяет начальную фазу PN-чипа. Значение 0x7FFFFFFF соответствует максимальному сдвигу одного чипа минус отсчет CLK, значение ноль соответствует нелуovому смещению. Это значение применяется при ближайшем событии эпохи. Значение не имеет знака, 31-битное.

### 20.3.2.2 CA5\_xx\_FHET

Смещение: 0x04

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.5. Регистр CA5\_xx\_FHET**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	CA5_FHET	RW	0x0	Коэффициент частоты гетеродина

Регистр используется для обозначения коэффициента синтезатора частоты цифрового гетеродина. Код описывается формулой:

$$CA5\_FHET = \text{round}(2^{32} * (F_{het} / F_{clk})) ,$$

где:

$F_{het}$  - частота гетеродина;

$F_{clk}$  - частота синхронизации.

Данное значение применяется при событии ближайшей эпохи. Коэффициент обозначается 32-битным значением.

### 20.3.2.3 CA5\_xx\_CFG1

Смещение: 0x08

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.6. Регистр CA5\_xx\_CFG1**

Биты	Обозначение	Доступ	Сброс	Описание
9:0	CA5_CFG1_G1	RW	0x0	Начальное состояние регистра генератора кода G1 или сдвиг кода.
10	CA5_CFG1_SINGLE	RW	0x0	Канал в режиме “одиночный старт”
11	CA5_CFG1_START	RW	0x0	Начальный канал
12	CA5_CFG1_GPGL	RW	0x0	Установка GPS ('0')/GLONASS ('1')
15:13	-	-	-	Зарезервировано
25:16	CA5_CFG1_G2	RW	0x0	Начальное состояние регистра генератора кода G2
28:26	-	-	-	Зарезервировано
31:29	CA5_CFG1_BITSEL	RW	0x0	Коэффициента веса выходных данных

Поля CA5\_CFG1\_G1 и CA5\_CFG1\_G2 определяют начальные значения для регистров G1 и G2 генератора PN-кода. Начальное состояние PN-генератора загружается из этих полей в сдвиговые регистры. PN-генераторы запускаются при этом начальном значении, что соответствует временному сдвигу между событием локальной эпохи и эпохи принятого сигнала.

Бит CA5\_CFG1\_SINGLE активирует “одиночное исполнение” режима канала: в этом режиме канал работает по всей длине эпох CA5\_CFG2\_ACCTIME и переходит в дежурный режим. Если этот режим отключен, то канал, зафиксировавшись однажды, будет последовательно перезапускать накопление после завершения.

Запись '1' в бит CA5\_CFG1\_START включает запуск канала CA5 при наступлении эпохи, определенной CA5\_CFG2\_ESTART. Если все биты CA5\_CFG2\_SPCx в регистре CA5\_CFG2 установлены в ноль, канал считается отключенным и любая попытка запуска игнорируется.

Бит CA5\_CFG1\_GPGL переключает тип GNSS: '0' – GPS, '1' – GLONASS.

Кроме того, проверка бита CA5\_CFG1\_START обеспечивает конвейер конфигурации данных. Считанное из бита значение '1' означает, что значения управления еще не применены, и при записи новых данных предыдущие данные будут перезаписаны. Считанный '0' означает, что были применены предыдущие данные, и регистры интерфейса готовы принять новые данные без перезаписи предыдущих.

Поле CA5\_CFG1\_BITSEL служит для выбора диапазона аккумуляторов для выхода. Аккумуляторы канала имеют разрядность 23 бит. На выходах канала из них представлены только 16 бит. Диапазон битов определен в таблице ниже:



**Таблица 20.7. Выбор битов (масштабирование) полем BITSEL**

0b000	[15:0]
0b001	[16:1]
0b010	[17:2]
0b011	[18:3]
0b100	[19:4]
0b101	[20:5]
0b110	[21:6]
0b111	[22:7]

**20.3.2.4 CA5\_xx\_CFG2**

Смещение: 0x0C

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.8. Регистр CA5\_xx\_CFG2**

Биты	Обозначение	Доступ	Сброс	Описание
3:0	CA5_CFG2_ACCTIME	RW	0x0	Длительность когерентного накопления CA5, эпохи + 1
7:4	-	-	-	Зарезервировано
15:8	CA5_CFG2_ESTART	RW	0x0	Номер эпохи, назначенной стартовому каналу
19:16	CA5_CFG2_SPC1	RW	0x0	Задержка подканала #1
23:20	CA5_CFG2_SPC2	RW	0x0	Задержка подканала #2
27:24	CA5_CFG2_SPC3	RW	0x0	Задержка подканала #3
31:28	CA5_CFG2_SPC4	RW	0x0	Задержка подканала #4

Поле CA5\_CFG2\_ACCTIME устанавливает продолжительность когерентного накопления в канале CA5. Значение 0x0 соответствует одной эпохе, 0xF – 16 эпохам.

Поле CA5\_CFG2\_ESTART определяет момент времени, когда канал должен запустить накопление. Канал запускается, когда 8-битное значение в CA5\_CFG2\_ESTART совпадает с 8 младшими битами в счетчике эпох в регистре TB\_ECNT. При соответствии бита накопление начинается со следующей эпохи.

Поля CA5\_CFG2\_SPCx определяют задержки между подканалами CA5 (интервалы подканалов). Задержка подканала 0 всегда устанавливается в ноль. Сигналы в подканалах от 1 до 4 задерживаются на количество отсчетов, определенных в полях CA5\_CFG2\_SPCx. Минимальная задержка равняется 1, максимальная - 15 отсчетам. Кроме того, эти поля служат для включения подканалов. Работа подканалов (от 1 до 4) CA5 включается записью ненулевого значения в соответствующем поле CA5\_CFG2\_SPC1... CA5\_CFG2\_SPC4 в регистре CA5\_CFG2; включено 0 подканалов, если хотя бы один из CA5\_CFG2\_SPC1... CA5\_CFG2\_SPC4 ненулевой. Если отключены все подканалы, CA5 переключается в режим низкого потребления мощности. Запрещение всех пяти подканалов отключает весь канал.

## 20.4 FSE (устройство быстрого поиска)

### 20.4.1 Обзор FSE

Устройство быстрого поиска (FSE) разработано для осуществления поиска GPS L1, GLONASS L1 и L2. Поиск выполняется путем, проверяя всю Доплеровскую частоту и гипотезы задержки для каждого спутникового PN-кода. Результатом поиска является максимальная амплитуда результирующего пика корреляции и его позиция во временном фрейме (задержка) относительно локального события эпохи. Точность результата поиска составляет 0.5 us во времени.

FSE вычисляет функции взаимной корреляции входного сигнала и PN-кода в 2046 (GPS) / 2044 (GLONASS) позиций временного сдвига. Значения корреляции когерентно накапливаются в течение 1-16 эпох и затем результат нескольких (от 1 до 16), когерентных накоплений накапливаются по абсолютной величине (некогерентно). Результаты становятся доступны по интерфейсу в конце обработки гипотезы.

Блок-схема FSE показана на рисунке ниже.

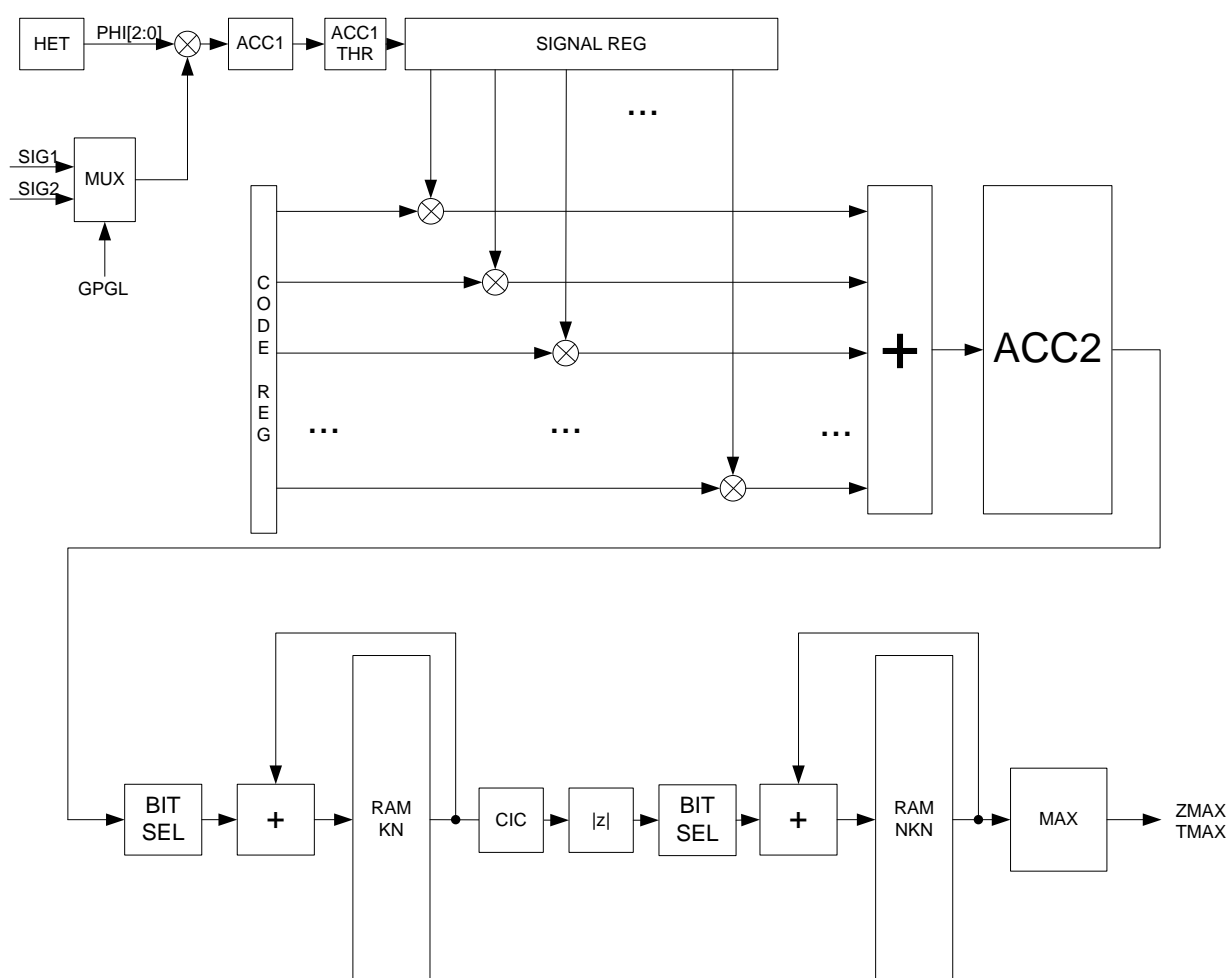


Рисунок 20.3. Блок-схема FSE

Тип входного сигнала (GPS или GLONASS) выбирается в поле FSE\_CFG\_GPGL регистра FSE\_CFG. Устройство может находиться либо в дежурном режиме, либо в режиме поиска. В дежурном режиме обработка входных данных не выполняется, и результаты последнего поиска сохраняются в регистрах интерфейса устройства. Все результаты последнего некогерентного накопления доступны через АНВ. CPU активирует режим поиска после записи конфигурации для проверки следующей гипотезы путем записи в регистр FSE\_CFG. После записи регистра накопление запускается следующим локальным событием эпохи.

Отсчеты сигнала умножаются на  $\exp(-j\omega t)$  Доплера гетеродина, частота которого задается регистром FSE\_FHET. Результат умножения накапливается в течение длительности обработки чипа, что определяется регистром FSE\_FCHIP. Затем накопленное значение квантуется тремя уровнями -1, 0, 1 с порогом, заданным полем FSE\_CFG\_ACC1THR регистра FSE\_CFG, и сохраняется в сдвиговом регистре на базе памяти. Максимальная длина этой линии задержки равняется 2046 отсчетам. 2046 отсчетов используются в случае GPS, а 2044 отсчетов - для GLONASS. Заметим, что в этом случае длина чипа считается частью чипа действительного сигнала. Часть времени чипа обработки относительно длины чипа действительного сигнала используется для улучшения точности поиска.

При помощи данных сдвигового регистра и сохраненной в памяти опорной последовательности, механизм согласованного фильтра вычисляет один выходной отсчет на один предварительно накопленный отсчет. Необходимое для каждого выходного отсчета реальное время определяется SCLK. Для вычисления одного отсчета требуется 32 цикла. Следовательно, SCLK clock должен быть по крайней мере в 32 раза быстрее чиповой скорости обработки.

Комплексный отсчет корреляционной функции накапливается в буфере когерентных накоплений в позиции, соответствующей фактической задержке между опорным PN-кодом и полученным сигналом. При завершении накопления текущего отсчета согласованного фильтра линия задержки смещается на 1, и следующий предварительно накопленный отсчет сохраняется в стартовой позиции. Затем вычисляется следующая выборка и накапливается в буфере когерентных данных.

После того, как локальное событие эпохи происходит снова, в буфере когерентных накоплений оказывается 2046 отсчетов. Таким образом, при запуске следующей эпохи когерентных накоплений, к ранее сохраненным в когерентном буфере данным добавляются отсчеты согласованного фильтра. Количество эпох когерентного накопления определяется полем FSE\_CFG\_CA регистра FSE\_CFG.

По завершении когерентного накопления отсчеты данных передаются через CIC-фильтр (который восстанавливает фактическую длину чипа сигнала) и модуль вычисления абсолютного значения, а затем накапливаются в буфере некогерентных данных. Буфер когерентных данных очищается, и когерентное накопление запускается снова. Количество некогерентных накоплений задается полем FSE\_CFG\_NCA регистра FSE\_CFG. После того,

как все требуемое накопление завершено, в некогерентном буфере происходит поиск максимального значения. Индекс максимума и его абсолютное значение составляют результат оценки гипотезы.

Другим критерием остановки поиска является абсолютный порог. Если какая-то ячейка некогерентного буфера достигает порога, заданного в поле FSE\_CFG\_STOPTHR регистра FSE\_CFG, это означает наличие очень сильного навигационного сигнала в позиции. В этом случае можно взять раннее решение о проверке гипотезы. Этот механизм позволяет сэкономить некоторое время при выполнении процедуры поиска.

#### 20.4.1.1 Когерентные накопления

Отсчеты входных данных отформатированы как квадратурный сигнал с 2 битами на компонент числа. Формат данных каждого компонента - “знак + амплитуда”. Старший бит содержит знак сигнала, в то время как второй параметр определяет выбор величины либо 1, либо 3 абсолютных значений. Таким образом, полный набор уровней отсчетов: (-3, -1, 1, 3).

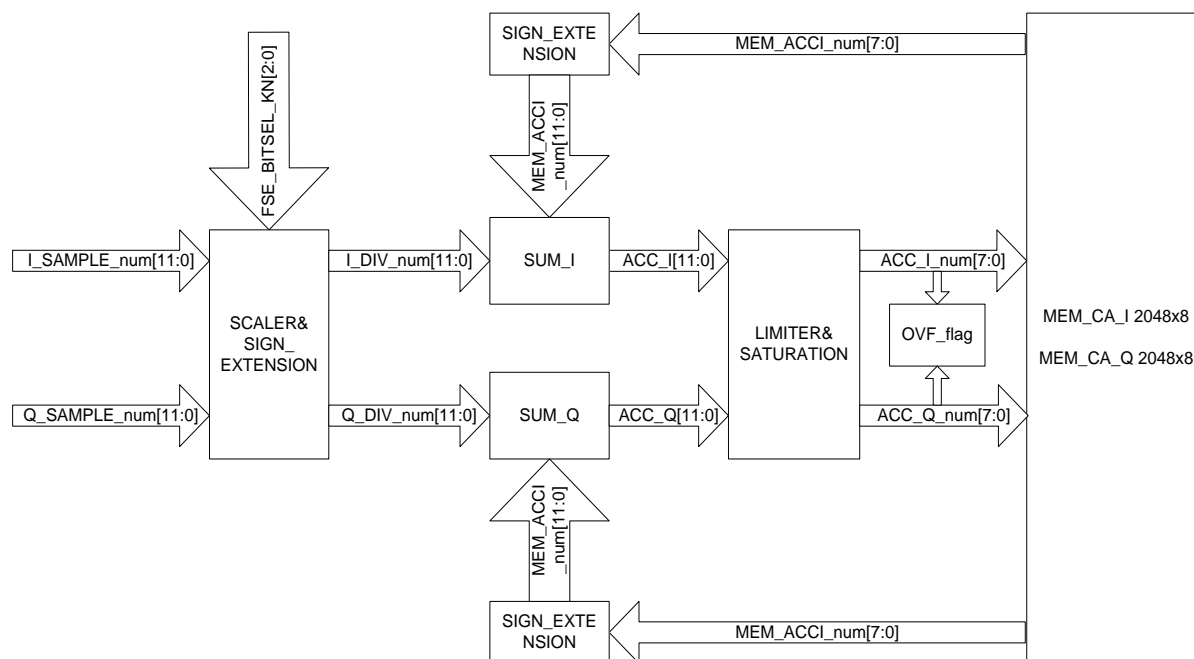
Сигнал гетеродина представляет собой 8-фазовый почти гармонический сигнал. Установка квантования фазы - (0, 45, 90, 135, 180, 225, 270, 315). Уровни квантования компонента - (-1, -0.5, 0.5, 1).

После перемножения входных отсчетов на отсчеты гетеродина, полученные компоненты комплексного сигнала имеют двоично-дополнительно симметрично ограниченный формат. Уровни квантования для каждого компонента принадлежат набору (-3, -2, -1, 0, 1, 2, 3).

8-битный регистр накопления, отмеченный на схеме ACC1, используется для накопления во время обработки чипа. Разрядность сигнала уменьшается до формата двух двоично-дополнительных бит и 3 уровней квантования (-1, 0, 1), чтобы упростить механизм вычисления согласованного фильтра. Механизм такого сокращения основан на сравнении абсолютных уровней сигнала относительно порога. Пороговое значение устанавливается при записи в регистр FSE\_CFG\_ACC1THR.

Отсчеты с уменьшенной разрядностью помещаются в регистр сдвига сигнала, обозначенный на рисунке 3 “SIGNAL REG”. Значения, записанные ранее, смещаются по одной позиции в сторону конца регистра. Опорная последовательность кода записывается в “CODE REG” при помощи CPU на этапе конфигурирования FSE. Одноразрядные отсчеты кода обрабатываются как (-1, 1) числа.

Массив перемножителей вычисляет скалярное произведение сигнала и отсчетов кода, которое представляет единственную выборку свертки. Результат сохраняется в комплексном 12-бит на компонент аккумуляторе ACC2. На рисунке ниже предоставлена подробная информация об обработке содержимого ACC2 в механизме когерентного накопления.



**Рисунок 20.4. Устройство когерентных накоплений**

На рисунке показан цикл обновления буфера памяти когерентного накопления. В начале когерентного накопления FSE сбрасывает индекс буфера и очищает области памяти MEM\_CA\_I и MEM\_CA\_Q. После этого начинается повторение последовательности когерентного накопления данных для каждого значения индекса. После каждого накопления индекс постепенно инкрементируется на 1. Весь индексный цикл от 0 до максимального значения 2047 повторяется несколько раз. Количество повторений определяется полем KN регистра FSE\_ACCTIME.

I\_SAMPLE\_num и Q\_SAMPLE\_num представляют собой двоично-дополнительные форматированные компоненты, взятые из ACC2. Прежде всего, эти компоненты понижаются (сдвигаются) на значение, определенное полем FSE\_CFG\_BITSEL\_CA. Эта операция необходима, чтобы предотвратить насыщение во время когерентного накопления на мощных сигналах. Знак компонентов расширяется согласно стандартной логике двоичного дополнения. Затем ранее накопленное комплексное значение извлекается из буферов памяти MEM\_KN согласно текущему индексу и добавляется к I\_DIV. Проверенный результат должен быть в пределах 8 бит диапазона ячеек буфера [7:0]. Если максимум превышен, значение насыщается – вместо расчетного значения используется максимальное 8-битное значение со знаком. В этом случае устанавливается флаг FSE\_OVF.

После насыщенности значение накопления записывается обратно в память MEM\_CA в той же индексной позиции. Затем индекс постепенно инкрементируется для обработки следующей позиции задержки. Операция повторяется для всех возможных индексов. Когда индекс достигает максимума, он либо сбрасывается, если количество когерентных накоплений не превышено, либо когерентное накопление останавливается, и данные передаются в модуль некогерентного накопления.

### 20.4.1.2 Некогерентные накопления

Этап некогерентного накопления запускается после завершения когерентного накопления. Содержание буфера когерентного накопления применяется для процедуры некогерентного накопления, затем буфер когерентного накопления очищается и запускается новое когерентное накопление.

Так как когерентное накопление выполняется в течение половины (GPS) или четверти (GLONASS) чипа сигнала, необходимо собрать всю энергию чипа, комбинируя 2 или 4 смежных отсчета в буфере когерентного накопления. С этой целью после буфера когерентного накопления помещен CIC-фильтр. Длина фильтра равна 2 выборкам для сигнала GPS и 4 выборкам для GLONASS. Программное обеспечение, при интерпретации результатов работы FSE, должно знать, что фильтр вводит дополнительную задержку пути сигнала. Задержка равна половине порядка CIC фильтра.

После CIC вычисляется амплитуда отсчетов когерентного накопления путем приближения: сравниваются величины двух 8-битных компонентов со знаком и затем меньшая половина прибавляется к большей. Затем приблизительная амплитуда усекается в соответствии с полем FSE\_CFG\_BITSEL\_NCA регистра FSE\_CFG и накапливается с насыщением в 8-разрядном буфере некогерентного накопления.

## 20.4.2 Описание регистров FSE

Адресное пространство регистров FSE запускается в 0x0100 с блока регистров FSE. Абсолютный адрес регистра получен путем сложения базового адреса и смещения регистра, определенного ниже, с базовым адресом пула.

Например, адрес регистра FSE\_CFG описан формулой:

$$\&\text{FSE\_FHET} = 0\text{x}0100 + 0\text{x}04 = 0\text{x}0104$$

Все регистры являются регистрами чтения-записи, если не указано иное. Поля, отмеченные 'зарезервировано', должны содержать запись только значения '0', значение чтения не определяется и им нужно пренебречь.

Любой не определенный бит зарезервирован.

Карта регистров одного устройства FSE приведена в Таблице ниже:

**Таблица 20.9. Регистры FSE**

Адрес	Обозначение	Доступ	Сброс	Описание
0x0100	FSE_FHET	RW	0x00000000	Коэффициент частоты гетеродина

0x0104	FSE_CFG	RW	0x850000FF	Режим FSE
0x0108	FSE_ACC	RW	0x00000000	Режим накопления FSE
0x010C	FSE_PR SIN	WO	0x00000000	Регистр загрузки PRS
0x0110	FSE_MAX	R	0x00000000	Результат поиска
0x0114	FSE_NCA_RES	R	0x00000000	Результаты некогерентного накопления
0x0118	FSE_FCHIP	RW	0x20000000	Коэффициент чиповой скорости FSE
0x011C-0x01FC	N/A			Зарезервировано

Описания отдельных регистров FSE даны ниже.

#### 20.4.2.1 FSE\_FHET

Смещение: 0x00

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.10. Регистр FSE\_FHET**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	FSE_FHET	RW	0x0	Коэффициент частоты гетеродина

Частота гетеродина устанавливается в регистре FSE\_x\_FHET. Значение для записи в регистр описывается формулой:

$$FSE\_FHET = \text{round}(2^{32} * (Fhet / Fclk)),$$

где:

Fhet - частота гетеродина (со знаком).

При расчете значения Fhet следует учитывать доплеровский сдвиг частоты и ID спутника GLONASS.

#### 20.4.2.2 FSE\_CFG

Смещение: 0x04

Значение сброса: 0x850000FF

Доступ: Чтение/запись в любой момент

**Таблица 20.11. Регистр FSE\_CFG**

Биты	Обозначение	Доступ	Сброс	Описание
15:0	FSE_CFG_STOPTHR	RW	0xFF	Порог остановки поиска
18:16	FSE_CFG_BITSEL_CA	RW	0x0	Вес входных данных когерентного аккумулятора 000b [11:4] 001b [11:3] 010b [11:2] 011b [11:1] 100b - 111b [11:0]

19	FSE_CFG_GPGL	RW	0x0	Установка GPS ('0')/GLONASS ('1')
22:20	FSE_CFG_BITSEL_NCA	RW	0x0	Вес входных данных некогерентного аккумулятора 000b [9:7] 001b [9:6] 010b [9:5] 011b [9:4] 100b [9:3] 101b [9:2] 110b [9:1] 111b [9:0]
26:23	FSE_CFG_ACC1THR	RW	0xA	Порог предварительного аккумулятора ACC1
28:27	-	-	-	Зарезервировано
29	FSE_CFG_NCA_RES_CLR	W	0x0	Сброс счетчика чтения некогерентных накоплений FSE
30	FSE_CFG_PRIS_CLR	W	0x0	Сброс счетчика загрузки PRS
31	FSE_CFG_RDY	R	0x1	Флаг готовности FSE

Регистр FSE\_CFG используется для конфигурации FSE.

Поле FSE\_CFG\_STOPTHR устанавливает порог для раннего завершения поиска. Если какое-либо значение в некогерентном буфере накопления превышает этот порог, поиск преждевременно останавливается, и позиция этого значения сохраняется.

Поле FSE\_CFG\_ACC1THR устанавливает порог, используемый для квантования сигнала после первого накопления в ACC1. Если величина ACC1 со знаком превышает порог, значение “-1” или “1” сохраняется в буфере SIGNAL REG. Знак совпадает со знаком ACC1. В противном случае сохраняется “0”.

Поля FSE\_CFG\_BITSEL\_CA и FSE\_CFG\_BITSEL\_NCA управляют двумя устройствами масштабирования и расширения знака. Одним из них, расположенным перед буфером когерентного накопления, управляет поле FSE\_CFG\_BITSEL\_CA. Другое устройство, которым управляет FSE\_CFG\_BITSEL\_NCA, размещено перед буфером некогерентного накопления. Обе процедуры масштабирования необходимы для приведения к соответствию разрядности регистров накопления разрядности памяти. Разрядность памяти выбирается меньше, чтобы сэкономить память, необходимую для FSE. Значения, загруженные в эти поля, должны выбираться согласно количествам когерентного и некогерентного накоплений и спекулятивному уровню сигнала на входе устройства.

Бит FSE\_CFG\_GPGL переключает тип GNSS, '0' – GPS, '1' – GLONASS.

Запись '1' в FSE\_CFG\_PRIS\_CLR очищает счетчик загрузки PRS.

Установка FSE\_CFG\_RDY означает, что FSE готов, и регистр FSE\_MAX содержит информацию о пиковом значении и пиковой позиции. Запись в регистр FSE\_ACC очищает поле FSE\_CFG\_RDY.



### 20.4.2.3 FSE\_ACC

Смещение: 0x08

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.12. Регистр FSE\_ACC**

Биты	Обозначение	Доступ	Сброс	Описание
3:0	FSE_ACC_CA	RW	0x0	Количество когерентных накоплений
13:4	FSE_ACC_NCA	RW	0x0	Количество некогерентных накоплений
31:14	-	-	-	Зарезервировано

Поля FSE\_ACC\_CA и FSE\_ACC\_NCA устанавливают параметры времени накопления результатов поиска. Значение, записанное в поле FSE\_ACC\_CA, определяет количество когерентных накоплений, а значение, записанное в поле FSE\_ACC\_NCA, – некогерентных.

Количество итераций каждого вида накопления выводится из этих двух значений прибавлением 1. Полное время обработки FSE (измеряется в эпохах) может быть выражено как  $\{(FSE\_ACC\_CA+1)*(FSE\_ACC\_NCA+1) + 1\}$ .

Запись в регистр FSE\_ACC включает механизм быстрого поиска.

### 20.4.2.4 FSE\_PR SIN

Смещение: 0x0C

Доступ: Запись в любое время

**Таблица 20.13. Регистр FSE\_PR SIN**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	FSE_PR SIN	W	-	Регистр последовательной загрузки PN-кода

PN-последовательность загружается в этот регистр пословно до конца. Младший бит PN-последовательности - первый. Каждое значение PN-последовательности помещается дважды и четырежды для GPS и GLONASS, соответственно. Остальные биты последнего слова должны быть заполнены стартовыми битами данной последовательности, что обеспечивает лучшую точность поиска.

### 20.4.2.5 FSE\_MAX

Смещение: 0x10

Значение сброса: 0x00000000

Доступ: Чтение в любое время

**Таблица 20.14. Регистр FSE\_MAX**

Биты	Обозначение	Доступ	Сброс	Описание
------	-------------	--------	-------	----------

15:0	FSE_MAX_ZMAX	R	0x0	Пиковое значение
26:16	FSE_MAX_TMAX	R	0x0	Пиковая позиция
30:27	-	-	-	Зарезервировано
31	FSE_MAX_OVF	R	0x0	В течение когерентного накопления детектируется переполнение

Оценка амплитуды и наиболее вероятная позиция сигнала, найденного FSE, представлены в регистре FSE\_MAX.

Поле FSE\_MAX\_ZMAX содержит оценку амплитуды в диапазоне от 0 до 65535.

Поле FSE\_MAX\_TMAX содержит максимальный индекс позиции от 0 до 2047.

В случае раннего завершения поиска при превышении порога эти два поля содержат данные переполненной позиции.

Состояния поисковых устройств доступны в регистре IRQ\_STAT\_FSE. FSE устанавливает соответствующий бит готовности в '1' при завершении процедуры поиска. После чтения данных из IRQ\_STAT\_FSE ЦП должен очистить флаг готовности при записи '1' в IRQ\_ACK\_FSE в соответствующей позиции. Запись нуля не изменяет флаг состояния, и таким образом, программное обеспечение может очистить флаги по отдельности.

#### 20.4.2.6 FSE\_NCA\_RES

Смещение: 0x14

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

Действует: если в регистре FSE\_CFG установлен бит FSE\_CFG\_RDY

**Таблица 20.15. Регистр FSE\_NCA\_RES**

Биты	Обозначение	Доступ	Сброс	Описание
31:16	NA			
15:0	FSE_NCA_RES	R	0x0	Данные некогерентного накопления FSE из памяти NCA

Результаты ненакопления доступны при помощи FSE\_NCA\_RES, когда накопление закончено (флаг готовности FSE\_CFG\_RDY установлен в '1'). Накопления могут быть считаны пословно (16 бит), последние 2 или 4 бита (GPS и GLONASS, соответственно) в последнем (128-м) слове не определяются и ими нужно пренебречь.

#### 20.4.2.7 FSE\_FCHIP

Смещение: 0x18

Значение сброса: 0x20000000

Доступ: Чтение/запись в любой момент

**Таблица 20.16. Регистр FSE\_FCHIP**

Биты	Обозначение	Доступ	Сброс	Описание
31	NA			Зарезервировано
30:0	FSE_FCHIP	RW	0x20000000	Коэффициент чиповой скорости FSE

Регистр FSE\_FCHIP определяет чиповую частоту обработки. Значение регистра определяется формулой:

$$\text{FSE\_FCHIP} = \text{round}(2^{33} * (\text{Fc} / \text{Fclk})),$$

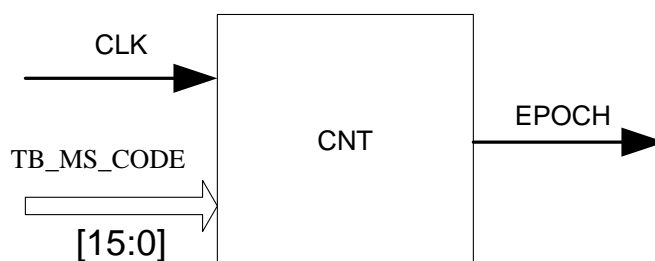
где:

Fc - чиповая частота обработки

Fclk - частота дискретизации RFFE.

Чиповая частота обработки предполагается 1.023 МГц или 1.022 МГц для GPS или GLONASS соответственно. Если требуются длинные когерентные накопления, следует принять во внимание эффект Доплера чиповой частоты.

## 20.5 Генератор временных интервалов



**Рисунок 20.5. Модуль RECV\_TIME\_BASE**

Модуль генератора временных интервалов представляет собой таймер на основе делителя частоты. Его значение инициализации определяет коэффициент деления. Модуль генерирует локальный импульс EPOCH. Длительность импульса составляет один период CLK. Период следования импульсов следует установить максимально близко к 1 мс. В качестве основания используется импульс синхронизации (CLK) частоты дискретизации RFFE. Значение инициализации вычитающего счетчика устанавливается в регистре TB\_MS\_CODE.

## 20.5.1 Регистры генератора временных интервалов

### 20.5.1.1 TB\_ECNT

Смещение: 0x08

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.17. Регистр TB\_ECNT**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	TB_ECNT	RW	0x0	Счетчик ЕРОСН

Регистр представляет собой 32-битный счетчик ЕРОСН и служит для синхронизации. Он инкрементируется на 1 при каждом событии ЕРОСН. Когда счетчик полон, он циклически переходит в нулевое значение. Младшие 8 бит регистра определяют момент запуска когерентного накопления в каналах СА5.

### 20.5.1.2 TB\_MS\_CODE

Смещение: 0x0C

Значение сброса: 0x00003FF0

Доступ: Чтение/запись в любой момент

**Таблица 20.18. Регистр TB\_MS\_CODE**

Биты	Обозначение	Доступ	Сброс	Описание
15:0	TB_MS_CODE	RW	0x3FF0	Коэффициент деления частоты
31:16	-	-	-	Зарезервировано

Значение для записи в поле TB\_MS\_CODE определяется формулой:

$TB\_MS\_CODE = \text{round}(Fclk/1000);$

## 20.6 Сборщик данных СА5

При каждом событии ЕРОСН механизм сервиса сбора данных проверяет состояние готовности всех 20 каналов слежения СА5. Для каждого канала, в котором имеется активный сигнал готовности данных, сборщик сохраняет данные с выходов СА5\_хх\_АССх в буфер FIFO. Глубина FIFO составляет 512 32-битных слов. Данные с выхода FIFO доступны для последовательного чтения регистром COL\_DATA.

Данные каждого чтения организуются в виде записи из 6 слов:

**Таблица 20.19.**

# слова	Содержит
0	CA5_xx_ACC0
1	CA5_xx_ACC1
2	CA5_xx_ACC2
3	CA5_xx_ACC3
4	CA5_xx_ACC4
5	{номер канала CA5, TB_ECNT[26:0]}

**20.6.1.1 COL\_CFG**

Смещение: 0x0070

Значение сброса: 0x00000004

Доступ: Чтение/запись в любой момент

**Таблица 20.20. Регистр COL\_CFG**

Биты	Обозначение	Доступ	Сброс	Описание
0	COL_ENABLE	RW	0x0	1 – устройство сбора данных включено
1	COL_FCLEAR	W	0x0	запись '1' очищает буфер FIFO данных
2	COL_FEMPTY	R	0x1	1 – FIFO пуст
3	COL_FFULL	R	0x0	1 – FIFO полон
5:4	-	-	-	Зарезервировано
6	COL_AFULL	R	0x0	1 – FIFO почти полон
13:7	-	-	-	Зарезервировано
16:14	COL_THR	RW	0x0	почти полный порог (см. Таблицу)
31:17	-	-	-	Зарезервировано

**Таблица 20.21. Почти полный порог COL\_THR**

COL_THR	Порог (32-битные слова)
0b000	0
0b001	64
0b010	128
0b011	192
0b100	256
0b101	320
0b110	386
0b111	448

**20.6.1.2 COL\_DATA**

Смещение: 0x0074

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

**Таблица 20.22. Регистр COL\_DATA**

Биты	Обозначение	Доступ	Сброс	Описание
------	-------------	--------	-------	----------

31:0	COL_DATA	R	0x0	Структурированный поток выходных данных CA5 с выхода FIFO
------	----------	---	-----	---

## 20.7 Модуль формирования секундных импульсов (PPS)

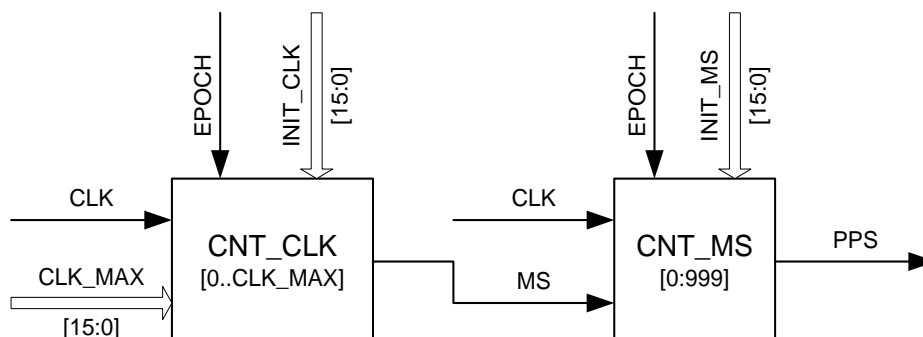


Рисунок 20.6. Блок-схема модуля формирования PPS

Модуль предназначен для генерации высокоточного сигнала Pulse-Per-Second (PPS), синхронизированного с глобальным временем GNSS. Модуль PPS обеспечивает средства слежения за импульсом относительно локальных часов таким образом, чтобы он оставался синхронным относительно второй метки сигналов, принятых со спутника.

Модуль генерации PPS состоит из двух последовательно соединенных таймеров/счетчиков, которыми управляют регистры PPS\_OFS и PPS\_MS.

Первыми коэффициентами деления частоты для CLK управляет поле PPS\_MS\_CODE регистра PPS\_MS. Второй фиксирован и равен 1000.

Длительностью импульса управляет поле PPS\_MS\_LEN. Единица измерения - период CLK.

Помимо периода и длительности импульса программное обеспечение может установить начальные значения для счетчиков MS и CLK, что позволяет более точно устанавливать время возникновения события относительно EPOCH PPS опорного сигнала.

### 20.7.1 Регистры PPS

#### 20.7.1.1 PPS\_MS

Смещение: 0x18

Значение сброса: 0x00013FF0

Доступ: Чтение/запись в любой момент

Таблица 20.23. Регистр PPS\_MS

Биты	Обозначение	Доступ	Сброс	Описание
------	-------------	--------	-------	----------

15:0	PPS_MS_CODE	RW	0x3FF0	Коэффициент деления частоты для быстрого (первого) таймера
31:16	PPS_MS_LEN	RW	0x1	Ширина PPS в периодах CLK

Поле PPS\_MS\_CODE определяет коэффициент деления частоты для первого таймера в цепочке. Значение определяется формулой:

$$\text{PPS\_MS\_CODE} = \text{round}(\text{Fclk}/1000) - 1$$

Значение PPS\_MS\_LEN обозначает количество периодов CLK, когда импульс PPS активен.

### 20.7.1.2 PPS\_OFS

Смещение: 0x1C

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.24. Регистр PPS\_OFS**

Биты	Обозначение	Доступ	Сброс	Описание
15:0	PPS_OFS_CLK_INIT	RW	0x0	Начальное значение для счетчика PPS CLK
31:16	PPS_OFS_MS_INIT	RW	0x0	Начальное значение для счетчика PPS MS

Два поля регистра определяют начальные значения для загрузки в оба таймера при наличии события EPOCH.

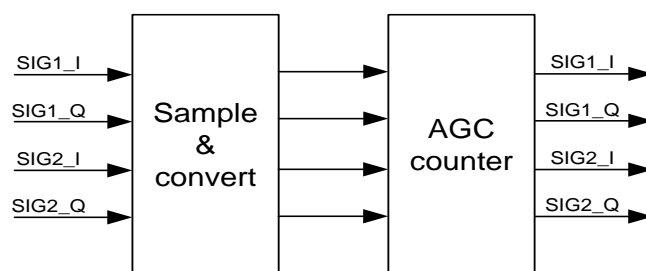
## 20.8 Интерфейс ADC

Назначение интерфейса ADC:

Получение отсчетов входного сигнала одного из фронтов отсчета выборки с дальнейшей передачей сигнала по переднему фронту

Преобразование отсчетов набора поддерживаемых представлений в “знак + амплитуда”

Вычисление повреждений уровней входного сигнала для регулировки усиления



### Рисунок 20.7. Схема интерфейса ADC (RFFE)

Поддерживаемые представления входных отсчетов описаны в Таблице ниже.

**Таблица 20.25. Поддерживаемые представления входных отсчетов**

Представление	Код	Значение
Двоично-дополнительный	0b00	+1
	0b01	+3
	0b10	-3
	0b11	-1
Знак + амплитуда	0b00	+1
	0b01	+3
	0b10	-1
	0b11	-3
Амплитуда одиночного бита 1	0b0	+1
	0b1	-1
Амплитуда одиночного бита 3	0b0	+3
	0b1	-3

Подблок счетчика AGC предназначен для вычисления среднего уровня входных сигналов от радио. Для оценки уровней сигналов в каждом подканале у подблока имеется четыре аккумулятора на каждый компонент сигнала. Количество состояний '1' между двумя событиями EPOCH рассчитывается для каждого бита всех компонентов входного сигнала. Аккумуляторы зафиксированы к 6 выходным регистрам ADC\_SIG\_x и очищаются при каждом событии EPOCH.

## 20.8.1 Регистры ADC

### 20.8.1.1 ADC\_SIG1\_I

Смещение: 0x20

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

**Таблица 20.26. Регистр ADC\_SIG1\_I**

Биты	Обозначение	Доступ	Сброс	Описание
15:0	ADC_SIG1_I_0	R	0x0	Количество состояний '1' бита 0
31:16	ADC_SIG1_I_1	R	0x0	Количество состояний '1' бита 1

Счетчики состояния бита '1' для SIG\_I за последний временной интервал EPOCH.

### 20.8.1.2 ADC\_SIG1\_Q

Смещение: 0x24

Значение сброса: 0x00000000

Доступ: Чтение в любой момент



Таблица 20.27. Регистр ADC\_SIG1\_Q

Биты	Обозначение	Доступ	Сброс	Описание
15:0	ADC_SIG1_Q_0	R	0x0	Количество состояний '1' бита 0
31:16	ADC_SIG1_Q_1	R	0x0	Количество состояний '1' бита 1

Счетчики состояния бита '1' для SIG\_Q за последний временной интервал EPOCH.

### 20.8.1.3 ADC\_SIG2\_I

Смещение: 0x28

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

Таблица 20.28. Регистр ADC\_SIG2\_I

Биты	Обозначение	Доступ	Сброс	Описание
15:0	ADC_SIG2_I_0	R	0x0	Количество состояний '1' бита 0
31:16	ADC_SIG2_I_1	R	0x0	Количество состояний '1' бита 1

Счетчики состояния бита '1' для SIG\_I за последний временной интервал EPOCH.

### 20.8.1.4 ADC\_SIG2\_Q

Смещение: 0x2C

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

Таблица 20.29. Регистр ADC\_SIG2\_Q

Биты	Обозначение	Доступ	Сброс	Описание
15:0	ADC_SIG2_Q_0	R	0x0	Количество состояний '1' бита 0
31:16	ADC_SIG2_Q_1	R	0x0	Количество состояний '1' бита 1

Счетчики состояния бита '1' для SIG\_Q за последний временной интервал EPOCH.

### 20.8.1.5 ADC\_CTRL

Input RFFE data representation and sampling control register.

Смещение: 0x30

Значение сброса: 0x00000053

Доступ: Чтение/запись в любой момент

Таблица 20.30. Регистр ADC\_CTRL

Биты	Обозначение	Доступ	Сброс	Описание
0	ADC_CTRL_SIG1_EDGE	RW	0x1	Выбор фронта SIG1. '1' – нарастающий фронт
1	ADC_CTRL_SIG2_EDGE	RW	0x1	Выбор фронта SIG2. '1' – нарастающий фронт

2	-	-	-	Зарезервировано
4:3	ADC_CTRL_SIG1_ENC	RW	0x2	Представление данных SIG1
6:5	ADC_CTRL_SIG2_ENC	RW	0x2	Представление данных SIG2
7	ADC_CTRL_IQ_SWAP	RW	0x0	Перестановка I и Q компонентов
8	ADC_CTRL_GNSS_SWAP	RW	0x0	Перестановка SIG1 и SIG2
31:9	-	-	-	Зарезервировано

Поля ADC\_CTRL\_SIGx\_EDGE управляют фронтом, который фиксирует входные данные. Значение '1' означает передний фронт, '0' - падающий. Данные всегда преобразовываются в выборку переднего фронта до передачи данных в модули обработки Navicore.

Поля ADC\_CTRL\_SIGx\_ENC выбирают представление входных данных согласно следующей таблице.

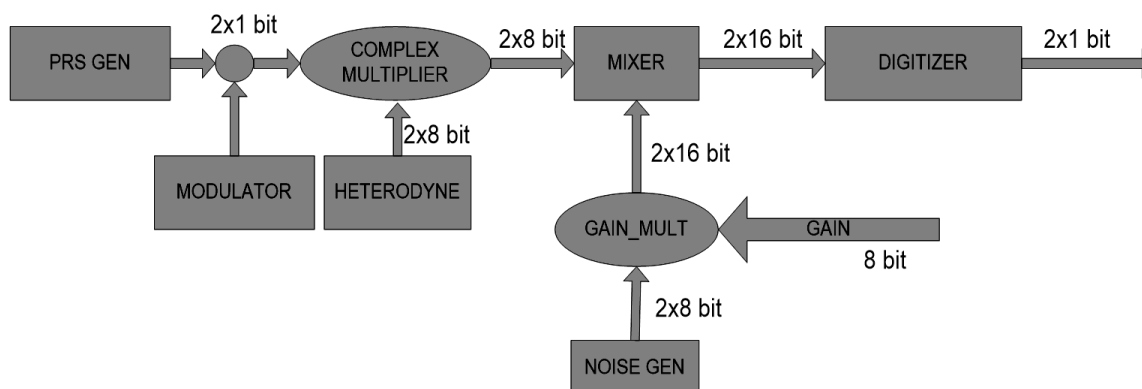
**Таблица 20.31.**

ADC_CTRL_SIGx_ENC state	Представление
0b00	2-битный, двоично-дополнительный
0b01	2-битный знаковый + амплитуда
0b10	1-бит (используется старший), амплитуда 1
0b11	1-бит (используется старший), амплитуда 3

Компоненты I и Q можно поменять местами при помощи поля ADC\_CTRL\_IQ\_SWAP; SIG1 и SIG2 также могут быть перестановлены полем ADC\_CTRL\_GNSS\_SWAP.

## 20.9 IMITATOR (симулятор сигнала спутника)

Симулятор разработан для формирования сигнала, который с достаточной надежностью воспроизводит сигнал одного из спутников GPS или GLONASS. Режим сигнала определен IMIT\_CFG\_GPGL. Этот сигнал может использоваться вместо входного сигнала от радиоприемника для целей отладки программного обеспечения.



**Рисунок 20.8. Схема имитатора**

Подблоки PRS\_GEN, MODULATOR и HETERODYNE моделируют спутниковый сигнал с модулированными данными, программируемым Доплеровским сдвигом частоты, который может быть смешан вместе с шумом. Программируемое усиление шума обеспечивает моделирование различных величин SNR. Модуль DIGITIZER преобразовывает сигнал в 2-битное представление со знаком + амплитуда, которое может применяться на входах RFFE MCC. Коэффициент чиповой скорости определяется регистрами FCHIP\_CA5\_GPS или FCHIP\_CA5\_GLO в зависимости от используемого режима GNSS.

## 20.9.1 Регистры имитатора

### 20.9.1.1 IMIT\_PCHIP

Смещение: 0x0034

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.32. Регистр IMIT\_PCHIP**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	IMIT_PCHIP	RW	0x00000000	Сдвиг фазы чипа

Регистр устанавливает начальное значение аккумулятора, которое используется для формирования чиповой частоты имитатора. Таким образом, это значение определяет начальную фазу PN-чипа. Значение 0xFFFFFFFF соответствует максимальному чиповому сдвигу 1 минус 1 CLK, нулевое значение соответствует отсутствию сдвига. Это значение будет применено при ближайшем событии эпохи. Значение не имеет знака, 32-разрядное.

### 20.9.1.2 IMIT\_FDOP

Смещение: 0x0038

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.33. Регистр IMIT\_FDOP**

Биты	Обозначение	Доступ	Сброс	Описание
31:0	IMIT_FDOP	RW	0x00000000	Коэффициент частоты Допплера

Доплеровский гетеродин генерирует квазигармонический сигнал, который используется для моделирования Доплеровского сдвига частоты. Частота определена этим регистром. Значение коэффициента вычисляется по формуле:

$$\text{IMIT\_FDOP} = \text{round}(2^{32} * (F_{\text{het}} / F_{\text{clk}}))$$

Данное значение применяется при событии ближайшей эпохи. Коэффициент представляет собой 32-битное значение без знака.

### 20.9.1.3 IMIT\_G1G2

Смещение: 0x003C

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.34. Регистр IMIT\_G1G2**

Биты	Обозначение	Доступ	Сброс	Описание
9:0	IMIT_G1G2_G1	RW	0x0	Начальное состояние G1 или G
15:10	-	-	-	Зарезервировано
25:16	IMIT_G1G2_G2	RW	0x0	Начальное состояние G2
31:26	-	-	-	Зарезервировано

Поля IMIT\_G1G2\_G1 и IMIT\_G1G2\_G2 определяют начальные состояния генераторов кода G1 и G2. Эти состояния обеспечивают моделирование ID спутника и моделирование задержки сигнала. Данные значения загружаются в сдвиговые регистры G1 и G2 генератора событием EPOCH. Структура генераторов соответствует документации на GNSS.

### 20.9.1.4 IMIT\_CFG

Смещение: 0x0040

Значение сброса: 0x00000018

Доступ: Чтение/запись в любой момент

**Таблица 20.35. Регистр IMIT\_CFG**

Биты	Обозначение	Доступ	Сброс	Описание
0	IMIT_CFG_EN	RW	0x0	Имитатор включен: '1' – включен
2:1	-	-	-	Зарезервировано
3	IMIT_CFG_ESYNC	RW	0x1	Режим синхронной PN-последовательности: '1' – включен
4	IMIT_CFG_MOD	RW	0x1	Моделирование модуляции данных (50 Гц): '1' – включен
5	IMIT_CFG_GPGL	RW	0x0	Установка GPS ('0')/GLONASS ('1')
31:6	-	-	-	Зарезервировано

Бит IMIT\_CFG\_EN включает симулятор и переключает входы Navicore на выход симулятора вместо входов интерфейса ADC.

IMIT\_CFG\_ESYNC включает начало перезагрузки кода G1&G2 событием EPOCH. В Имитаторе будет использоваться один и тот же стартовый код в каждом цикле, при отключении режима синхронизации.

IMIT\_MOD включает синхронизированную с EPOCH меандровую модуляцию 50 Гц сгенерированных PN-последовательностей. Эта модуляция моделирует наличие навигационных данных в сигнале.

### 20.9.1.5 IMIT\_NPF

Смещение: 0x0044

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

**Таблица 20.36. Регистр IMIT\_NPF**

Биты	Обозначение	Доступ	Сброс	Описание
7:0	IMIT_NPF	RW	0x0	Коэффициент мощности шума
31:8	-	-	-	Зарезервировано

Регистр используется для настройки SNR на выходе симулятора.

Поля IMIT\_NPF определяют, сколько раз амплитуда шума превысила амплитуду полезного сигнала. Значение имеет знак и допустимо в диапазоне от 0 до 127.

## 20.10 Регистры служебных модулей Navicore

### 20.10.1 Генератор начального состояния G1-G2

Служебный блок ISG генерирует начальное состояние генераторов PN-кода в соответствии с данной позицией кода.

#### 20.10.1.1 ISG\_CHIP

Смещение: 0x0050

Значение сброса: 0x80000000

Доступ: Чтение/запись в любой момент

**Таблица 20.37. Регистр ISG\_CHIP**

Биты	Обозначение	Доступ	Сброс	Описание
9:0	ISG_CHIP_G1	RW	0x0	Позиция G1 или G
15:10	-	-	-	Зарезервировано
25:16	ISG_CHIP_G2	RW	0x0	Позиция G2
30:26	-	-	-	Зарезервировано
31	ISG_CHIP_READY	R	0x1	'1' если готовы данные в ISG_GPS/ISG_GLO

Чтобы получить значение инициализации кода для данных позиций кода, необходимо загрузить смещения позиции в поля ISG\_CHIP\_G1 и ISG\_CHIP\_G2 и проверить, что бит ISG\_CHIP\_READY установлен. Установка бита ISG\_CHIP\_READY означает, что регистр данных содержит допустимый код для соответствующего GNSS.

### 20.10.1.2 ISG\_GPS

Смещение: 0x0054

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

Допустимо: когда бит ISG\_READY установлен в регистре ISG\_CHIP

**Таблица 20.38. Регистр ISG\_GPS**

Биты	Обозначение	Доступ	Сброс	Описание
9:0	ISG_GPS_G1	R	0x0	Код G1
15:10	-	-	-	Зарезервировано
25:16	ISG_GPS_G2	R	0x0	Код G2
31:26	-	-	-	Зарезервировано

### 20.10.1.3 ISG\_GLO

Смещение: 0x0058

Значение сброса: 0x00000000

Доступ: Чтение в любой момент

Допустимо: когда бит ISG\_READY установлен в регистре ISG\_CHIP

**Таблица 20.39. Регистр ISG\_GLO**

Биты	Обозначение	Доступ	Сброс	Описание
8:0	ISG_GLO_G	R	0x0	Код G
31:9	-	-	-	Зарезервировано

## 20.10.2 Управление прерываниями

### 20.10.2.1 IRQ\_CFG

Смещение: 0x0060

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

Таблица 20.40. Регистр IRQ\_CFG

Биты	Обозначение	Доступ	Сброс	Описание
5:0	IRQ_CFG_EPOCH_DEC	RW	0x0	Коэффициент децимации события EPOCH +1. Количество пустых 1мс импульсов между событиями EPOCH. 0 - без децимации. Макс. значение равно 63, что соответствует периоду EPOCH 64 мс
26:6	-	-	-	Зарезервировано
27	IRQ_CFG_COL	RW	0x0	1 – Прерывание по событию готовности устройства сбора данных CA5 включено
28	IRQ_CFG_EPOCH	RW	0x0	1 – Прерывание по децимированной EPOCH включено
31:29	-	-	-	Зарезервировано

### 20.10.2.2 IRQ\_STAT

Смещение: 0x0064

Значение сброса: 0x00000000

Доступ: Чтение/запись в любой момент

Таблица 20.41. Регистр IRQ\_STAT

Биты	Обозначение	Доступ	Сброс	Описание
26:0	-	-	-	Зарезервировано
27	IRQ_STAT_COL	RW	0x0	1 – Прерывание устройства сбора данных CA5 включено
28	IRQ_STAT_EPOCH	RW	0x0	1 – Прерывание по EPOCH включено
31:29	-	-	-	Зарезервировано

### 20.10.2.3 IRQ\_ACK

Смещение: 0x0068

Значение сброса: 0x00000000

Доступ: Запись в любой момент

Таблица 20.42. Регистр IRQ\_ACK

Биты	Обозначение	Доступ	Сброс	Описание
26:0	-	-	-	Зарезервировано
27	IRQ_ACK_COL	RW	0x0	1 – Сигнал подтверждения прерывания от устройства сбора данных CA5
28	IRQ_ACK_EPOCH	RW	0x0	1 – Сигнал подтверждения прерывания от EPOCH
31:29	-	-	-	Зарезервировано

## 20.10.3 Управление чиповой частотой

### 20.10.3.1 FCHIP\_CA5\_GPS

Смещение: 0x10

Значение сброса: 0x20000000

Доступ: Чтение/запись в любой момент

**Таблица 20.43. Регистр FCHIP\_CA5\_GPS**

Биты	Обозначение	Доступ	Сброс	Описание
31	NA			Зарезервировано
30:0	FCHIP_CA5_GPS	RW	0x20000000	Коэффициент чиповой скорости GPS

Регистр FCHIP\_CA5\_GPS определяет чиповую частоту обработки. Значение в регистре определяется формулой:

$$FCHIP\_CA5\_GPS = \text{round}(2^{33} * (F_c / F_{clk})),$$

где:

$F_c$  - чиповая частота для обработки

$F_{clk}$  - частота дискретизации RFFE

Для GPS предполагается чиповая частота обработки 1.023 МГц. При необходимости в длинных когерентных накоплениях следует учитывать Доплеровский сдвиг частоты чипа. FCHIP\_GPS используется для каналов CA5 и симулятора спутникового сигнала, работающего в режиме GLONASS.

### 20.10.3.2 FCHIP\_CA5\_GLO

Смещение: 0x14

Значение сброса: 0x20000000

Доступ: Чтение/запись в любой момент

**Таблица 20.44. Регистр FCHIP\_CA5\_GLO**

Биты	Обозначение	Доступ	Сброс	Описание
31	NA			Зарезервировано
30:0	FCHIP_CA5_GLO	RW	0x20000000	Коэффициент чиповой скорости GLONASS

Регистр FCHIP\_CA5\_GLO определяет чиповую частоту обработки. Значение в регистре определяется формулой:

$$FCHIP\_CA5\_GLO = \text{round}(2^{33} * (F_c / F_{clk})),$$

где:



$F_c$  - чиповая частота для обработки

$F_{clk}$  - частота дискретизации RFFE

Для GLONASS предполагается чиповая частота обработки 1.022 МГц. При необходимости в длинных когерентных накоплениях следует учитывать Доплеровский сдвиг частоты чипа. FCHIP\_GLO используется для каналов CA5 и симулятора спутникового сигнала, работающего в режиме GLONASS.

## 21. SDMMC: ИНТЕРФЕЙС КАРТ SECURE DIGITAL INPUT/OUTPUT И MULTI MEDIA

### 21.1 Основные особенности

- Совместимость со стандартами:
  - SD Host Controller Standard Specification Version 3.00
  - SDIO card specification Version 3.0
  - SD Memory Card Specification Version 3.01
  - SD Memory Card Security Specification version 1.01
  - MMC Specification version 4.51
- Системный/хост интерфейс
  - Пересылка в режиме PIO через слейв интерфейс хоста, пересылка в режиме DMA через мастер интерфейс хоста.
- SD/ SDIO интерфейс
  - Частота хоста от 0 до 52 МГц
  - Поддержка режимов Default speed (DS) и High speed (HS)
  - Поддержка режимов SDR12, SDR25 и DDR50 с внешним переключателем напряжения
  - До 400 Мбит/с в 4-битном режиме (DDR50)
  - Пересылка в 1-битном и 4-битном режимах SD
  - Контроль целостности CRC7 для команд и CRC16 для данных
  - Настраиваемый размер пересылок
  - Поддержка I/O, Read-only и Read/Write карт
  - Поддержка операций “Read wait Control”, “Suspend/Resume”
- MMC интерфейс
  - Частота хоста от 0 до 52 МГц
  - До 416 Мбит/с в 8-ми битном режиме (mmc8 bit SDR)
  - До 832 Мбит/с в 8-ми битном режиме (mmc8 bit DDR)
  - Пересылка в 1-битном, 4-битном и 8-битном режимах
  - Контроль целостности CRC7 для команд и CRC16 для данных
  - Поддержка MMC Plus и MMC Mobile
  - Обнаружение карты (Вставка / Удаление)
- Разное
  - 1 Кбайтное FIFO для пересылок между CPU и контроллером
  - Приостановка частоты SD при переполнении или недогрузке FIFO

### 21.2 Параметры

Таблица 21.1. Параметры

Параметр/статический вход	Значение
---------------------------	----------

SDHC_BUFFER_SIZE	256
corecfg_asyncwkupena	1
corecfg_tuningcount	32
corecfg_timeoutclkfreq	1
corecfg_timeoutclkunit	1
corecfg_maxblklength	0
corecfg_8bitsupport	1
corecfg_adma2support	1
corecfg_highspeedsupport	1
corecfg_sdmasupport	1
corecfg_suspressupport	1
corecfg_asynchintrsupport	1
corecfg_64bitsupport	0
corecfg_sdr104support	0
corecfg_retuningmodes	0
corecfg_clockmultiplier	0
corecfg_spisupport	1
corecfg_spiblkmode	0
corecfg_sdr104presetval	0

## 21.3 Регистры

### 21.3.1 Перечень регистров

Поддерживаются байтовые, 2-х байтовые и 4-х байтовые обращения.

Неопределенные области памяти зарезервированы. При чтении таких областей возвращается ноль, записи игнорируются.

**Таблица 21.2. Регистры SDMMC**

Условное обозначение	Название регистра	Исходное состояние	Тип доступа	Смещение
	См. регистры контроллера			0x000 – 0x0FF
SDMMC_ITAPCTR	Управления задержкой входов	0x0000 0000	RW	0x100
SDMMC_OTAPCTR	Управления задержкой выходов	0x0000 0000	RW	0x104
SDMMC_CORECFG0	Регистр конфигурации ядра 0	0x0400 0000	RW	0x108
SDMMC_CORECFG1	Регистр конфигурации ядра 1	0x0000 0000	RW	0x10C
SDMMC_CORECFG2	Регистр конфигурации ядра 2	0x0001 0040	RW	0x110
SDMMC_CORECFG3	Регистр конфигурации ядра 3	0x0001 0000	RW	0x114
SDMMC_CORECFG4	Регистр конфигурации ядра 4	0x0000 0000	RW	0x118
SDMMC_CORECFG5	Регистр конфигурации ядра 5	0x0000 0000	RW	0x11C

### 21.3.2 Регистр SDMMC\_ITAPCTR

Таблица 21.3. Регистр SDMMC\_ITAPCTR

Номер бита	Условное Обозначение	Назначение
31:13	-	Резерв
12:8	ITAPDLYSEL	Выбирает один из 32 вентилях задержки на сигнале <code>gxclk_in</code> . Действует при установленном ITAPDLYEN и отключённой подстройке.
7:2	-	Резерв
1	ITAPCHGWIN	Используется для гейтирования выходов вентилях задержки, чтобы избежать распространения глитчей внутрь ядра. Сигнал должен быть установлен за несколько тактов до изменения сигналов <code>corectrl_itapdlysel</code> и удерживаться несколько тактов после.
0	ITAPDLYEN	Используется для включения выборочной вентилях задержки на сигнале <code>gxclk_in</code> (возвращенной с падов /looped back частоты SD). Совместно с ITAPDLYSEL[4:0] определяет величину задержки. При включенной подстройке (для SDR104 и опционально для SDR50) игнорируется и вместо данного сигнала используется внутреннее управление. Не следует устанавливать в режиме DS.

### 21.3.3 Регистр SDMMC\_OTAPCTR

Таблица 21.4. Регистр SDMMC\_OTAPCTR

Номер бита	Условное обозначение	Назначение
31:12	-	Резерв
11:8	OTAPDLYSEL	Выбирает один из 16 вентилях задержки на сигнале <code>sdcard_clk</code> . Действует при установленном OTAPDLYEN.
7:1	-	Резерв
0	OTAPDLYEN	Используется для включения выборочной вентилях задержки на сигнале <code>sdcard_clk</code> , для формирования задержанной версии <code>sdcard_clk</code> . Используется для защёлкивания CMD/DAT выходов, чтобы создать на них задержку относительно выходного сигнала CLK. Используется совместно с OTAPDLYSEL[3:0] для выбора величины задержки на сигнале тактирования. Не следует устанавливать в режиме DS.

### 21.3.4 Регистр SDMMC\_CORECFG0

Таблица 21.5. Регистр SDMMC\_CORECFG0

Номер бита	Условное обозначение	Назначение
31:27	-	Резерв
26	VSPT3P3	Поддержка 3,3В Предполагаемое значение - 1'b1, так как напряжение по умолчанию для SD интерфейса равно 3,3 В.
25	VSPT3P0	Поддержка 3,0В Определяется поддержкой 3,0 В.
24	VSPT1P8	Поддержка 1,8В Предполагаемое значение - 1'b1 (Ядро поддерживает переключение в режим 1,8В).
23:22	-	Резерв

Номер бита	Условное обозначение	Назначение
21	SDR50SPT	Поддержка режима SDR50
20	DDR50SPT	Поддержка режима DDR50
19	-	Резерв
18	ADRVSP	Поддержка драйверов типа A (Driver Type A). Устанавливать в 1 при поддержке драйверов типа A в режиме 1,8 В.
17	CDRVSP	Поддержка драйверов типа C. Устанавливать в 1 при поддержке драйверов типа C в режиме 1,8 В.
16	DDRVSP	Поддержка драйверов типа D. Устанавливать в 1 при поддержке драйверов типа D в режиме 1,8 В.
15	-	Резерв
14:13	SLOTTYPE	Тип слота 00 – Сменный (Removable SDCard Slot) 01 – Встроенный (Embedded Slot for One Device) 10 – Разделяемая шина (Shared Bus Slot) 11 – Зарезервировано Тип слота отображается в регистре Capabilities, а также используется логикой обнаружения карты (Card Detect Logic) для определения длительности времени дрейфа. Если SLOTTYPE=2'b01 (встроенный слот), для событий детектирования/вставки карты используется меньшее время. Для всех остальных значений при определении CD# используется гораздо большее время
12	SDR50TUNING	Использовать подстройку для SDR50. Бит должен быть установлен, если программе необходимо выполнить подстройку для режима SDR50. Ядро работает с подстройкой или без в режиме SDR50, так частота может быть подстроена вручную, используя tap delay.
11:8	RETUNETIMCNT	Счётчик перенастройки (Timer Count for Re-Tuning). Счётчик для Re-Tuning режимов 1-3. Установка в 4'b0 отключает Re-Tuning таймер.
7:0	BASEFREQ	Базовая частота SD. Частота входа xin_clk.

### 21.3.5 Регистр SDMMC\_CORECFG1

Таблица 21.6. Регистр SDMMC\_CORECFG1

Номер бита	Условное обозначение	Назначение
31:24	-	Резерв
23:16	MAXCUR1P8V	Максимальный ток для режима 1,8 В
15:8	MAXCUR3P0V	Максимальный ток для режима 3,0 В
7:0	MAXCUR3P3V	Максимальный ток для режима 3,3 В

### 21.3.6 Регистр SDMMC\_CORECFG2

Таблица 21.7. Регистр SDMMC\_CORECFG2

Номер бита	Условное обозначение	Назначение
31:26	-	Резерв
25:16	DSPRESETVAL	Выбор частоты SDCLK для режима Default Speed
15:10	-	Резерв
9:0	INITPRESETVAL	Выбор частоты SDCLK для режима Initialization

### 21.3.7 Регистр SDMMC\_CORECFG3

Таблица 21.8. Регистр SDMMC\_CORECFG3

Номер бита	Условное обозначение	Назначение
31:26	-	Резерв
25:16	SDR12PRESETVAL	Выбор частоты SDCLK для режима SDR12
15:10	-	Резерв
9:0	HSPRESETVAL	Выбор частоты SDCLK для режима High Speed

### 21.3.8 Регистр SDMMC\_CORECFG4

Таблица 21.9. Регистр SDMMC\_CORECFG4

Номер бита	Условное обозначение	Назначение
31:26	-	Резерв
25:16	SDR50PRESETVAL	Выбор частоты SDCLK для режима SDR50
15:10	-	Резерв
9:0	SDR25PRESETVAL	Выбор частоты SDCLK для режима SDR25

### 21.3.9 Регистр SDMMC\_CORECFG5

Таблица 21.10. Регистр SDMMC\_CORECFG5

Номер бита	Условное обозначение	Назначение
31:10	-	Резерв
9:0	DDR50PRESETVAL	Выбор частоты SDCLK для режима DDR50

## 22. USB: UNIVERSAL SERIAL BUS 2.0 OTG

### 22.1 Основные особенности

Блок контроллера USB является двухролевым устройством и может выполнять функции периферийного устройства (Device), хоста (Host), а также поддерживает стандарт USB On-The-Go (дополнение к спецификации USB 2.0, Ревизия 1.3а и 2.0). Блок USB может быть сконфигурирован как хост-контроллер (Host-only) или как периферийное устройство (Device-only) с полной совместимостью со спецификацией USB 2.0.

Поддерживает следующие функции:

- программный выбор режима работы OTG1.3 или OTG2.0
  - поддержка ADP (Attach detection Protocol) протокола для micro-AB разъёма
- скорость передачи данных
  - High-Speed (HS, 480-Mbps)
  - Full-Speed (FS, 12-Mbps)
  - Low-Speed (LS, 1.5-Mbps)
- несколько вариантов управления энергопотреблением
- использование встроенного DMA для передачи данных внутри SoC
- возможность работы как с интегрированным PHY по UTMI+ интерфейсу (не поддерживает LS в режиме девайса) так и с внешним по ULPI интерфейсу
- возможность работы с 8-ю двунаправленными эндпоинтами включая управляющий 0-й эндпоинт
- поддержка до 16-ти хост каналов
- поддержка автоматического пинга

### 22.2 Описание функционирования

#### 22.2.1 Включение питания

Ниже описана последовательность подачи сигналов сброс при включении питания.

1. Перевести «POR» в активное состояние (  $\text{USBSYSRESETN}[\text{RESETN}] = 0$  ) и подождать 10 микросекунд.
2. Через регистры настройки установить нужные значения на следующих сигналах:
  - FSEL
  - WORDINTERFACE
  - PLLITUNE
  - PLLPTUNE
  - PLLBTUNE
  - EFUSEHSPB0
  - EFUSESEL0
  - EQBYPASSEN0
  - COMPDISTUNE0
  - SQRXTUNE0
  - VDATREFTUNE0
  - TXHSXVTUNE0
  - TXFSLSTUNE0
  - TXVREFTUNE0
  - TXRISETUNE0
  - TXRESTUNE0
  - TXPREEMPAMPTUNE0
  - TXPREEMPPULSETUNE0
3. Перевести POR из состояния 1'b0 в 1'b1 ( $\text{USBSYSRESETN}[\text{RESETN}] = 1$ )
4. Дождаться начала подачи синхросигнала (обычно 45 микросекунд)
5. Перевести PRSTN из состояния 1'b0 в 1'b1 ( $\text{USBSYSRESETN}[\text{PRSTN}] = 1$ )
6. Подождать минимум 6 тактов системной частоты
7. Перевести HRESETN из состояния 1'b0 в 1'b1 ( $\text{USBSYSRESETN}[\text{HRESETN}] = 1$ )



## 22.3 Регистры

Базовый адрес регистров USB 0x020\_0000

Регистры USB делятся на две группы:

1. Регистры контроллера (смещение относительно базового адреса 0x0\_0000).  
Описание представлено в документе на контроллер USB (глава «Control and Status Registers»)
2. Регистры настройки USB (смещение относительно базового адреса 0x4\_0000)

### 22.3.1 Регистры настройки USB

Таблица 22.1. Список регистров настройки USB

Условное обозначение	Описание	Исходное состояние	Смещение
USBSYSRESETN	Регистр управления сигналами сброса контроллера и физического уровня (PHY).	0x2	0x00040000
USBPHYREFFREQSEL	Регистр выбора референсной частоты PHY	0x3	0x00040004
USBPHYLOOPBACKTEST	Регистр тестирования PHY – режим обратной петли	0x0	0x00040008
USBCNTRSIMSCALEDOWN	Регистр управления временем моделирования (только для модели).	0x0	0x0004000C
USBCNTRFILTERBYPASS	Регистр отключения debounce фильтров (OTG-сигнал).	0x0	0x00040010
USBGENPARAMETERS	Регистр общих параметров PHY (зависящих от реализации в кремнии).	0x59470	0x00040014
USBTXPARAMETERS	Регистр параметров PHY TXTUNE (зависящих от реализации в кремнии).	0x14CF	0x00040018
USBCONFIGCHANNEL	Регистр настройки/мониторинга конфигурационного канала (CC).	0x10F	0x0004001C
USBPHYSLEEPN	Регистр перевода PHY в спящий режим.	0x1	0x00040020
USBOTGCONTROL	Регистр управления OTG сигналами	0x1	0x00040024
USBUTMICONTROL	Регистр управления UTMi сигналами	0x0	0x00040028
USBTESTANDDEBUG	Регистр управления сигналами отладки	0x0	0x00040000
USBCIRQCONTROL	Регистр управления запросами на прерывание CC IRQ	0x0	0x00040030

### 22.3.2 Регистр USBSYSRESETN

Регистр управления сигналами сброса контроллера и физического уровня (PHY).

Таблица 22.2. Регистр USBSYSRESETN

Bit Number	Name	Description	Reset
31:4	res0		?
3	HRESETN	Управление сбросом AHB Clock домена контроллера. Активный уровень «0»	1'b0

2	PRSTN	Управление сбросом PHY Clock домена контроллера. Активный уровень «0»	1'b0
1	PORTRESETN	Управление сбросом UTMI Clock домена физического уровня (PHY). Активный уровень «0»	1'b1
0	RESETN	Управление сбросом по включению питания (POR). Должен находиться в активном состоянии не менее 10 us. Активный уровень «0»	1'b0

### 22.3.3 Регистр USBPHYREFFREQSEL

Регистр выбора референсной частоты PHY

Таблица 22.3. Регистр USBPHYREFFREQSEL

Bit Number	Name	Description	Reset
31:4	res0		?
3	PLLBTUNE	Регулировка ширины полосы ФАПЧ - удваивает ширину полосы ФАПЧ по мере необходимости для некоторых входных опорных тактовых частот, как показано ниже: 1'b0 @ 16MHz (normal mode) и 50MHz (testburnin-mode) 1'b1 @ 19.2MHz, 20MHz, 24MHz, 26MHz и 32MHz	1'b0
2:0	FSEL	Выбор номинала входной частоты PHY (вместе с PLLBTUNE): 3'b000 = 19.2MHz 3'b001 = 20MHz 3'b010 = 24MHz 3'b011 = 16MHz (default) при PLLBTUNE=1'b0, в противном случае 32MHz 3'b100 = reserved 3'b101 = reserved 3'b110 = 26MHz 3'b111 = 50MHz (неиспользуемый 'testburnin' режим @ PLLBTUNE=1'b0)	3'b011

### 22.3.4 Регистр USBPHYLOOPBACKTEST

Регистр тестирования PHY – режим обратной петли.

Таблица 22.4. Регистр USBPHYLOOPBACKTEST

Bit Number	Name	Description	Reset
31:2	res0		?
1	TESTBURNIN	Выбор burn-in режима: 1'b0 = нормальный режим 1'b1 = burn-in test режим (этот режим также требует 50MHz на CLKCORE, FSEL[2:0]=3'b111 и PLLBTUNE=1'b0)	1'b0

0	LBDEBUG	Выбор loopback режима: 1'b0 = нормальный режим (логика приема отключена во время передачи данных) 1'b1 = loopback test режим (логика приема включена во время передачи данных)	1'b0
---	---------	--	------

### 22.3.5 Регистр USBCNTRSIMSCALEDOWN

Регистр управления временем моделирования (только для модели).

Таблица 22.5. Регистр USBCNTRSIMSCALEDOWN

Bit Number	Name	Description	Reset
31:2	res0		?
1:0	SCALE	Выбор режима ускорения моделирования (см. Стр. 255 в «controller databook» главу «scales-down simulation timing»): 2'b00 = ускорение отключено (используются фактические временные значения), используйте это значение во время синтеза; 2'b01 = разрешает уменьшение всех значений времени моделирования (кроме «suspend» и «resume» в режиме devace); 2'b10 = разрешает уменьшение значений времени моделирования в режиме devace «suspend» и «resume»; 2'b11 = разрешает оба варианта уменьшения времени моделирования.	2'b00

### 22.3.6 Регистр USBCNTRFILTERBYPASS

Регистр отключения debounce фильтров (OTG-сигнал).

Таблица 22.6. Регистр USBCNTRFILTERBYPASS

Bit Number	Name	Description	Reset
31:1	res0		?
0	BYPASS	Управление обходом debounce фильтров OTG-сигналов (avalid, bvalid, vbusvalid, sessend, iddig): 1'b1 = байпас включен 1'b0 = байпас выключен (default)	1'b0

### 22.3.7 Регистр USBGENPARAMETERS

Регистр общих параметров PHY (зависящих от реализации в кремнии).

Таблица 22.7. Регистр USBGENPARAMETERS

Bit Number	Name	Description	Reset
31:20	res0		?

19:18	VDATEFTUNE	Регулировка напряжения обнаружения данных (в этом поле настраивается пороговый уровень напряжения (Vdat_ref), используемый для обнаружения данных во время определения типа зарядного устройства): 2'b01 = по умолчанию	2'b01
17:15	SQRXTUNE	Регулировка порога шумоподавления (это поле регулирует уровень напряжения для порога, используемого для обнаружения действительных высокоскоростных данных): 3'b011 = по умолчанию	3'b011
14:12	COMPDISTUNE	Регулировка порога отключения (в этом поле настраивается уровень напряжения для порога, используемого для обнаружения события отключения на хосте): 3'b001 = по умолчанию	3'b001
11	EQBYPASSEN	Обход эквалайзера шумоподавления: 1'b0 = эквалайзер включен (по умолчанию) 1'b1 = эквалайзер выключен	1'b0
10:7	EFUSEMATH	Регулировка внутреннего сопротивления для достижения цели 200 (Ом) 4'b1000 (по умолчанию) = 2% вариация процесса	4'b1000
6	EFUSESEL	Включение внутреннего резистора: 1'b1 = внутренний резистор (по умолчанию), используется вместе с EFUSEMATH-полем 1'b0 = внешний резистор Rext 200 (Ом), подключенный к выводу TXRTUNE	1'b1
5:2	PLLPTUNE	Настройка пропорционального тракта PLL (см. controller databook, стр. 75): 4'b1100 (дизайн по умолчанию)	4'b1100
1:0	PLLITUNE	Настройка интегрального тракта PLL (см. controller databook, стр. 74): 2'00 = дизайн по умолчанию	2'b00

## 22.3.8 Регистр USBTXPARAMETERS

Регистр параметров PHY TXTUNE (зависящих от реализации в кремнии).

**Таблица 22.8. Регистр USBTXPARAMETERS**

Bit Number	Name	Description	Reset
31:17	res0		?
16	TXPREEMPULSETUNE	Контроль длительности pre-emphasis передатчика HS: в этом поле контролируется длительность, с которой ток pre-emphasis HS поступает на DP или DM («длительность pre-emphasis» определяется в единицах измерения - 1 единица составляет ~ 580ps, что определяется здесь как x1): Внимание! этот сигнал действителен, только если TXPREEMPAMPTUNE! = 2'b00  1'b0 = по умолчанию (удвоенная длительность pre-emphasis) 1'b1 = x1 короткая длительность pre-emphasis	1'b0

15:14	TXPREEMPAMP TUNE	Передачик HS pre-emphasis управление током: это поле контролирует величину тока, поступающего на DP0 и DM0 после перехода J-K или K-J. 2'b00 = проектное значение по умолчанию (передатчик HS pre-emphasis отключен). Примечание. Передатчик HS 'pre-emphasis current' определяется в единицах измерения (1 единица равна ~ 600 мкА, что здесь x1). Выбор варьируется от x0 (отключено) до x3.	2'b00
13:12	TXRESTUNE	Регулировка полного сопротивления источника USB: в некоторых приложениях может быть значительное последовательное сопротивление на трактах D + и D- между трансивером и кабелем. В этом поле настраивается полное сопротивление источника драйвера для компенсации дополнительного последовательного сопротивления на USB. 2'b01 = по умолчанию (изменение нулевого уровня)	2'b01
11:10	TXRISETUNE	Регулировка времени нарастания/спада HS передатчика: 2'b01 = значение по умолчанию (регулировка нулевого уровня)	2'b01
9:6	TXVREFTUNE	HS регулировка уровня DC-voltage: 4'b0011 = по умолчанию (регулировка нулевого уровня)	4'b0011
5:2	TXFSLSTUNE	Регулировка полного сопротивления источника FS/LS (Следующие значения регулировки основаны на номинальном процессе, напряжении и температуре): 4'b0011 = по умолчанию (регулировка нулевого уровня)	4'b0011
1:0	TXHSXVTUNE	Регулировка переключения передатчика HS (это поле регулирует напряжение, при котором сигналы DP0 и DM0 пересекаются при передаче в режиме HS): 2'b11 = настройка по умолчанию (настройка нуля)	2'b11

### 22.3.9 Регистр USBCONFIGCHANNEL

Регистр настройки/мониторинга конфигурационного канала (CC).

**Таблица 22.9. Регистр USBCONFIGCHANNEL**

Bit Number	Name	Description	Reset
31:11	res0		?
10	IDDIG	Индикатор mini-A/B разъема. Указывает, является ли подключенный штекер mini-A или mini-B Этот сигнал действителен, только если для сигнала CC1MODE=1'b1. 1'b1: Mini-A подсоединен 1'b0: Mini-B подсоединен	1'b0

9	CC1MODE	Режим работы вывода CC1 0 - CC1 как ID. CC1DIG подключен к регистру USBCONFIGCHANNEL[4], DMPULLDOWN0/DPPULLDOWN0 (входам PHY) и utmiotg_iddig (входу CTR). Можно использовать в качестве ID пина для micro-AB разъема. 1 - CC1DIG подключен только к регистру USBCONFIGCHANNEL[4]. Не участвует в управлении DMPULLDOWN0/DPPULLDOWN0 (PHY) и utmiotg_iddig (CTR).	1'b0
8	CCSEL	Выбор режима работы CC канала. 1: CC режим. CC1 и CC2 используются как функциональные КП; 0: High impedance режим. Если VATESTENB[1:0] = 2'b01, аналоговое напряжение может быть выведено на CC1.	1'b1
7:6	res0		?
5	CC2DIG	Бит статуса. Отображает значение КП CC2 (подробнее в описании физического уровня)	1'b0
4	CC1DIG	Бит статуса. Отображает значение КП CC1 (подробнее в описании физического уровня)	1'b0
3:1	CCTUNE	Биты настройки для порогового уровня CC (подробнее в описании физического уровня) 3'b000 = power down 3'bxx0 (за исключением значения 'power down') - 3'bxx1 = DFP (host) 3'b001 - 3'b111 = UPF (device)	3'b111
0	UFPDFP	Это поле используется в субблоках CC1 и CC2 для выбора: 1'b1 = UFP (device) mode 1'b0 = DFP (host) mode	1'b1

### 22.3.10 Регистр USBPHYSLEEPN

Регистр перевода PHY в спящий режим.

**Таблица 22.10. Регистр USBPHYSLEEPN**

Bit Number	Name	Description	Reset
31:2	res0		?
1	COMMONONN	Общее отключение питания (активный уровень низкий): Это поле управляет сигналами отключения питания REFCLK домена, bias-блока и PLL, если USB находится в режиме 'Suspend' или 'Sleep': 1'b1 = в 'suspend' refclk-логика, bias и PLL отключены; в режиме 'sleep' отключаются только bias и PLL; 1'b0 = в режимах 'suspend' или 'sleep', блоки refclk-logic, bias и PLL остаются включенными	1'b0
0	SLEEPN	SW управление сигналом Sleep (активный уровень низкий): 1'b1 = функциональный режим (default) 1'b0 = режим сна	1'b1

### 22.3.11 Регистр USBOTGCONTROL

Регистр управления OTG сигналами

Таблица 22.11. Регистр USBOTGCONTROL

Bit Number	Name	Description	Reset
31:3	res0		?
2	OTGSESSVLD0	OTG Device Session Valid индикатор (активный уровень низкий) RO бит отображающий значение сигнала «Session Valid» и указывает, находится ли напряжение на VBUS ниже порогового значения. 1: напряжение на VBUS выше порогового значения 0: напряжение на VBUS ниже порогового значения	1'b1
1	VBUSVLDEXTSEL0	Выбор источника сигнала валидности VBUS для установки нагрузочного резистор DP0. Активация подтягивающего резистора DP0 также зависит от состояния XCVRSEL0[1:0], OPMODE0[1:0], TERMSEL0, DPPULLDOWN0 и DMPULLDOWN0. Чтобы избежать возможных сбоев в DP, VBUSVLDEXTSEL0 должен быть статическим до сброса при включении питания и оставаться статичным во время нормальной работы. 1: Вход VBUSVLDEXT0 используется для задания подтягивающего резистора DP0. 0: Внутренний компаратор Session Valid используется для задания подтягивающего резистора DP0.	1'b0
0	OTGDISABLE0	Отключение OTG функционала PHY Если OTG функционал не используется, то его можно отключить для уменьшения потребления. 1: OTG блок отключен. 0: OTG блок включен.	1'b1

### 22.3.12 Регистр USBUTMICONTR0L

Регистр управления UTMI сигналами

Таблица 22.12. Регистр USBUTMICONTR0L

Bit Number	Name	Description	Reset
31:3	res0		?
2	TXBITSTUFFENH0	Управление режимом Transmit Bit-Stuffing для старшего байта Этот сигнал управляет заполнением битов на DATAINH0[7:0], когда OPMODE0[1:0] = 2'b11. 1: Bit-Stuffing включен. 0: Bit-Stuffing выключен.	1'b0

1	TXBITSTUFFEN 0	Управление режимом Transmit Bit-Stuffing для младшего байта Этот сигнал управляет заполнением битов на DATAIN0[7:0], когда OPMODE0[1:0] = 2'b11. 1: Bit-Stuffing включен. 0: Bit-Stuffing выключен.	1'b0
0	VBUSVLDEXT0	Внешний индикатор валидности VBUS Этот сигнал действителен в режиме Device и только в том случае, если для сигнала VBUSVLDEXTSEL0 установлено значение 1'b1. VBUSVLDEXT0 указывает наличие сигнала VBUS. Кроме того, VBUSVLDEXT0 включает нагрузочный резистор на линии D+. VBUSVLDEXT0 не изменяет выход OTGSESSVLD0. 1: сигнал VBUS действителен, и нагрузочный резистор на D+ включен. 0: сигнал VBUS недействителен, и подтягивающий резистор на D+ отключен.  В режиме Host этот регистр не используется и должен быть установлен в 1'b0.	1'b0

### 22.3.13 Регистр USBTESTANDDEBUG

Регистр управления сигналами отладки

Таблица 22.13. Регистр USBTESTANDDEBUG

Bit Number	Name	Description	Reset
31	TESTCLK0	Сигнал синхронизации для TESTDATAIN0[7:0] Это сигнал синхронизации для TESTDATAIN0[7:0]. Тестовый интерфейс использует положительный фронт TESTCLK0 для фиксации TESTDATAIN0[7:0].	1'b0
30	TESTDATAOUTSEL0	Выбор тестовых данных Этот сигнал определяет, какие данные подаются на TESTDATAOUT[3:0]. 1: выводится содержимое тестового регистра, определенного в режиме. 0: выводятся внутренне генерируемые сигналы определенные в режиме.	1'b0
29:24	res1		?
23:20	TESTDATAOUT0	Шина чтения тестовых данных Эта шина содержит тестовые данные, которые должны быть считаны с USB 2.0 femtoPHY, и представляет либо сгенерированные внутри себя сигналы, либо содержимое тестового регистра на основе значения TESTDATAOUTSEL0.	4'b0
19:16	TESTADDR0	Адрес регистра тестового интерфейса Эта шина задает адрес регистра для записи или чтения из регистра тестового интерфейса.	4'b0
15:8	TESTDATAIN0	Шина записи тестовых данных Эта шина содержит тестовые данные для записи на USB 2.0 femtoPHY. Тестовые данные представлены в битах [3:0], а соответствующее разрешение записи представлено в битах [7:4].	8'b0



7:4	res0		?
3:2	VATESTENB	<p>Выбор аналогового тестового пина</p> <p>Позволяет вывести аналоговое напряжение на вывод ANALOGTEST.</p> <p>11: Зарезервировано.</p> <p>10: Аналоговое напряжение выведено на площадку ANALOGTEST.</p> <p>01: Аналоговое напряжение выведено на площадку CC1 (CCSEL должен быть равен 1'b0 при использовании этого режима).</p> <p>00: Аналоговое напряжение не выводится.</p> <p>Используйте это, когда CCSEL = 1'b1</p>	1'b0
1	SIDDQ	<p>Тест IDDQ включен</p> <p>Этот тестовый сигнал позволяет выполнять тестирование IDDQ путем отключения всех аналоговых блоков.</p> <p>Перед утверждением SIDDQ убедитесь, что VDATSRCENB, VDATDETENB, DCDENB, BYPASSSEL и TESTBURNIN установлены в 1'b0.</p> <p>1: аналоговые блоки выключены.</p> <p>0: аналоговые блоки включены.</p> <p>Важно! Когда для SIDDQ установлено значение 1'b1 для проверки на утечку или для получения нефункционального состояния с низким энергопотреблением, рекомендуется также установить POR в 1'b1.</p> <p>При возвращении в функциональное состояние POR должен оставаться в 1'b1 в момент переключения SIDDQ в 1'b0. После переключения SIDDQ в 1'b0, POR следует удерживать в 1'b1 как минимум 10 мкс. Сигнал POR можно сбросить в 1'b0 не менее чем через 10 мкс после того, как все источники и входные сигналы PHY (включая SIDDQ) достигли стабильных допустимых состояний.</p>	1'b0
0	ATERESET	<p>Сигнал сброса автоматического тестового оборудования</p> <p>Когда USB 2.0 femtoPHY включен (не в режиме Suspend или Sleep режиме), автоматический тестер может использовать этот тестовый сигнал для отключения PHYCLOCK0 и FREECLK, а затем повторно включить их с выровненной фазой.</p> <p>1: Выходы PHYCLOCK0 и FREECLK отключены.</p> <p>0: выходы PHYCLOCK0 и FREECLK доступны в течение определенного периода после сброса ATERESET.</p>	1'b0

## 22.3.14 Регистр USBCCIRQCONTROL

Регистр управления запросами на прерывание CC IRQ

Таблица 22.14. Регистр USBCCIRQCONTROL

Bit Number	Name	Description	Reset
------------	------	-------------	-------

31	CC_global_irq_en	Бит разрешения формирования запросов на прерывание от СС логики. 0 - формирование запросов запрещено 1 - формирование запросов разрешено	1'b0
29:20	res2		?
19	CC2_irq_fall_clear	Сброс запроса на прерывание по переключению СС2 из 1 в 0 Сброс осуществляется записью «1» Читается всегда «0»	1'b0
18	CC2_irq_rise_clear	Сброс запроса на прерывание по переключению СС2 из 0 в 1 Сброс осуществляется записью «1» Читается всегда «0»	1'b0
17	CC1_irq_fall_clear	Сброс запроса на прерывание по переключению СС1 из 1 в 0 Сброс осуществляется записью «1» Читается всегда «0»	1'b0
16	CC1_irq_rise_clear	Сброс запроса на прерывание по переключению СС1 из 0 в 1 Сброс осуществляется записью «1» Читается всегда «0»	1'b0
15:12	res1		?
11	CC2_irq_fall_status	Статус запроса на прерывание по переключению СС2 из 1 в 0 0 - нет запроса на прерывание 1 - есть запрос на прерывание RO доступ	1'b0
10	CC2_irq_rise_status	Статус запроса на прерывание по переключению СС2 из 0 в 1 0 - нет запроса на прерывание 1 - есть запрос на прерывание RO доступ	1'b0
9	CC1_irq_fall_status	Статус запроса на прерывание по переключению СС1 из 1 в 0 0 - нет запроса на прерывание 1 - есть запрос на прерывание RO доступ	1'b0
8	CC1_irq_rise_status	Статус запроса на прерывание по переключению СС1 из 0 в 1 0 - нет запроса на прерывание 1 - есть запрос на прерывание RO доступ	1'b0
7:4	res0		?
3	CC2_irq_fall_mask	Маска запроса на прерывание по переключению СС2 из 1 в 0 0 - запрос на прерывание запрещен 1 - запрос на прерывание разрешен	1'b0
2	CC2_irq_rise_mask	Маска запроса на прерывание по переключению СС2 из 0 в 1 0 - запрос на прерывание запрещен 1 - запрос на прерывание разрешен	1'b0
1	CC1_irq_fall_mask	Маска запроса на прерывание по переключению СС1 из 1 в 0 0 - запрос на прерывание запрещен 1 - запрос на прерывание разрешен	1'b0
0	CC1_irq_rise_mask	Маска запроса на прерывание по переключению СС1 из 0 в 1 0 - запрос на прерывание запрещен 1 - запрос на прерывание разрешен	1'b0

## 23. DMA: КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ

### 23.1 Основные особенности

Микросхема содержит два идентичных контроллера прямого доступа в память – DMA0 и DMA1.

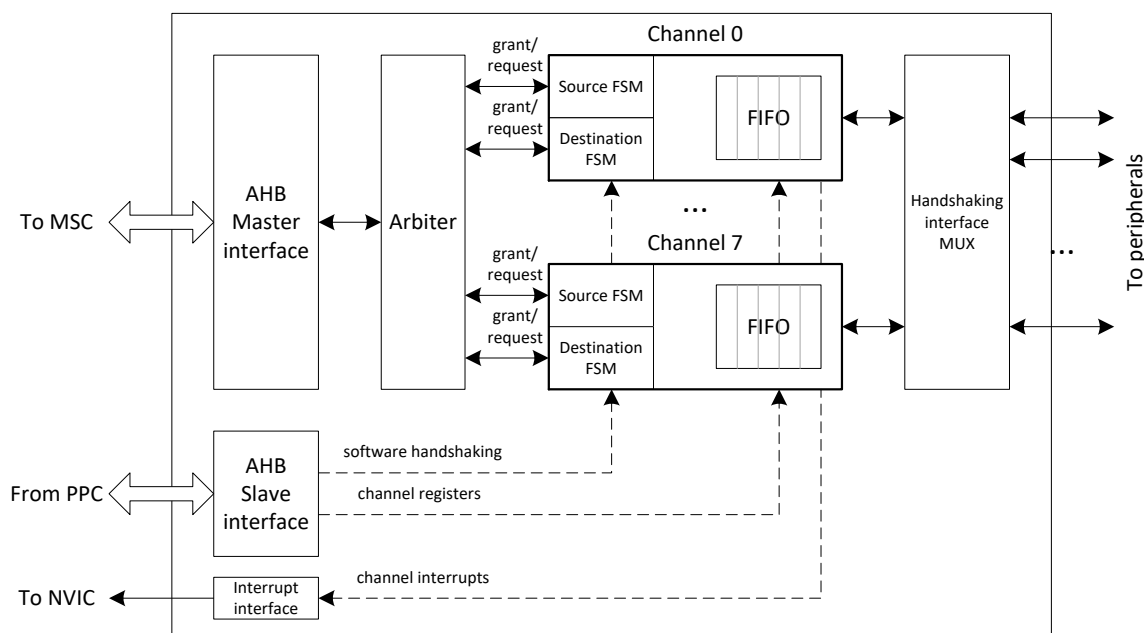


Рисунок 23.1. Блок-схема контроллера DMA

Далее перечислены основные функциональные особенности контроллеров.

#### 23.1.1 Общие

- Один AHB мастер
- Восемь однонаправленных каналов с программируемым приоритетом
- Поддержка пересылок память-память, память-периферия, периферия-память, периферия-периферия
- Component ID для поддержки конфигурируемых программных драйверов
- Размер блока до 4095 слов

#### 23.1.2 Генерация адреса

- Программируемые адреса источника и приёмника (на шине AHB)
- Инкремент, декремент или фиксированный адрес
- Многоблочная передача:
  - Связанные списки (цепочки блоков)
  - Автообновление регистров канала
  - Смежные блоки

- Независимый выбор источника и приёмника для многоблочных передач
- Разброс/сбор (Scatter/Gather)

### 23.1.3 Буферизация каналов

- Одно FIFO на канал для источника и приёмника
- Глубина FIFO: 128 слов (каналы 0, 1); 32 слова (каналы 2 – 7)
- Автоматическая упаковка/распаковка данных в соответствии с шириной FIFO

### 23.1.4 Управление каналом

- Программируемые источник и приёмник в каждом канале
- Программируемый тип передачи в каждом канале (память-память, память-периферия, периферия-память, периферия-периферия)
- Программируемый размер пакетной транзакции для каждого канала
- Программируемое включение/выключение канала
- Поддержка отключения канала без потери данных
- Поддержка приостановки DMA операций
- Поддержка откликов ошибки
- Программируемый максимальный размер пакетной пересылки
- Настраиваемая разрядность пересылки для оптимизации количества вентилей
- Настраиваемый максимальный размер блока для оптимизации количества вентилей
- Блокировка шины – программируется на уровне транзакции, блока, DMA обмена
- Блокировка канала – программируется на уровне транзакции, блока, DMA обмена
- Обратная запись в регистр управления каналом при завершении пересылки каждого блока

### 23.1.5 Инициирование пересылки

- Handshaking интерфейс (интерфейс запросов) для источника и приёмника
  - 16 аппаратных Handshaking интерфейсов
  - Программный Handshaking интерфейс
  - Handshaking Интерфейс прерываний периферии
- Handshaking для одиночных или пакетных транзакций
- Включение/отключение отдельных Handshaking интерфейсов

### 23.1.6 Прерывания

- Объединённый и отдельные запросы прерываний
- Генерация прерываний:
  - Завершение многоблочной пересылки
  - Завершение пересылки блока

- Завершение одиночной и пакетной транзакций
- По признаку ошибки
- Поддержка разрешения и маскирования прерываний

### 23.1.7 Режим пониженного потребления

- Глобальное отключение частоты
- Отключение частоты на уровне канала

### 23.1.8 Пересечение 1К границы

Согласно протоколу АНВ непрерывная пересылка АНВ не должна пересекать 1К границу адресов. DMA автоматически разрешает эту ситуацию. В случае пересылки с пересечением 1К границы DMA автоматически разобьёт её на две, первая из которых закончится на границе 1К, а следующая начнётся после.

## 23.2 Особенности конфигурации и подключения

### 23.2.1 Подключение интерфейсов запросов

Каждый DMA контроллер имеет 16 аппаратных интерфейсов запросов, подключенных к периферийным блокам. В следующей таблице приведено соответствие между интерфейсами и периферийными блоками.

Таблица 23.1.

DMA request number	DMA0 connection	DMA1 connection
0	UART0_TX	UART2_TX
1	UART0_RX	UART2_RX
2	UART1_TX	UART3_TX
3	UART1_RX	UART3_RX
4	SPI0_TX	SPI0_TX
5	SPI0_RX	SPI0_RX
6	SPI1_TX	SPI1_TX
7	SPI1_RX	SPI1_RX
8	SPI2_TX	SPI2_TX
9	SPI2_RX	SPI2_RX
10	I2C0_TX	I2C0_TX
11	I2C0_RX	I2C0_RX
12	I2C1_TX	I2C1_TX
13	I2C1_RX	I2C1_RX
14	QSPI_TX	QSPI_TX
15	QSPI_RX	QSPI_RX

Периферийные устройства, подключенные к интерфейсам #4 – #15, могут быть использованы как с контроллером DMA0, так и с DMA1. Включение интерфейса одного и того же периферийного блока в обоих контроллерах одновременно не допустимо.

## 23.1 Организация пересылок DMA

Настроенный и запущенный в работу *Канал* осуществляет *DMA Пересылку*. *DMA Пересылка* - это наиболее крупная высокоуровневая задача по передаче данных от *Источника* к *Приёмнику*.

*DMA Пересылка* состоит из *Блоков*. *Блок* - блок данных, который необходимо передать от *Источника* к *Приёмнику*. ПО управляет количеством *Блоков* в *DMA Пересылке*. По окончании *DMA Пересылки* *Канал* завершает работу, при необходимости генерируется прерывание. После этого ПО может перенастроить *Канал* на новую *DMA Пересылку*.

Существуют следующие виды *DMA Пересылок*:

1. Одноблочная Пересылка.
2. Пересылка цепочкой Блоков (связным списком).
3. Пересылка с автоповтором.
4. Пересылка смежными блоками.

*Источник* и *Приёмник* можно независимо настроить на необходимый тип *DMA Пересылки*.

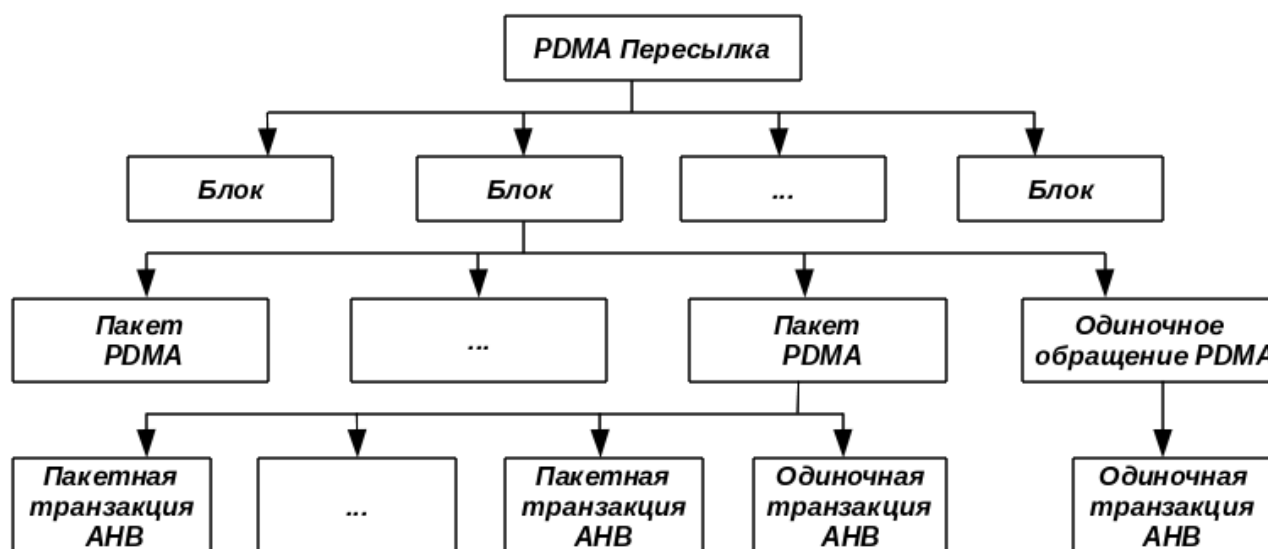
Размер *Блока* задаётся программно, если он известен заранее (размером *Блока* управляет DMA). Размер *Блока* задаётся количеством *Одиночных обращений* на шине АНВ.

Если размер *Блока* заранее неизвестен, тогда размером *Блока* управляет периферийное устройство (*Источник* или *Приёмник*), через *Интерфейс запросов*.

*Замечание:* Все *транзакции АНВ* (тип *транзакции АНВ* определяется сигналом HBURST) состоят из *Одиночных обращений АНВ* (beat АНВ). Например, *транзакция* типа INCR8 состоит из 8-ми *Одиночных обращений*, а *транзакция* типа SINGLE состоит из одного *Одиночного обращения*. *Транзакция* типа SINGLE - *Одиночная транзакция АНВ*. *Транзакции* других типов (INCR, INCR8 и др.) - *Пакетные транзакции АНВ*.

Пересылка *Блока* от *Источника* к DMA, а затем от DMA к *Приёмнику* осуществляется не непрерывным потоком, а *пакетами DMA* и *одиночными обращениями DMA*. *Блок* разбивается на *пакеты DMA* и *одиночные обращения DMA*. *Пакет DMA* — порция данных заданного размера, на которые разбивается *Блок DMA*.

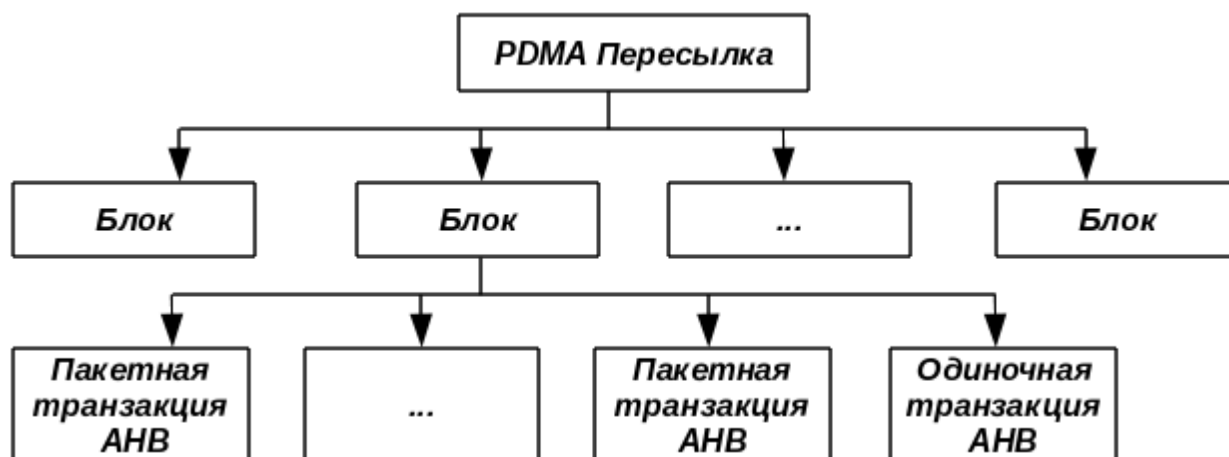
Размер *пакета DMA* (другими словами, количество *одиночных обращений АНВ* внутри *пакета DMA*) настраивается программно. Если *Блок* невозможно разбить на целое количество пакетов, тогда данные, не уместившиеся в пакет, можно переслать *одиночными обращениями DMA*. *Одиночное обращение DMA* соответствует *одиночной транзакции АНВ*. *Пакет DMA* или *Одиночное обращение DMA* - *DMA транзакция*. *Пакет DMA* состоит из *пакетных* и *одиночных транзакций АНВ*.



**Рисунок 23.2. Организация пересылок DMA**

Таким образом, *DMA Пересылка* состоит из одного или нескольких *Блоков*. *Блок* разбивается на *Пакеты DMA* заданного размера. *Пакеты DMA* преобразуются в *транзакции АНВ*.

Для периферийных устройств типа *Память*, структура пересылок выглядит несколько иначе:



**Рисунок 23.3. Организация пересылок DMA для Памяти**

В пересылках между DMA и *Памятью* отсутствует уровень *Пакетов DMA*. Блоки напрямую преобразуются в *транзакции АНВ*, так как подразумевается, что *Память* всегда готова к обмену. При пересылках между *Памятью* и *DMA Интерфейс запросов* не используется.

Для обычного *Источника/Приёмника* инициирование передачи *пакетов DMA* происходит по готовности периферийного устройства (Например, по заполнению FIFO

*Источника/Приёмника* через *Интерфейс запросов* формируется сигнал готовности периферии к обмену).

## 23.2 Интерфейсы запросов

*Интерфейс запросов* используется при пересылке между DMA и периферийным устройством (*Источником/Приёмником*) отличным от *Памяти*. Через *Интерфейс запросов* периферийное устройство сообщает DMA о своей готовности к выдаче/приёму данных. Существуют *Программный* и *Аппаратный Интерфейсы запросов*. Для каждого *Канала* программно настраивается используемый *Интерфейс запросов* (как для *Источника*, так и для *Приёмника*).

### 23.2.1 Область одиночных обращений

В некоторых ситуациях *Блок DMA* невозможно передать, используя только *Пакеты DMA*. Например, размер *Блока DMA* такой, что его невозможно передать за целое количество *Пакетов DMA*. Тогда DMA передаёт *Пакеты* до тех пор, пока количество оставшихся данных не станет меньше размера *Пакета*. В этом случае считается, что периферия переходит в *Область Одиночных Обращений*. Оставшиеся данные *Блока* передаются, используя *Одиночные Обращения DMA*. Замечание: Существует возможность завершить передачи в *Области Одиночных Обращений*, используя *Прерванные Пакеты DMA*. DMA обрабатывает только запросы *Пакетов*, игнорируя запросы *Одиночных Обращений DMA*, если периферия не находится в *Области Одиночных Обращений*.

*Область Одиночных Обращений* определяется только для периферии, которая не управляет размером *Блока*.

### 23.2.2 Прерванный пакет DMA

Запросить *Пакет DMA* возможно, даже если *Источник* или *Приёмник* находятся в *Области Одиночных Обращений*. В этом случае *Пакет DMA* будет запущен, но будет считано только необходимое количество данных. Затем *Пакет* прервётся, не обрабатывая все запрограммированное количество байт в *Пакете* (поля SRC\_MSIZE, DST\_MSIZE, SRC\_TR\_WIDTH, DST\_TR\_WIDTH регистров CTLx).

*Прерванный Пакет DMA* определяется только для периферии, которая не управляет размером *Блока*.

### 23.2.3 Аппаратный интерфейс запросов

Как для *Источника*, так и *Приёмника* каждого *Канала* можно назначить любой из аппаратных *Интерфейсов запросов*.



### 23.2.3.1 Обмен по готовности Источника/Приёмника

При готовности к обмену периферийное устройство устанавливает сигнал запроса. После получения сигнала запроса DMA инициирует передачу на шине АНВ (передачу *пакета DMA* или *одиночного обращения DMA*). После выполнения необходимого обмена контроллер DMA сообщает об этом периферии. Периферийное устройство снимает сигнал запроса.

Для более эффективного использования шины АНВ размер *пакетов DMA* должен соответствовать количеству доступных ресурсов периферии (например, размеру FIFO периферии, по заполнению которого, периферия формирует сигнал готовности к обмену).

### 23.2.3.2 Управление размером Блока

Если размером *Блока* управляет периферийное устройство (т.е. размер *Блока* заранее не известен), то при выдаче/приёме необходимого количества данных, периферийное устройство формирует признак окончания *Блока*. После чего DMA прекращает дальнейшие обращения к периферийному устройству, завершает обработку *Блока*, информирует об этом периферийное устройство.

Если размером *Блока* управляет DMA (т.е. размер *Блока* известен заранее и задаётся через регистры DMA), тогда сигнал завершения *Блока* от периферии не используется.

### 23.2.4 Программный Интерфейс запросов

При использовании *программного Интерфейса запросов* определение готовности периферии к обмену, а также инициирование передач производится программно. Например, периферийное устройство может сформировать сигнал прерывания, а обработчик прерывания, считав состояния периферии, программно запустить необходимые передачи.

Управление передачами осуществляется через следующие регистры:

- ReqSrcReg - Запрос передачи *пакета DMA Источника*;
- SglReqSrcReg - Запрос одиночного обращения DMA к *Источнику*;
- LstSrcReg - Признак последней передачи (*пакета DMA* или *одиночного обращения DMA*) *Источника*;
- ReqDstReg, SglReqDstReg, LstDstReg - регистры *Приёмника*, аналогичные регистрам ReqSrcReg, SglReqSrcReg, LstSrcReg *Источника*.

#### 23.2.4.1 Программный Интерфейс запросов, периферия не управляет размером Блока

Регистры LstSrcReq и LstDstReq в этом режиме не используются и их значения игнорируются.

#### 23.2.4.2 Источник/Приёмник находится вне Области Одиночных Обращений

Для того, чтобы инициировать пересылку *Пакета Источника* для *Канала N* необходимо установить соответствующие *Каналу* разряды запросов в регистрах ReqSrcReq и SglReqSrcReq. Т.е для запуска *Пакета* необходимо выполнение следующих условий:

$\text{ReqSrcReq}[N] = 1$  (*Запрос передачи Пакета*)

$\text{SglReqSrcReq}[N] = 1$  (*Запрос Одиночного Обращения*)

Последовательность обращений к регистрам ReqSrcReq и SglReqSrcReq вне *Области Одиночных Обращений* не важна. После выполнения передачи *Пакета*, установленные ранее разряды запросов в регистрах ReqSrcReq и SglReqSrcReq сбросятся аппаратно. Для *Приёмника* все работает аналогично (вместо регистров ReqSrcReq и SglReqSrcReq используются регистры ReqDstReq и SglReqDstReq).

#### 23.2.4.3 Источник/Приёмник находится в Области Одиночных Обращений

Для того, чтобы инициировать пересылку *Одиночного Обращения Источника* для *Канала N* необходимо установить соответствующий *Каналу* разряд запроса в регистре SglReqSrcReq:

$\text{SglReqSrcReq}[N] = 1$  (*Запрос Одиночного Обращения*)

После этого запись запроса в  $\text{ReqSrcReq}[N]$  будет проигнорирована. Разряды запросов в регистрах ReqSrcReq и SglReqSrcReq будут сброшены аппаратно после завершения *Одиночного Обращения DMA*.

Для того, чтобы запустить *Пакет DMA* в *Области Одиночных Обращений* необходимо сначала установить разряд запроса в регистре ReqSrcReq, а затем в регистре SglReqSrcReq:

$\text{ReqSrcReq}[N] = 1$  (*Запрос передачи Пакета*)

$\text{SglReqSrcReq}[N] = 1$  (*Запрос Одиночного Обращения*)

Запущенный *Пакет* будет обработан как *Прерванный Пакет DMA*. Разряды запросов в регистрах ReqSrcReq и SglReqSrcReq будут сброшены аппаратно после завершения *Прерванного Пакета DMA*.

Для *Приёмника* все работает аналогично (вместо регистров ReqSrcReq и SglReqSrcReq используются регистры ReqDstReq и SglReqDstReq).

Таким образом, ПО может определять моменты завершения выполнения *Пакетов и Одиночных обращений* или опрашивая регистры ReqSrcReq/ReqDstReq, SglReqSrcReq/SglReqDstReq, или настроив прерывания IntSrcTran, IntDstTran. Прерывания генерируются по выполнению *транзакции DMA* (как по завершению *Одиночного Обращения DMA*, так и *Пакета DMA*).

#### 23.2.4.4 Программный Интерфейс запросов, периферия управляет размером Блока

Регистры ReqSrcReq/ReqDstReq, SglReqSrcReq/SglReqDstReq используются несколько иначе, чем случае, когда размером *Блока* управляет не периферия. Сначала при необходимости устанавливается LstSrcReg / LstDstReg или SglReqSrcReq / SglDstSrcReq. После этого записью в ReqSrcReq / ReqDstReq инициируется передача. Тип передачи определяется заранее предустановленными SglReqSrcReq / SglDstSrcReq (Запрос Одиночного Обращения) и LstSrcReg/ LstDstReg (Признак последней передачи в *Блоке*). Порядок записи в регистры LstSrcReg/ LstDstReg и SglReqSrcReq/SglDstSrcReq не важен. Запись в ReqSrcReq/ ReqDstReq всегда должна следовать после записей в SglReqSrcReq/SglDstSrcReq и LstSrcReg/ LstDstReg. После завершения *транзакции DMA* (*Пакета или Одиночного обращения*) разряды запросов в регистрах *Интерфейса запроса* аппаратно сбрасываются. Если периферия управляет размером *Блока*, и блок не умещается в целое количество *Пакетов Источника* или *Приёмника*, ПО должно использовать *Одиночные Обращения*, чтобы завершить *Блок*.

### 23.3 Виды DMA Пересылок

#### 23.3.1 Одноблочная Пересылка

В случае одноблочной пересылки после выполнения передачи *Блока Канал* останавливает свою работу.

#### 23.3.2 Пересылка цепочкой Блоков (связным списком)

Если для *Источника* или *Приёмника* настроена передача цепочкой *Блоков*, тогда после завершения передачи *Блока* следующий *Блок* автоматически проинициализируется из *Описателя Блока* (LLI - linked list Item), адрес которого задаётся в регистре LLPx.

*Описатель Блока* (он же элемент связного списка или LLI) представляет из себя следующую область памяти:

Таблица 23.2. Структура Описателя Блока (LLI)

Адрес	Название	Описание
{LLPx[31:2], 2'b00} + 0x18	LLI.DSTAT	Статус <i>Приёмника</i> . Сюда после завершения <i>Блока</i> DMA может записать значение, считанное по адресу указанному в регистре DSTATARx.
{LLPx[31:2], 2'b00} + 0x14	LLI.SSTAT	Статус <i>Источника</i> . Сюда после завершения <i>Блока</i> DMA может записать значение, считанное по адресу указанному в регистре SSTATAR.
{LLPx[31:2], 2'b00} + 0x10	LLI.CTL[63:32]	Управляющий регистр <i>Блока</i>
{LLPx[31:2], 2'b00} + 0xC	LLI.CTL[31:0]	Управляющий регистр <i>Блока</i>
{LLPx[31:2], 2'b00} + 0x8	LLI.LLP	Указатель на следующий <i>Описатель Блока</i>
{LLPx[31:2], 2'b00} + 0x4	LLI.DAR	Начальный адрес <i>Приёмника</i>
{LLPx[31:2], 2'b00}	LLI.SAR	Начальный адрес <i>Источника</i>

Из этой области инициализируются соответствующие регистры *Канала* при запуске *Блока*.

### 23.3.2.1 Пересылка с автоповтором

Если для *Источника* или *Приёмника* настроена *Пересылка с автоповтором*, тогда содержимое регистров *Канала*, после завершения передачи *Блока* будет обновляться значениями, которые содержались в них в момент запуска *Канала*.

### 23.3.2.2 Пересылка смежными Блоками

После выполнения текущего *Блока*, начало следующего *Блока* является продолжением текущего.

Возможны следующие варианты многоблочных DMA Пересылок:

- автоповтор Блока *Источника* - Автоповтор Блока *Приёмника*;
- автоповтор Блока *Источника* - Цепочка Блоков *Приёмника*;
- автоповтор Блока *Источника* - Смежные Блоки *Приёмника*;
- цепочка Блоков *Источника* - Автоповтор Блока *Приёмника*;
- цепочка Блоков *Источника* - Цепочка Блоков *Приёмника*;
- цепочка Блоков *Источника* - Смежные Блоки *Приёмника*;
- смежные Блоки *Источника* - Автоповтор Блока *Приёмника*;
- смежные Блоки *Источника* - Цепочка Блоков *Приёмника*.

DMA не поддерживает многоблочную передачу типа *Смежные Блоки Источника* - *Смежные Блоки Приёмника*. Если необходима такая функциональность, следует либо

использовать *Блоки* большего размера, либо использовать пересылки *Цепочками Блоков* для *Источника* и *Приёмника* с необходимыми настройками адресов SARx и DARx.

Тип *Пересылки* определяется полями регистров LLPx, CTLx, CFGx. Содержимое регистров SARx, DARx, CTLx, LLPx *Канала* может автоматически обновляться в начале *Блока* в зависимости от типа *Пересылки*. В следующей таблице приведены возможные типы *Пересылок* и варианты изменений регистров *Канала*.

**Таблица 23.3. Типы пересылок и обновление регистров Канала**

Т и п	LO C = 0 LL Px.	CTLx.LLP_S RC_EN	CFGx.RELO AD_SRC	CTLx.LLP_ DST_EN	CFGx.RELOA D_DST	Обновл ение CTLx и LLPx	Обнов ление SARx	Обнов ление DARx	Обратн ая запись
Одноблочная <i>Пересылка</i> или последний <i>Блок</i> многоблочной									
1	Да	0	0	0	0	Нет	Нет	Нет	Нет
Многоблочные <i>Пересылки</i> без использования цепочек <i>Блоков</i>									
2	Да	0	0	0	1	Нач.	Смеж.	Нач.	Нет
3	Да	0	1	0	0	Нач.	Нач.	Смеж.	Нет
4	Да	0	1	0	1	Нач.	Нач.	Нач.	Нет
Одноблочная <i>Пересылка</i> или последний <i>Блок</i> многоблочной									
5	Нет	0	0	0	0	Нет	Нет	Нет	Да
Многоблочные <i>Пересылки</i> цепочками <i>Блоков</i>									
6	Нет	0	0	1	0	LLI	Смеж.	LLI	Да
7	Нет	0	1	1	0	LLI	Нач.	LLI	Да
8	Нет	1	0	0	0	LLI	LLI	Смеж.	Да
9	Нет	1	0	0	1	LLI	LLI	Нач.	Да
10	Нет	1	0	1	0	LLI	LLI	LLI	Да

В таблице используются следующие условные обозначения и сокращения:

*Нач.* - начальное значение. При старте *Блока* с автоповтором, в регистры автоматически записывается значение, которое было записано в них до запуска *Блока*.

*Смеж.* - смежные адреса *Блоков*.

*LLI* - значение регистра загружается и LLI (из Описателя Блока). Используется при обработке *Блока* в цепочке *Блоков*.

*Обратная запись (write back)* - Под обратной записью подразумевается запись управляющей и статусной информации контроллером DMA в *Описатель Блока* при завершении обработки *Блока*. DMA может обновлять поля DSTAT, SSTAT, CTL LLI по завершению обработки *Блока*.

## 23.4 Изменение адреса внутри *Блока*

При выполнении пересылки *Блока* существует несколько возможностей изменения адреса *Источника/Приёмника* внутри *Блока*. Способ изменения адреса *Источника/Приёмника* внутри *Блока* настраивается независимо как для *Источника*, так и для *Приёмника*.

### 23.4.1 Адрес не меняется

Все обращения DMA к *Источнику/Приёмнику* производятся по одному и тому же адресу. Например, такой способ может подойти, если в качестве буфера данных *Источника/Приёмника* используется FIFO.

### 23.4.2 Адрес увеличивается

В этом случае все обращения DMA к *Источнику/Приёмнику* производятся по последовательным адресам с возрастанием адреса.

### 23.4.3 Адрес уменьшается

Обращения DMA к *Источнику/Приёмнику* производятся по последовательным адресам с убыванием адреса.

### 23.4.4 Пересылка с Разбросом/Сбором

Для пересылок с увеличением или уменьшением адреса можно настроить дополнительные возможности *Сбора*(gather) и *Разброса*(scatter) данных.

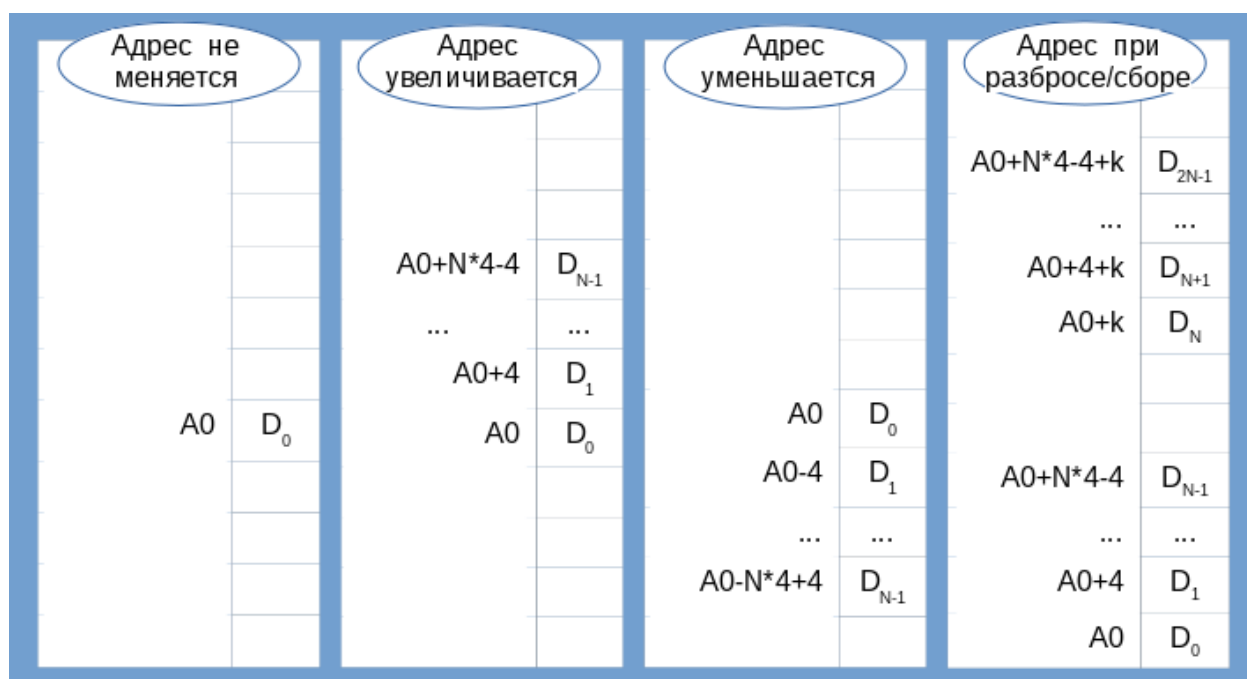
Возможность *разброса* данных относится только к *Приёмнику*.

Для *Блока*, с включённым *Разбросом* данных, программно определяется количество *Одиночных обращений АНВ* к *Приёмнику*, после которых адрес обращения увеличивается/уменьшается на заданную величину.

*Сбор* данных относится только к *Источнику*.

Для *Блока*, с включённым *Сбором* данных, программно определяется количество *Одиночных обращений АНВ* к *Источнику*, после которых адрес обращения увеличивается/уменьшается на заданную величину.

*Граница Разброса/Сбора* — адрес, после которого происходит скачкообразное изменение адреса *Приёмника/Источника*. Граница достигается после заданного количества обращений со стороны DMA (см. SGR.SGI и DSR.DSI).



**Рисунок 23.4. Пример изменения адреса в Блоке для 32-х разрядных пересылок**

На Рисунок 23.5 показан пример изменения адреса *Источника* при включённой возможности *Сбора*:

Адрес	Данные	
A + 68	D <sub>11</sub>	
A + 64	D <sub>10</sub>	
A + 60	D <sub>9</sub>	
A + 56	D <sub>8</sub>	
		Разрядность одиночного обращения АНВ Источника: CTLx.SRC_TR_WIDTH = 3'b010 (32разряда/8 = 4 байта)
		Интервал Сбора Источника: SGR.SGI = 3
A + 40	D <sub>7</sub>	Величина изменения адреса при достижении границы Сбора = количество байт одиночного обращения * SGR.SGI = 12 байт
A + 36	D <sub>6</sub>	
A + 32	D <sub>5</sub>	
A + 28	D <sub>4</sub>	Количество обращений PDMA к Источнику с непрерывным адре- сом (до достижения границы Сбора): SGR.SGC = 4
A + 12	D <sub>3</sub>	
A + 8	D <sub>2</sub>	
A + 4	D <sub>1</sub>	
A	D <sub>0</sub>	

**Рисунок 23.5. Пример изменения адреса для пересылки со Сбором**

В примере DMA последовательно считывает из *Источника* 4 слова (задаётся в SGR.SGC). Затем адрес обращения изменяется скачкообразно (задаётся в SGR.SGI), и следующие 4 слова считываются не из последовательного адреса, а из этого нового адреса и т.д.

При включённом *Разбросе* в *Приёмнике* обращения будут происходить аналогично.

## 23.5 Выключение канала до завершения пересылки

При обычном порядке работы *Канал* включается программно (записью единицы в ChEnReg.CH\_EN) и аппаратно отключается после выполнения пересылки (ChEnReg.CH\_EN сбрасывается аппаратно).

Чтобы избежать потери данных при необходимости программной остановки *Канала* до завершения пересылки, рекомендуется использовать следующую последовательность действий:



- установить разряд CFGx.CH\_SUSP (запрос прекращения пересылок от Источника, после этого FIFO канала не будет получать новых данных);
- опрашивать разряд CFGx.FIFO\_EMPTY до тех пор, пока FIFO не опустошится;
- программно сбросить разряд ChEnReg.CH\_EN.

Если разрядность передач от *Источника*(CTLx.SRC\_TR\_WIDTH) меньше разрядности передач к *Приёмнику*(CTLx.DST\_TR\_WIDTH) и был установлен разряд CFGx.CH\_SUSP при непустом FIFO, может возникнуть ситуация, когда данные считанные из *Источника* не будут переданы *Приёмнику* (считано недостаточно разрядов для одиночного обращения к *Приёмнику*). Разрешается возобновить работу канала (записью нуля в CFGx.CH\_SUSP).

*Замечание:* При программной остановке *Канала* не гарантируется подтверждение на запрос одиночного или пакетного обращения.

Канал также может быть остановлен без предварительных действий просто записью нуля в ChEnReg.CH\_EN. Запись в CH\_EN следует рассматривать как запрос на прекращение работы. Чтобы убедиться, что канал действительно остановлен, следует опрашивать ChEnReg.CH\_EN до тех пор, пока из него не будет считан ноль.

Возможно, прервать работу всех каналов сразу сбросом глобального разрешения работы DMA DmaCfgReg[0]. Для того, чтобы убедиться, что каналы остановлены, необходимо опрашивать регистр ChEnReg.

*Замечание:* Если разряд разрешения работы сброшен при непустом FIFO, данные из FIFO *Приёмнику* не передаются. Это может привести к потере данных. При программной остановке *Канала* не гарантируется подтверждение на запрос одиночного или пакетного обращения.

## 23.6 Прерывания

Для каждого *Канала* существует пять источников прерываний:

- **IntTfr** - Завершение *DMA Пересылки*.

Прерывание формируется после завершения всей *DMA Пересылки Приёмнику*.

- **IntBlock** - Завершение передачи *Блока*.

Прерывание формируется после передачи всего *Блока Приёмнику*.

- **IntDstTran** - Завершение передачи *Приёмнику*.

Формируется после выполнения последней АНВ транзакции в *Пакете DMA* или *Одиночном обращении DMA Приёмника*. *Пакет* или *Одиночное обращение DMA* иницируются через *Интерфейс запросов* (программный или аппаратный). Если *Приёмник* является

устройством типа *Память*, тогда это прерывание не будет формироваться, так как для *Памяти* не существует разбиения *Блока* на *Пакеты* и *Одиночные обращения DMA*.

- **IntSrcTran** - Завершение передачи *Источника*.

Формируется после выполнения последней АНВ транзакции в *Пакете DMA* или *Одиночном обращении DMA Источника*. *Пакет* или *Одиночное обращение DMA* иницируются через *Интерфейс запросов* (программный или аппаратный). Если *Источник* является устройством типа *Память*, тогда это прерывание не будет формироваться, так как для *Памяти* не существует разбиения *Блока* на *Пакеты* и *Одиночные обращения DMA*.

- **IntError** - Ошибка передачи АНВ.

Это прерывание формируется, если во время выполнения передачи через ведущий порт АНВ будет получен ответ ERROR (см. сигнал HRESP АНВ). Дополнительно происходит отмена *DMA Пересылки* и отключение *Канала*.

Для работы с прерываниями используются следующие группы регистров:

- немаскируемые регистры прерываний.

RawBlock, RawDstTran, RawErr, RawSrcTran, RawTfr. Регистры содержат информацию о выполнении условий формирования прерываний для каждого канала. Всегда устанавливаются при выполнении условия прерывания (независимо от CTLx.INT\_EN).

- регистры масок прерываний.

MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, MaskTfr.

- регистры масок прерываний каналов для всех типов прерываний;
- маскируемые регистры прерываний.

StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr.

Содержимое регистров Raw\* маскируется регистрами Mask\* и записывается в регистры Status\*. Для того, чтобы прерывания могли формироваться в регистрах Status\*, необходимо также глобально разрешить формирование сигналов всех прерываний через CTLx.INT\_EN.

- регистры сброса прерываний.

ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr.

Запись в регистры Clear\* очищает запросы прерываний в регистрах Raw\* и Status\* на одном и том же такте.

- объединённый регистр прерываний StatusInt.

Каждый из пяти значащих разрядов регистра StatusInt содержит взятое по ИЛИ содержимое соответствующего регистра StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr. Таким образом, регистр StatusInt указывает на наличие прерываний определённого типа. Для того, чтобы определить конкретный *Канал* прерывания, необходимо считать значения регистров StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr.

Физические выводы сигналов прерываний:

- IntTfr - Завершение *DMA Пересылки* в любом *Канале*;
- IntBlock - Завершение передачи *Блока* в любом *Канале*;
- IntDstTran - Завершение передачи *Приёмнику* в любом *Канале*;
- IntSrcTran - Завершение передачи *Источника* в любом *Канале*;
- IntErr - Ошибка передачи АНВ в любом *Канале*;
- IntCombined - взятые по ИЛИ IntTfr, IntBlock, IntDstTran, IntSrcTran, IntErr;
- запросы прерываний на физических выводах формируются, если прерывания разрешены через CTLx.INT\_EN и установлены соответствующие маски в регистрах Mask\*.

## 23.7 Регистры

Все регистры АНВ DMA 64-х разрядные. В основном используются только младшие 32 разряда. Записи в зарезервированные разряды регистров игнорируются. При чтении в зарезервированных разрядах возвращаются нули.

### 23.7.1 Ошибки при доступе к регистрам

Следующие обращения к регистрам DMA считаются запрещёнными и будут вызывать отклик на шине АНВ с установленным признаком ошибки:

- попытка обращения по АНВ с hsize более 64;
- попытка обращения по несуществующему адресу при установленном hsel;
- запись в регистры SARx, DARx, LLPx, CTLx, SSTATx, DSTATx, SSTATARx, DSTATARx, SGRx, DSRx при включенном канале;
- попытка чтения из регистров ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr (доступны только по записи);
- попытка записи в регистры StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr, StatusInt, DmaIdReg, DMA\_Component\_ID\_Register (доступны только по чтению).

## 23.7.2 Карта памяти

Таблица 23.4. Список регистров DMA

Сокращённое название	Смещение	Тип	Название
<b>Регистры Канала 0</b>			
SAR0	0x000	RW	Адрес Источника
DAR0	0x008	RW	Адрес Приёмника
LLP0	0x010	RW	Адрес описателя следующего Блока
CTL0	0x018	RW	Регистр управления
SSTAT0	0x020	RW	Статус Источника
DSTAT0	0x028	RW	Статус Приёмника
SSTATAR0	0x030	RW	Адрес Статуса Источника
DSTATAR0	0x038	RW	Адрес Статуса Приёмника
CFG0	0x040	RW	Регистр конфигурации
SGR0	0x048	RW	Регистр Сбора Источника
DSR0	0x050	RW	Регистр Разброса Приёмника
<b>Регистры Канала 1</b>			
SAR1	0x058	RW	См. регистры Канала 0
DAR1	0x060	RW	См. регистры Канала 0
LLP1	0x068	RW	См. регистры Канала 0
CTL1	0x070	RW	См. регистры Канала 0
SSTAT1	0x078	RW	См. регистры Канала 0
DSTAT1	0x080	RW	См. регистры Канала 0
SSTATAR1	0x088	RW	См. регистры Канала 0
DSTATAR1	0x090	RW	См. регистры Канала 0
CFG1	0x098	RW	См. регистры Канала 0
SGR1	0x0A0	RW	См. регистры Канала 0
DSR1	0x0A8	RW	См. регистры Канала 0
<b>Регистры Канала 2</b>			
SAR2	0x0B0	RW	См. регистры Канала 0
DAR2	0x0B8	RW	См. регистры Канала 0
LLP2	0x0C0	RW	См. регистры Канала 0
CTL2	0x0C8	RW	См. регистры Канала 0
SSTAT2	0x0D0	RW	См. регистры Канала 0
DSTAT2	0x0D8	RW	См. регистры Канала 0
SSTATAR2	0x0E0	RW	См. регистры Канала 0
DSTATAR2	0x0E8	RW	См. регистры Канала 0
CFG2	0x0F0	RW	См. регистры Канала 0
SGR2	0x0F8	RW	См. регистры Канала 0
DSR2	0x100	RW	См. регистры Канала 0
<b>Регистры Канала 3</b>			
SAR3	0x108	RW	См. регистры Канала 0
DAR3	0x110	RW	См. регистры Канала 0
LLP3	0x118	RW	См. регистры Канала 0
CTL3	0x120	RW	См. регистры Канала 0
SSTAT3	0x128	RW	См. регистры Канала 0
DSTAT3	0x130	RW	См. регистры Канала 0
SSTATAR3	0x138	RW	См. регистры Канала 0
DSTATAR3	0x140	RW	См. регистры Канала 0
CFG3	0x148	RW	См. регистры Канала 0
SGR3	0x150	RW	См. регистры Канала 0
DSR3	0x158	RW	См. регистры Канала 0
<b>Регистры Канала 4</b>			
SAR4	0x160	RW	См. регистры Канала 0
DAR4	0x168	RW	См. регистры Канала 0
LLP4	0x170	RW	См. регистры Канала 0
CTL4	0x178	RW	См. регистры Канала 0

Сокращённое название	Смещение	Тип	Название
SSTAT4	0x180	RW	См. регистры Канала 0
DSTAT4	0x188	RW	См. регистры Канала 0
SSTATAR4	0x190	RW	См. регистры Канала 0
DSTATAR4	0x198	RW	См. регистры Канала 0
CFG4	0x1A0	RW	См. регистры Канала 0
SGR4	0x1A8	RW	См. регистры Канала 0
DSR4	0x1B0	RW	См. регистры Канала 0
<b>Регистры Канала 5</b>			
SAR5	0x1B8	RW	См. регистры Канала 0
DAR5	0x1C0	RW	См. регистры Канала 0
LLP5	0x1C8	RW	См. регистры Канала 0
CTL5	0x1D0	RW	См. регистры Канала 0
SSTAT5	0x1D8	RW	См. регистры Канала 0
DSTAT5	0x1E0	RW	См. регистры Канала 0
SSTATAR5	0x1E8	RW	См. регистры Канала 0
DSTATAR5	0x1F0	RW	См. регистры Канала 0
CFG5	0x1F8	RW	См. регистры Канала 0
SGR5	0x200	RW	См. регистры Канала 0
DSR5	0x208	RW	См. регистры Канала 0
<b>Регистры Канала 6</b>			
SAR6	0x210	RW	См. регистры Канала 0
DAR6	0x218	RW	См. регистры Канала 0
LLP6	0x220	RW	См. регистры Канала 0
CTL6	0x228	RW	См. регистры Канала 0
SSTAT6	0x230	RW	См. регистры Канала 0
DSTAT6	0x238	RW	См. регистры Канала 0
SSTATAR6	0x240	RW	См. регистры Канала 0
DSTATAR6	0x248	RW	См. регистры Канала 0
CFG6	0x250	RW	См. регистры Канала 0
SGR6	0x258	RW	См. регистры Канала 0
DSR6	0x260	RW	См. регистры Канала 0
<b>Регистры Канала 7</b>			
SAR7	0x268	RW	См. регистры Канала 0
DAR7	0x270	RW	См. регистры Канала 0
LLP7	0x278	RW	См. регистры Канала 0
CTL7	0x280	RW	См. регистры Канала 0
SSTAT7	0x288	RW	См. регистры Канала 0
DSTAT7	0x290	RW	См. регистры Канала 0
SSTATAR7	0x298	RW	См. регистры Канала 0
DSTATAR7	0x2A0	RW	См. регистры Канала 0
CFG7	0x2A8	RW	См. регистры Канала 0
SGR7	0x2B0	RW	См. регистры Канала 0
DSR7	0x2B8	RW	См. регистры Канала 0
<b>Регистры прерываний</b>			
RawTfr	0x2C0	R	Немаскируемые прерывания IntTfr
RawBlock	0x2C8	R	Немаскируемые прерывания IntBlock
RawSrcTran	0x2D0	R	Немаскируемые прерывания IntSrcTran
RawDstTrn	0x2D8	R	Немаскируемые прерывания IntDstTran
RawErr	0x2E0	R	Немаскируемые прерывания IntErr
StatusTfr	0x2E8	R	Статус прерывания IntTfr
StatusBlock	0x2F0	R	Статус прерываний IntBlock
StatusSrcTran	0x2F8	R	Статус прерываний IntSrcTran
StatusDstTran	0x300	R	Статус прерываний IntDstTran
StatusErr	0x308	R	Статус прерываний IntErr
MaskTfr	0x310	RW	Маска прерываний IntTfr
MaskBlock	0x318	RW	Маска прерываний IntBlock
MaskSrcTran	0x320	RW	Маска прерываний IntSrcTran

Сокращённое название	Смещение	Тип	Название
MaskDstTran	0x328	RW	Маска прерываний IntDstTran
MaskErr	0x330	RW	Маска прерываний IntErr
ClearTfr	0x338	W	Сброс прерываний IntTfr
ClearBlock	0x340	W	Сброс прерываний IntBlock
ClearSrcTran	0x348	W	Сброс прерываний IntSrcTran
ClearDstTran	0x350	W	Сброс прерываний IntDstTran
ClearErr	0x358	W	Сброс прерываний IntErr
StatusInt	0x360	W	Объединённый регистр прерываний
<b>Регистры программного Интерфейса Запросов</b>			
ReqSrcReg	0x368	RW	Запрос передачи пакета Источника
ReqDstReg	0x370	RW	Запрос передачи пакета Приёмника
SglReqSrcReg	0x378	RW	Запрос одиночного обращения Источника
SglReqDstReg	0x380	RW	Запрос одиночного обращения Приёмника
LstSrcReg	0x388	RW	Признак последнего обращения Источника
LstDstReg	0x390	RW	Признак последнего обращения Приёмника
<b>Регистры конфигурации и управления DMA</b>			
DmaCfgReg	0x398	RW	Конфигурационный регистр DMA
ChEnReg	0x3a0	RW	Регистр включения каналов DMA
DmaIdReg	0x3a8	R	ID регистр DMA
DmaTestReg	0x3b0	RW	Тестовый регистр DMA
<b>Регистры аппаратных параметров DMA</b>			
Reserved	0x3b8		Зарезервировано
Reserved	0x3c0		Зарезервировано
DMA_COMP_PARAMS_6	0x3c8	R	6-ой регистр параметров DMA
DMA_COMP_PARAMS_5	0x3d0	R	5-ый регистр параметров DMA
DMA_COMP_PARAMS_4	0x3d8	R	4-ый регистр параметров DMA
DMA_COMP_PARAMS_3	0x3e0	R	3-ий регистр параметров DMA
DMA_COMP_PARAMS_2	0x3e8	R	2-ой регистр параметров DMA
DMA_COMP_PARAMS_1	0x3f0	R	1-ый регистр параметров DMA
Dma Component ID	0x3f8	R	ID регистр компонента DMA

### 23.7.3 Регистры Каналов

Каждый из восьми *Каналов* DMA имеет следующий набор регистров:

SAR<sub>x</sub>, DAR<sub>x</sub>, LLP<sub>x</sub>, CTL<sub>x</sub>, SSTAT<sub>x</sub>, DSTAT<sub>x</sub>, SSTATAR<sub>x</sub>, DSTATAR<sub>x</sub>, CFG<sub>x</sub>, SGR<sub>x</sub>, DSR<sub>x</sub> (x от 0 до 7).

#### 23.7.3.1 SAR<sub>x</sub> - Адрес Источника

Регистр SAR<sub>x</sub> (Source address register for channel x) -

Регистр адреса *Источника* для *Канала* x.

Смещение	SAR0 – 0x000 SAR1 – 0x058 SAR2 – 0x0b0 SAR3 – 0x108 SAR4 – 0x160 SAR5 – 0x1b8 SAR6 – 0x210 SAR7 – 0x268
Тип	RW
Состояние по сбросу	0x0

Смещение	SAR0 – 0x000 SAR1 – 0x058 SAR2 – 0x0b0 SAR3 – 0x108 SAR4 – 0x160 SAR5 – 0x1b8 SAR6 – 0x210 SAR7 – 0x268
Разрядность	64

Начальное значение регистра SARx задаётся или программно до включения *Канала*, или считывается из *Описателя Блока (LLI)* перед началом обработки *Блока*. Во время обработки *DMA Пересылки* регистр содержит текущее значение адреса, по которому DMA производит АНВ чтения из *Источника*.

Подробнее о возможностях загрузки начального значения SARx при обработке *Блока* см. в таблице «Типы пересылок и обновление регистров Канала» в разделе «Виды DMA Пересылок».

**Таблица 23.5. Описание полей регистра SARx**

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	SAR	Текущий адрес <i>Источника</i> данных. Обновляется после каждого считывания данных контроллером DMA из <i>Источника</i> . Поле SINC регистра CTLx определяет увеличивается ли адрес, уменьшается или остаётся неизменным после каждой передачи внутри <i>Блока</i> . Адрес должен быть выровнен согласно разрядности обращений по шине АНВ для <i>Источника</i> (определяется полем CTLx.SRC_TR_WIDTH).

### 23.7.3.2 DARx - Адрес Приёмника

Регистр DARx (Destination address register for channel x) — Регистр адреса *Приёмника* для *Канала x*.

Смещение	DAR0 – 0x008 DAR1 – 0x060 DAR2 – 0x0b8 DAR3 – 0x110 DAR4 – 0x168 DAR5 – 0x1c0 DAR6 – 0x218 DAR7 – 0x270
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Начальное значение регистра DARx задаётся или программно до включения *Канала*, или считывается из *Описателя Блока (LLI)* перед началом обработки *Блока*. Во время обработки *DMA Пересылки* регистр содержит текущее значение адреса, по которому DMA производит АНВ записи в *Приёмник*.



Подробнее о возможностях загрузки начального значения DARx при обработке *Блока* см. в таблице «Типы пересылок и обновление регистров Канала» в разделе «Виды DMA Пересылок».

**Таблица 23.6. Описание полей регистра DARx**

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	DAR	Текущий адрес <i>Приёмника</i> данных. Обновляется после каждой АНВ записи данных контроллером DMA в <i>Приёмник</i> . Поле DINC регистра CTLx определяет увеличивается ли адрес, уменьшается или остаётся неизменным после каждой передачи внутри <i>Блока</i> . Адрес должен быть выровнен согласно разрядности обращений по шине АНВ для <i>Приёмника</i> (определяется полем CTLx.DST_TR_WIDTH).

Автоматическое выравнивание адреса Приёмника

В определённых ситуациях при *Пересылке Смежными Блоками* после окончания передачи *Блока* адрес *Приёмника* может оказаться невыровненным для передачи следующего *Блока*. В таких ситуациях DMA автоматически выравнивает адрес *Приёмника*, перед стартом следующего *Блока*.

*Пример.*

- размер *Блока* - 9. Настраивается в CTLx.BLOCK\_TS;
- разрядность передач *Источника* - 16 (полуслово). Настраивается в CTLx.SRC\_TR\_WIDTH;
- разрядность передач *Приёмника* - 32 (слово). Настраивается в CTLx.DST\_TR\_WIDTH;
- для *Приёмника* настроена пересылка *Смежными Блоками*.

DMA выполнит 4 записи по целому слову в *Приёмник*, а затем запись полуслова для того, чтобы завершить передачу *Блока*. В конце передачи *Блока* адрес *Приёмника* будет выровнен к 16-битной границе, так как последней пересылкой на АНВ была пересылка полуслова. Это не совпадает с заданной разрядностью обращений *Приёмника* (32 разряда). В этом случае DMA выравнивает начальный адрес следующего *Блока Приёмника* к ближайшему 32 разрядному адресу (следующему 32 разрядному адресу, если адрес настроен на увеличение, предыдущему 32-разрядному адресу, если адрес настроен на уменьшение).

DMA автоматически выравнивает адрес *Приёмника* при одновременном выполнении следующих условий:

- приёмник настроен на работу со *Смежными Блоками*;
- разрядность обращений АНВ к Приёмнику больше разрядности обращений к Источнику (DST\_TR\_WIDTH > SRC\_TR\_WIDTH);



- невозможно обработать Блок за целочисленное количество обращений разрядности DST\_TR\_WIDTH к Приёмнику ((BLOCK\_TS\*SRC\_TR\_WIDTH)/DST\_TR\_WIDTH != целое\_число).

### 23.7.3.3 LLPx – Адрес Описателя следующего Блока

Регистр LLPx (Linked list pointer register for channel x)- Регистр указателя на *Описатель следующего Блока* для Канала x.

При работе с *Пересылкой цепочками Блоков* до того как включить Канал необходимо записать в этот регистр адрес, указывающий на элемент LLI (*Описатель следующего Блока*) в памяти.

Смещение	LLP0 – 0x010 LLP1 – 0x068 LLP2 – 0x0c0 LLP3 – 0x118 LLP4 – 0x170 LLP5 – 0x1c8 LLP6 – 0x220 LLP7 – 0x278
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.7. Описание полей регистра LLPx**

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:2	LOC	Адрес <i>Описателя следующего Блока</i> . Младшие два разряда адреса не сохраняются, так как предполагается, что адрес выровнен по 32-разрядной границе. Доступ к LLI всегда 32-разрядный (HSIZE=2), выровнен по 32-разрядной границе.
1:0	LMS	List Master Select. Выбор АНВ интерфейса, по которому DMA произведёт чтение LLI. 00 — АНВ Master 1 01 — АНВ Master 2 10 — АНВ Master 3 11 — АНВ Master 4 DMA может служить мостом между несколькими различными шинами АНВ (до 4-х) и осуществлять передачи между устройствами на этих разных шинах.

Регистр LLPx выполняет две функции:

- Определение типа Пересылки.* Нулевое значение поля LOC выключает использование *Пересылок цепочками Блоков* и позволяет задать одноблочную *Пересылку*. См. таблицу «Типы пересылок и обновление регистров Канала» в разделе «Виды DMA Пересылок».
- Хранение адреса Описателя следующего Блока или адреса Статуса Источника/Приёмника.* Если поле LOC отлично от нуля, тогда в нем содержится

адрес *Описателя следующего Блока*. Регистр также может содержать адрес по которому производилась запись статуса *Источника/Приёмника* после завершения обработки Блока. См. таблицу «Структура *Описателя Блока (LLI)*» в разделе «Виды DMA Пересылок».

### 23.7.3.4 CTLx – Регистр управления

Регистр CTLx (Control register for channel x) - Регистр управления *Каналом x*.

Управление *Пересылкой DMA* осуществляется через поля этого регистра. При использовании *Пересылок цепочками Блоков* регистр CTLx обновляется из соответствующей области *Описателя Блока (LLI)*. Если включена возможность *обратной записи*, в конце передачи каждого *Блока* разряды CTLx[63:32] записываются в соответствующее местоположение *Описателя Блока*. См. таблицу «Типы пересылок и обновление регистров *Канала*» в разделе «Виды DMA Пересылок». Значение регистра CTLx необходимо задать до включения *Канала*.

Смещение	CTL0 – 0x018 CTL1 – 0x070 CTL2 – 0x0c8 CTL3 – 0x120 CTL4 – 0x178 CTL5 – 0x1d0 CTL6 – 0x228 CTL7 – 0x280
Тип	RW
Состояние по сбросу	0x00000002_00104825
Разрядность	64

**Таблица 23.8. Описание полей регистра CTLx**

Разряды	Поле	Описание
63:45	Undefined	Зарезервировано.
44	DONE	При использовании возможности <i>обратной записи</i> разряды CTLx[63:32] записываются в LLI в конце обработки <i>Блока</i> с установленным разрядом DONE. Программно опрашивая разряд DONE LLI.CTL, можно определить выполнен ли конкретный <i>Блок</i> в цепочке или нет. При настройке цепочки <i>Блоков</i> разряд DONE регистра LLI.CTL должен быть сброшен до запуска <i>Канала</i> .

Разряды	Поле	Описание
43:32	BLOCK_TS	Размер <i>Блока</i> (Block Transfer Size). Если размером <i>Блока</i> управляет DMA, то это поле устанавливается пользователем до запуска <i>Канала</i> . Значение поля задаёт количество <i>Одиночных обращений</i> на шине АНВ к <i>Источнику</i> , которое будет произведено за один <i>Блок</i> . <i>Одиночное обращение</i> соответствует одному «beat» АНВ. Разрядность <i>Одиночного обращения</i> определяется полем CTLx.SRC_TR_WIDTH. После запуска передачи BLOCK_TS возвращает текущее количество элементов данных, считанных из <i>Источника</i> (независимо от того управляет ли размером <i>Блока</i> DMA или периферийное устройство). Если размером <i>Блока</i> управляет <i>Источник/Приёмник</i> , тогда максимальное значение, прочитанное из BLOCK_TS, ограничено параметром DMAH_CHx_MAX_BLK_SIZE. Хотя фактических чтений из <i>Источника</i> может производиться больше (размер <i>Блока</i> может быть больше). Значение по сбросу — 0x2. <i>Замечание:</i> Для каждого <i>Канала</i> аппаратно задаётся максимальный размер <i>Блока</i> — DMAH_CHx_MAX_BLK_SIZE. Если размером <i>Блока</i> управляет DMA, тогда DMAH_CHx_MAX_BLK_SIZE определяет максимальное разрешённое значение для BLOCK_TS.
31:29	Undefined	Зарезервировано.
28	LLP_SRC_EN	Разрешение цепочки <i>Блоков Источника</i> . Цепочки <i>Блоков Источника</i> разрешены при LLPx.LOC != 0 и LLP_SRC_EN в 1.
27	LLP_DST_EN	Разрешение цепочки <i>Блоков Приёмника</i> . Цепочки <i>Блоков Приёмника</i> разрешены при LLPx.LOC != 0 и LLP_DST_EN в 1.
26:25	SMS	Source Master Select. Выбор АНВ интерфейса, по которому DMA обращается к <i>Источнику</i> . 00 — АНВ Master 1 01 — АНВ Master 2 10 — АНВ Master 3 11 — АНВ Master 4 DMA может служить мостом между несколькими различными шинами АНВ (до 4-х) и осуществлять передачи между устройствами на этих разных шинах.
24:23	DMS	Destination Master Select. Выбор АНВ интерфейса, по которому DMA обращается к <i>Приёмнику</i> . 00 — АНВ Master 1 01 — АНВ Master 2 10 — АНВ Master 3 11 — АНВ Master 4 DMA может служить мостом между несколькими различными шинами АНВ (до 4-х) и осуществлять передачи между устройствами на этих разных шинах.

Разряды	Поле	Описание																											
22:20	TT_FC	<p>Тип передачи и Управление размером <i>Блока</i> (Transfer Type and Flow Control). Возможны следующие варианты:</p> <table border="1"> <thead> <tr> <th>TT_FC</th><th>Тип передачи</th><th>Управление размером Блока</th></tr> </thead> <tbody> <tr> <td>000</td><td>Память-Память</td><td>DMA</td></tr> <tr> <td>001</td><td>Память-Периферия</td><td>DMA</td></tr> <tr> <td>010</td><td>Периферия-Память</td><td>DMA</td></tr> <tr> <td>011</td><td>Периферия-Периферия</td><td>DMA</td></tr> <tr> <td>100</td><td>Периферия-Память</td><td>Периферия</td></tr> <tr> <td>101</td><td>Периферия-Периферия</td><td>Источник</td></tr> <tr> <td>110</td><td>Память-Периферия</td><td>Периферия</td></tr> <tr> <td>111</td><td>Периферия-Периферия</td><td>Приёмник</td></tr> </tbody> </table> <p>При использовании цепочки <i>Блоков</i> поле TT_FC должно оставаться неизменным для всех <i>Блоков</i> цепочки.</p> <p><b>Примечание.</b> Для каждого <i>Канала</i> аппаратно задан тип управления размером <i>Блока</i>:  Размером <i>Блока</i> управляет только DMA (DMA_FC_ONLY)  Размером <i>Блока</i> управляет только <i>Источник</i> (SRC_FC_ONLY)  Размером <i>Блока</i> управляет только <i>Приёмник</i> (DST_FC_ONLY)  Размером <i>Блока</i> управляет или <i>Периферия</i> или DMA (ANY_FC)  Значение по сбросу:  TT_FC[0] = 1  TT_FC[1] = 1 если тип отличен от SRC_FC_ONLY  TT_FC[2] = 1 при SRC_FC_ONLY или DST_FC_ONLY</p> <p>Поле TT_FC[2] не реализовано и при чтении возвращает 0 при DMA_FC_ONLY.  Поле TT_FC[2:1] не реализовано и при чтении возвращает 10 при SRC_FC_ONLY.  Поле TT_FC[2:1] не реализовано и при чтении возвращает 11 при DST_FC_ONLY.</p>	TT_FC	Тип передачи	Управление размером Блока	000	Память-Память	DMA	001	Память-Периферия	DMA	010	Периферия-Память	DMA	011	Периферия-Периферия	DMA	100	Периферия-Память	Периферия	101	Периферия-Периферия	Источник	110	Память-Периферия	Периферия	111	Периферия-Периферия	Приёмник
TT_FC	Тип передачи	Управление размером Блока																											
000	Память-Память	DMA																											
001	Память-Периферия	DMA																											
010	Периферия-Память	DMA																											
011	Периферия-Периферия	DMA																											
100	Периферия-Память	Периферия																											
101	Периферия-Периферия	Источник																											
110	Память-Периферия	Периферия																											
111	Периферия-Периферия	Приёмник																											
19	Undefined	Зарезервировано.																											
18	DST_SCATTER_EN	<p>Разрешение режима <i>Разброса данных Приёмника</i>.  0 - Разброс запрещён.  1 - Разброс разрешён.  <i>Режим Разброса</i> применим только в том случае, если поле CTLx.DINC настроено на увеличение или уменьшение адреса <i>Приёмника</i>.</p> <p><b>Примечание.</b> Поле не реализовано и при чтении возвращает ноль, если не задан аппаратный параметр DMAH.CHx.DST.SCA.EN.</p>																											
17	SRC_GATHER_EN	<p>Разрешение режима <i>Сбора данных Источника</i>.  0 - Сбор запрещён.  1 - Сбор разрешён.  <i>Режим Сбора</i> применим только в том случае, если поле CTLx.SINC настроено на увеличение или уменьшение адреса <i>Источника</i>.</p> <p><b>Примечание.</b> Поле не реализовано и при чтении возвращает ноль, если не задан аппаратный параметр DMAH.CHx.SRC.GAT.EN.</p>																											

Разряды	Поле	Описание																		
16:14	SRC_MSIZЕ	<p>Размер пакета <i>Источника</i> (Source Burst Transaction Length). Определяет количество элементов данных разрядностью CTLx.SRC_TR_WIDTH, которое DMA считывает из <i>Источника</i> после поступления сигнала <i>запроса передачи пакета</i> от программного или аппаратного <i>Интерфейса Запросов</i>.</p> <p>Значение по сбросу - 1.</p> <p><i>Замечание:</i> это поле не имеет отношения к сигналам HBURST AHB. Зависимость значения SRC_MSIZЕ и числа элементов в пакете:</p> <table><tr><th>SRC_MSIZЕ</th><th>Число элементов данных разрядностью CTLx.SRC_TR_WIDTH в пакете</th></tr><tr><td>000</td><td>1</td></tr><tr><td>001</td><td>4</td></tr><tr><td>010</td><td>8</td></tr><tr><td>011</td><td>16</td></tr><tr><td>100</td><td>32</td></tr><tr><td>101</td><td>64</td></tr><tr><td>110</td><td>128</td></tr><tr><td>111</td><td>256</td></tr></table> <p><b>Примечание.</b> Аппаратный параметр DMAH_CHx_MAX_MULT_SIZE определяет максимальное значение SRC_MSIZЕ, которое можно задать программно и соответственно количество используемых разрядов MSIZE. При чтении неиспользуемых разрядов возвращается ноль.</p>	SRC_MSIZЕ	Число элементов данных разрядностью CTLx.SRC_TR_WIDTH в пакете	000	1	001	4	010	8	011	16	100	32	101	64	110	128	111	256
SRC_MSIZЕ	Число элементов данных разрядностью CTLx.SRC_TR_WIDTH в пакете																			
000	1																			
001	4																			
010	8																			
011	16																			
100	32																			
101	64																			
110	128																			
111	256																			
13:11	DEST_MSIZЕ	<p>Размер пакета <i>Приёмника</i> (Destination Burst Transaction Length). Определяет количество элементов данных разрядностью CTLx.DST_TR_WIDTH, которое DMA записывает в <i>Приёмник</i> после поступления сигнала <i>запроса передачи пакета</i> от программного или аппаратного <i>Интерфейса Запросов</i>.</p> <p>Значение по сбросу, размер пакета и ограничения те же, что и у SRC_MSIZЕ.</p>																		
10:9	SINC	<p>Тип изменения адреса <i>Источника</i> (Source Address Increment).</p> <p>Поле SINC и изменение адреса <i>Источника</i> после каждого обращения DMA к <i>Источнику</i>:</p> <p>00 - Инкремент 01 - Декремент 1x - Не меняется (подходит для работы с FIFO)</p> <p>Увеличение/уменьшение адреса выровнено согласно CTLx.SRC_TR_WIDTH.</p>																		
8:7	DINC	<p>Тип изменения адреса <i>Приёмника</i> (Destination Address Increment).</p> <p>Поле DINC и изменение адреса <i>Приёмника</i> после каждого обращения DMA к <i>Приёмнику</i>:</p> <p>00 - Инкремент 01 - Декремент 1x - Не меняется (подходит для работы с FIFO)</p> <p>Увеличение/уменьшение адреса выровнено согласно CTLx.DST_TR_WIDTH.</p>																		

Разряды	Поле	Описание																
6:4	SRC_TR_WIDTH	<p>Разрядность одиночного обращения АНВ <i>Источника</i>(Source Transfer Width).</p> <p>Разрядность обращения в зависимости от значения SRT_TR_WIDTH (см. HSIZE АНВ):</p> <table><tr><th>SRC_TR_WIDTH</th><th>Разрядность обращения АНВ (в битах)</th></tr><tr><td>000</td><td>8</td></tr><tr><td>001</td><td>16</td></tr><tr><td>010</td><td>32</td></tr><tr><td>011</td><td>64</td></tr><tr><td>100</td><td>128</td></tr><tr><td>101</td><td>256</td></tr><tr><td>11x</td><td>256</td></tr></table> <p>Разрядность настраиваемая через SRC_TR_WIDTH не должна превышать разрядности шины АНВ мастер порта. Поле SRC_TR_WIDTH существует, если не задан аппаратный параметр DMAH_CHx_STW, жёстко определяющий разрядность обращений. Если параметр DMAH_CHx_STW задан, то при чтении поля будет возвращаться аппаратно заданное значение DMAH_CHx_STW.</p>	SRC_TR_WIDTH	Разрядность обращения АНВ (в битах)	000	8	001	16	010	32	011	64	100	128	101	256	11x	256
SRC_TR_WIDTH	Разрядность обращения АНВ (в битах)																	
000	8																	
001	16																	
010	32																	
011	64																	
100	128																	
101	256																	
11x	256																	
3:1	DST_TR_WIDTH	<p>Разрядность одиночного обращения АНВ <i>Приёмника</i> (Destination Transfer Width).</p> <p>Возможные значения аналогичны полю SRC_TR_WIDTH.</p>																
0	INT_EN	<p>Разрешение прерываний (Interrupt Enable Bit).</p> <p>Если установлено в 1, то включаются все возможные источники прерываний. Работает как глобальная маска для всех прерываний <i>Канала</i>. Но прерывания в немаскируемых(Raw*) регистрах будут формироваться, даже если поле установлено в 0.</p> <p>Значение по сбросу - 1.</p>																

### 23.7.3.5 SSTATx – Статус Источника

Регистр SSTATx (Source status register for channel x) - Регистра статуса *Источника* для *Канала* x.

После завершения передачи *Блока* DMA может прочитать информацию о состоянии *Источника* по адресу заданному в SSTATARx. Эта информация сохраняется в регистре SSTATx и записывается в местоположение SSTATx в блоке LLI до старта следующего *Блока*. Регистр является временным буфером статуса *Источника* при его записи в LLI. Программному обеспечению следует брать информацию о статусе из блока LLI, а не из этого регистра.

Смещение	SSTAT0 – 0x020 SSTAT1 – 0x078 SSTAT2 – 0x0d0 SSTAT3 – 0x128 SSTAT4 – 0x180 SSTAT5 – 0x1d8 SSTAT6 – 0x230 SSTAT7 – 0x288
Тип	RW

Смещение	SSTAT0 – 0x020 SSTAT1 – 0x078 SSTAT2 – 0x0d0 SSTAT3 – 0x128 SSTAT4 – 0x180 SSTAT5 – 0x1d8 SSTAT6 – 0x230 SSTAT7 – 0x288
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.9. Описание полей регистра SSTATx

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	SSTAT	Информация о состоянии <i>Источника</i> , аппаратно загружаемая из местоположения указанного в регистре SSTATARx. Разрешение считывания статуса источника задаётся в регистре CFGx.SS_UPD_EN. Поле реализовано, если задан аппаратный параметр DMAH_CHx_STAT_SRC. Если параметр DMAH_CHx_STAT_SRC не задан, при чтении возвращается ноль.

### 23.7.3.6 DSTATx – Статус Приёмника

Регистр DSTATx (Destination status register for channel x) - Регистра статуса *Приёмника* для Канала x.

После завершения передачи *Блока* DMA может прочитать информацию о состоянии *Приёмника* по адресу заданному в DSTATARx. Эта информация сохраняется в регистре DSTATx и записывается в местоположение DSTATx в блоке LLI до старта следующего *Блока*. Регистр является временным буфером статуса *Приёмника* при его записи в LLI. Программному обеспечению следует брать информацию о статусе из блока LLI, а не из этого регистра.

Смещение	DSTAT0 – 0x028 DSTAT1 – 0x080 DSTAT2 – 0x0d8 DSTAT3 – 0x130 DSTAT4 – 0x188 DSTAT5 – 0x1e0 DSTAT6 – 0x238 DSTAT7 – 0x290
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.10. Описание полей регистра DSTATx

Разряд	Имя	Описание
63:32	Undefined	Зарезервировано.
31:0	DSTAT	Информация о состоянии <i>Приёмника</i> , аппаратно загружаемая из местоположения указанного в регистре DSTATARx.

Разряд	Имя	Описание
		Разрешение считывания статуса источника задаётся в регистре CFGx.DS_UPD_EN. Поле реализовано, если задан аппаратный параметр DMAH_CHx_STAT_DST. Если параметр DMAH_CHx_STAT_DST не задан, при чтении возвращается ноль.

### 23.7.3.7 SSTATARx – Адрес Статуса Источника

Регистр SSTATARx (Source status address register for channel x) - Адрес статуса *Источника* для *Канала x*.

После завершения передачи *Блока DMA* может аппаратно считать информацию о состоянии *Источника*. Адрес статусной информации определяется пользователем в регистре SSTATARx. Пользователь может выбрать любое местоположение системной памяти, где находится 32-разрядная информация о состоянии *Источника*. Например, если в качестве периферии используется блок SPI, пользователь может использовать один из регистров SPI для индикации состояния приёмника. В SSTATARx можно задать адреса таких регистров SPI как: SSI.CTRL, SSI.ISR, SSI.RXFLR и т.д.

Смещение	SSTATAR0 – 0x030 SSTATAR1 – 0x088 SSTATAR2 – 0x0e0 SSTATAR3 – 0x138 SSTATAR4 – 0x190 SSTATAR5 – 0x1e8 SSTATAR6 – 0x240 SSTATAR7 – 0x298
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.11. Описание полей регистра SSTATARx

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	SSTATAR	Указывает на местоположение памяти, из которого DMA считывает информацию о состоянии <i>Источника</i> в регистр SSTATx, а затем сохраняет её в соответствующую ячейку LLI до запуска следующего <i>Блока</i> .

### 23.7.3.8 DSTATARx – Адрес Статуса Приёмника

Регистр DSTATARx (Destination status address register for channel x) - адрес статуса *Приёмника* для *Канала x*.

После завершения передачи *Блока DMA* может считать информацию о состоянии *Приёмника*. Адрес статусной информации определяется пользователем в регистре DSTATARx. Пользователь может выбрать любое местоположение системной памяти, где находится 32-разрядная информация о состоянии *Приёмника*. Например, если в качестве периферии используется блок SPI, пользователь может использовать один из регистров SPI



для индикации состояния приёмника. В DSTATARx можно задать адреса таких регистров SPI как: SSI.CTRL, SSI.ISR, SSI.TXFLR и т.д.

Смещение	DSTATAR0 – 0x038 DSTATAR1 – 0x090 DSTATAR2 – 0x0e8 DSTATAR3 – 0x140 DSTATAR4 – 0x198 DSTATAR5 – 0x1f0 DSTATAR6 – 0x248 DSTATAR7 – 0x2a0
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.12. Описание полей регистра DSTATARx**

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	DSTATAR	Указывает на местоположение памяти, из которого DMA считывает информацию о состоянии <i>Приёмника</i> в регистр DSTATx, а затем сохраняет её в соответствующую ячейку LLI до запуска следующего <i>Блока</i> .

### 23.7.3.9 CFGx – Регистр Конфигурации

Регистр CFGx (Configuration register for channel x) - Регистр конфигурации *Канала* x.

Настройка *Пересылки* осуществляется через этот регистр. Регистр конфигурации останется неизменным для всех *Блоков* многоблочной *Пересылки*. Регистр необходимо проинициализировать до включения *Канала*.

Смещение	CFG0 – 0x040 CFG1 – 0x098 CFG2 – 0x0f0 CFG3 – 0x148 CFG4 – 0x1a0 CFG5 – 0x1f8 CFG6 – 0x250 CFG7 – 0x2a8
Тип	RW
Состояние по сбросу	0x0000_0004_0000_0Ex0
Разрядность	64

**Таблица 23.13. Описание полей регистра CFGx**

Разряды	Поле	Описание
63:47	Undefined	Зарезервировано.
46:43	DEST_PER	Определяет аппаратный Интерфейс запросов Приёмника для Канала x, если выбрана работа с аппаратным Интерфейсом запросов (CFGx.HS_SEL_DST = 0). Поле игнорируется, если выбран программный Интерфейс запросов (CFGx.HS_SEL_DST = 1). DMA будет взаимодействовать с Приёмником через заданный Интерфейс запросов.

Разряды	Поле	Описание
42:39	SRC_PER	Определяет аппаратный интерфейс Запросов Источника для Канала x, если выбрана работа с аппаратным Интерфейсом запросов (CFGx.HS_SEL_SRC = 0). Поле игнорируется, если выбран программный Интерфейс запросов (CFGx.HS_SEL_SRC = 1). DMA будет взаимодействовать с Источником через заданный Интерфейс запросов.
38	SS_UPD_EN	Разрешение считывания статуса Источника (Source Status Update Enable). Если SS_UPD_EN в 1, тогда информация о состоянии Источника считывается из местоположения, на которое указывает регистр SSTATARx, сохраняется в регистр SSTATx и записывается в соответствующую ячейку блока LLI.
37	DS_UPD_EN	Разрешение считывания статуса Приёмника (Destination Status Update Enable). Если DS_UPD_EN в 1, тогда информация о состоянии Приёмника считывается из местоположения, на которое указывает регистр DSTATARx, сохраняется в регистр DSTATx и записывается в соответствующую ячейку блока LLI.
36:34	PROTCTL	(Protection Control). Используется для задания сигналов HPROT[3:1] шины АНВ. Спецификация AMBA рекомендует по умолчанию устанавливать «non-cached, non-buffered, privileged data access». HPROT[0] всегда в единице. По сбросу устанавливается в 0x1, что соответствует рекомендациям AMBA. Зависимость сигналов АНВ от значения поля PROTCTL: HPROT[0] - 1 HPROT[1] - PROTCTL[0] HPROT[2] - PROTCTL[1] HPROT[3] - PROTCTL[2]
33	FIFO_MODE	Выбор режима FIFO(FIFO Mode Select). Определяет количество данных или свободное пространство FIFO, достаточное для обслуживания запроса передачи пакета. 0 - Для обслуживания запроса передачи пакета в FIFO должно быть достаточно данных/места для одиночной АНВ передачи заданной разрядность. 1 - Для обслуживания запроса передачи пакета Приёмника FIFO должно быть заполнено на половину и более. Для обслуживания запроса приёма пакета Источника FIFO должно быть свободно более чем на половину. Исключениями являются окончание пакета или окончание блока.
32	FCMODE	Flow Control Mode. Определяет момент обслуживания запроса передачи от источника, когда размером блока управляет приёмник. 0 - Запросы от источника обслуживаются по мере их поступления. DMA считывает данные из источника даже если не было запросов от приёмника (т.е. предварительная выборка/data pre-fetching разрешена). 1 - Запрос от источника не обрабатывается до тех пор, пока приёмник не установит свой запрос. В этом режиме гарантируется, что количество данных записанных в приёмник совпадёт с количеством, считанных из источника (приёмник определяет величину блока, и может прервать блок в любой момент.) Предварительная выборка данных запрещена.
31	RELOAD_DST	Автоматическое обновление адреса приёмника (Automatic Destination Reload). Поле задаёт режим автоповтора Блока для Приёмника. При пересылке с автоповтором в конце каждого блока регистр DARx автоматически возвращается к своему начальному значению. После этого запускается передача нового блока.
30	RELOAD_SRC	Автоматическое обновление адреса источника (Automatic Source Reload). Поле задаёт режим автоповтора Блока для Источника. При пересылке с автоповтором в конце каждого блока регистр SARx автоматически возвращается к своему начальному значению. После этого запускается передача нового блока.

Разряды	Поле	Описание
29:20	MAX_ABRST	Максимальный размер пакетной транзакции АНВ (Maximum AMBA Burst Length). 0 - ПО не ограничивает максимальный размер АНВ транзакции. Если MAX_ABRST больше нуля, тогда количество обращений внутри пакетной транзакции АНВ ограничено значением MAX_ABRST. <i>Замечание:</i> Это поле не реализовано и при чтении возвращает ноль, если не установлен аппаратный параметр DMAH_MABRST.
19	SRC_HS_POL	Активный уровень интерфейса запросов источника (Source Handshaking Interface Polarity). 0 - Активный уровень высокий. 1 - Активный уровень низкий.
18	DST_HS_POL	Активный уровень интерфейса запросов приёмника (Destination Handshaking Interface Polarity). 0 - Активный уровень высокий. 1 - Активный уровень низкий.
17	LOCK_B	Блокировка шины (Bus Lock Bit). Сигнал АНВ HLOCK формируется на период, определённый в CFGx.LOCK_B_L, если установлен разряд LOCK_B. <i>Замечание:</i> Это поле не реализовано и при чтении возвращает ноль, если не установлен аппаратный параметр DMAH_CHx_LOCK_EN.
16	LOCK_CH	Блокировка канала (Channel Lock Bit). Если каналу предоставлено управление мастер интерфейсом АНВ и установлен разряд CFGx.LOCK_CH, то ни один другой канал не получит доступ к АНВ в течении периода, задаваемого в CFGx.LOCK_CH_L. <i>Замечание:</i> Это поле не реализовано и при чтении возвращает ноль, если не установлен аппаратный параметр DMAH_CHx_LOCK_EN.
15:14	LOCK_B_L	Bus Lock Level. Определяет период в течении которого применяется разряд CFGx.LOCK_B. 00 - До завершения DMA Пересылки. 01 - До завершения передачи Блока. 1x - До завершения одиночного обращения DMA или пакета DMA. <i>Замечание:</i> Это поле не реализовано и при чтении возвращает ноль, если не установлен аппаратный параметр DMAH_CHx_LOCK_EN.
13:12	LOCK_CH_L	Channel Lock Level. Определяет интервал в течении которого применяется разряд CFGx.LOCK_CH. 00 - До завершения DMA Пересылки. 01 - До завершения передачи Блока. 1x - До завершения DMA транзакции. <i>Замечание:</i> Это поле не реализовано и при чтении возвращает ноль, если не установлен аппаратный параметр DMAH_CHx_LOCK_EN.
11	HS_SEL_SRC	Выбор интерфейса запросов источника (Source Software or Hardware Handshaking Select). 0 - Аппаратный интерфейс. Программно инициируемые транзакции DMA игнорируются. 1 - Программный интерфейс запросов. Аппаратно инициируемые транзакции игнорируются. Если источник является памятью, тогда разряд игнорируется. Значение по сбросу - 0x1 (программный интерфейс).
10	HS_SEL_DST	Выбор интерфейса запросов приёмника (Destination Software or Hardware Handshaking Select). 0 - Аппаратный интерфейс. Программно инициируемые транзакции DMA игнорируются. 1 - Программный интерфейс запросов. Аппаратно инициируемые транзакции игнорируются. Если приёмник является памятью, тогда разряд игнорируется. Значение по сбросу - 0x1 (программный интерфейс).

Разряды	Поле	Описание
9	FIFO_EMPTY	Признак наличия данных в FIFO. Может использоваться совместно с CFGx.CH_SUSP, для «чистого» отключения канала (преждевременного, до штатного отключения). 1 - FIFO свободно. 0 - В FIFO есть слова. Значение по сбросу - 0x1.
8	CH_SUSP	Channel Suspend. Запись 1 приостанавливает все передачи данных от источника, до тех пор пока не будет записан 0. Нет гарантии, что текущая транзакция будет завершена. Можно использовать совместно с FIFO_EMPTY, чтобы остановить канал без потери данных. 0 - Нормальная работа. 1 - Приостановка передач от источника.
7:5	CH_PRIOR	Приоритет Канала (Channel priority). Приоритет 7 - наивысший. 0 - низший. Значение по сбросу - соответствует номеру канала (для 0 - 0, ..., для 7 - 7).
4:0	Undefined	Зарезервировано.

### 23.7.3.10 SGRx – Регистр Сбора Источника

Регистр SGRx (Source gather register for channel x) - Регистр Сбора для канала x.

Если включён режим Сбора(Gather), то через определённое количество чтений из Источника по АНВ, адрес чтения автоматически увеличивается/уменьшается на заданную величину. Регистр позволяет задать количество чтений, при которых адрес меняется неразрывно (последовательно увеличивается/уменьшается), а также величину, на которую изменяется адрес чтения при приёме заданного количества слов. Увеличение/уменьшение адреса чтения задаётся полем SINC регистра CTLx. Если поле SINC регистра CTLx настроено на неизменный адрес, тогда адрес не изменяется и регистр SGRx игнорируется.

Смещение	SGR0 – 0x048 SGR1 – 0x0a0 SGR2 – 0x0f8 SGR3 – 0x150 SGR4 – 0x1a8 SGR5 – 0x200 SGR6 – 0x258 SGR7 – 0x2b0
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.14. Описание полей регистра SGRx

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:20	SGC	Счётчик Сбора Источника (Source gather count). Задаёт количество чтений АНВ разрядностью CTLx.SRC_TR_WIDTH, после которых адрес чтения изменяется скачком (см. поле SGI).
19:0	SGI	Интервал Сбора Источника (Source gather interval). Определяет величину изменения адреса чтения источника после достижения границы сбора. Т.е. каждые N чтений (задаётся полем SGC) из источника адрес чтения изменяется на величину, заданную этим полем, умноженную на

Разряды	Поле	Описание
		разрядность обращения по АНВ к Источнику в байтах (см. CTLx.SRC_TR_WIDTH).

### 23.7.3.11 DSRx – Регистр Разброса Приёмника

Регистр DSRx (Destination scatter register for channel x) - Регистр Разброса Приёмника для канала x.

Если включён режим Разброса(Scatter), то через определённое количество записей по АНВ в приёмник, адрес записи автоматически увеличивается/уменьшается на заданную величину. Регистр позволяет задать количество передач, при которых адрес меняется неразрывно (последовательно увеличивается/уменьшается), а также величину, на которую изменяется адрес передачи при выдаче заданного количества слов. Увеличение/уменьшение адреса записи задаётся полем DINC регистра CTLx. Если поле DINC регистра CTLx настроено на неизменный адрес, тогда адрес не изменяется и регистр DSRx игнорируется

Смещение	DSR0 – 0x050 DSR1 – 0x0a8 DSR2 – 0x100 DSR3 – 0x158 DSR4 – 0x1B0 DSR5 – 0x208 DSR6 – 0x260 DSR7 – 0x2b8
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.15. Описание полей регистра DSRx

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:20	DSC	Счётчик Разброса Приёмника (Destination scatter count). Задаёт количество записей по АНВ в Приёмник разрядностью CTLx.DST_TR_WIDTH, после которых адрес передачи изменяется скачком (см. поле DSI).
19:0	DSI	Интервал Разброса Приёмника (Destination scatter interval). Определяет величину изменения адреса записи приёмника при достижении границы разброса. Т.е. каждые N записей (задаётся полем DSC) в приёмник адрес передачи изменяется на величину, заданную этим полем, умноженную на разрядность обращения к Приёмнику (см. CTLx.DST_TR_WIDTH).

## 23.7.4 Регистры Прерываний

### 23.7.4.1 Регистры RawTfr, RawBlock, RawSrcTran, RawDstTran, RawErr

Перед маскированием события прерываний формируются в регистрах RawBlock, RawDstTran, RawErr, RawSrcTran, RawTfr. Каждый из регистров содержит информацию о прерываниях определённого типа

(IntTfr, IntBlock, IntSrcTran, IntDstTran, IntErr). См. Раздел «1.7 Прерывания». Каждый из разрядов этих регистров сбрасывается записью единицы в соответствующие разряды регистров ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr. Регистры Raw\* доступны по записи только для целей тестирования. В обычном режиме не рекомендуется производить запись в эти регистры.

Смещение	RawTfr RawBlock RawSrcTran RawDstTran RawErr	- 0x2c0 - 0x2c8 - 0x2d0 - 0x2d8 - 0x2e0
Тип	RW	
Состояние по сбросу	0x0	
Разрядность	64	

**Таблица 23.16. Описание полей регистров RawTfr, RawBlock, RawSrcTran, RawDstTran, RawErr**

Разряды	Поле	Описание
63:8	Undefined	Зарезервировано.
7:0	RAW	Разряды с седьмого по нулевой, указывают на прерывания в соответствующих каналах. Разряд 0 указывает на прерывание в канале 0, разряд 1 - на прерывание в канале 1, и т.д.

#### 23.7.4.2 Регистры StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran, StatusErr

Регистры StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran, StatusErr - маскируемые регистры прерываний. После маскирования события прерываний сохраняются в этих регистрах. См. Раздел «1.7 Прерывания».

Смещение	StatusTfr StatusBlock StatusSrcTran StatusDstTran StatusErr	- 0x2e8 - 0x2f0 - 0x2f8 - 0x300 - 0x308
Тип	R	
Состояние по сбросу	0x0	
Разрядность	64	

**Таблица 23.17. Описание полей регистров StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran, StatusErr**

Разряды	Поле	Описание
63:8	Undefined	Зарезервировано.
7:0	STATUS	Разряды с седьмого по нулевой указывают на событие прерывания в соответствующих каналах после маскирования. Разряд 0 указывает на прерывание в канале 0, разряд 1 - на прерывание в канале 1, и т.д.

### 23.7.4.3 Регистры MaskTfr, MaskBlock, MaskSrcTran, MaskDstTran, MaskErr

Регистры MaskTfr, MaskBlock, MaskSrcTran, MaskDstTran, MaskErr - регистры масок прерываний. См. Раздел «1.7 Прерывания». Содержимое регистров Raw\* маскируется регистрами Mask\* и сохраняется в регистры Status\*.

Смещение	MaskTfr MaskBlock MaskSrcTran MaskDstTran MaskErr	- 0x310 - 0x318 - 0x320 - 0x328 - 0x330
Тип	RW	
Состояние по сбросу	0x0	
Разрядность	64	

**Таблица 23.18. Описание полей регистров MaskTfr, MaskBlock, MaskSrcTran, MaskDstTran, MaskErr**

Разряды	Поле	Описание
63:8	Undefined	Зарезервировано.
15:8	INT_MASK_WE	Разрешение записи в соответствующие разряды поля INT_MASK. Например, запись 01X1(шестн.) в регистр MaskTfr установит 1 в нулевой разряд MaskTfr, не изменяя значения остальных разрядов.
7:0	INT_MASK	Запись единицы в разряды INT_MASK разрешает формирования прерывания для соответствующего Канала. Разряд 0 разрешает/запрещает прерывание в канале 0, разряд 1 разрешает/запрещает прерывание в канале 1, и т.д.

### 23.7.4.4 Регистры ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr

Регистры ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr - регистры сброса прерываний. См. Раздел «1.7 Прерывания».

Каждый разряд регистров Raw\* и Status\* очищается на одном и том же такте записью единицы в соответствующий разряд регистров Clear\*. Запись нуля не оказывает влияния.

Регистры недоступны по чтению.

Смещение	ClearTfr ClearBlock ClearSrcTran ClearDstTran ClearErr	- 0x338 - 0x340 - 0x348 - 0x350 - 0x358
Тип	W	
Состояние по сбросу	0x0	
Разрядность	64	



**Таблица 23.19. Описание полей регистров ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr**

Разряды	Поле	Описание
63:8	Undefined	Зарезервировано.
7:0	CLEAR	Разряды с седьмого по нулевой отвечают за сброс события прерывания в соответствующем канале. Разряд 0 сбрасывает прерывание в канале 0, разряд 1 - прерывание в канале 1, и т.д.

### 23.7.4.5 Регистр StatusInt

Регистр StatusInt (Status for each interrupt type) - Объединённый регистр прерываний. См. Раздел «Прерывания». Регистр доступен только по чтению.

Смещение	0x360
Тип	R
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.20. Описание полей регистра StatusInt**

Разряды	Поле	Описание
63:5	Undefined	Зарезервировано.
4	ERR	Взятое по ИЛИ содержимое регистра StatusErr.
3	DSTT	Взятое по ИЛИ содержимое регистра StatusDstTran.
2	SRCT	Взятое по ИЛИ содержимое регистра StatusSrcTran.
1	BLOCK	Взятое по ИЛИ содержимое регистра StatusBlock.
0	TFR	Взятое по ИЛИ содержимое регистра StatusTfr.

## 23.7.5 Регистры Интерфейса Запросов

В данном разделе описаны регистры *Программного Интерфейса Запросов*. Регистры *Программного Интерфейса Запросов* позволяют инициировать DMA транзакции (Пакеты DMA или *Одиночные обращения DMA*), аналогично сигналам Аппаратных Интерфейсов Запросов. См. Раздел «*Интерфейсы Запросов*». Чтобы разрешить Программный Интерфейс Запросов Источника для Канала *x*, необходимо установить в единицу CFGx.HS\_SEL\_SRC. Чтобы разрешить Программный Интерфейс Запросов Приёмника для Канала *x*, необходимо установить в единицу CFGx.HS\_SEL\_DST.

### 23.7.5.1 Регистр ReqSrcReg

Регистр ReqSrcReg (Source Software Transaction Request Register) - Программный запрос передачи пакета Источника.

Каждому Каналу соответствует определённый разряд запроса в этом регистре. Включение запроса передачи пакета от источника инициирует чтение пакета контроллером DMA от Источника. Если программный интерфейс запросов для источника канала *N* отключён, то соответствующие разряды этого регистра игнорируются.



Смещение	0x368
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.21. Описание полей регистра ReqSrcReg

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	SRC_REQ_WE	Разрешение записи в разряды SRC_REQ. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	SRC_REQ	Запросы передачи пакета от источника. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды происходит только с совместно установленными соответствующими разрядами поля SRC_REQ_WE, при этом канал должен быть включён через регистр ChEnReg.

### 23.7.5.2 Регистр ReqDstReg

Регистр ReqDstReg (Destination Software Transaction Request Register) - Программный запрос передачи пакета Приёмнику.

Каждому каналу соответствует определённый разряд запроса в этом регистре. Включение запроса передачи пакета Приёмнику инициирует передачу контроллером пакета приёмнику по шине АНВ. Если программный интерфейс запросов для приёмника канала N отключён, то соответствующие разряды этого регистра игнорируются.

Смещение	0x370
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.22. Описание полей регистра ReqDstReg

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	DST_REQ_WE	Разрешение записи в разряды DST_REQ. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	DST_REQ	Запросы передачи пакета приёмнику. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды происходит только, если установлены соответствующие разряды поля DST_REQ_WE.

### 23.7.5.3 Регистр SglReqSrcReg

Регистр SglReqSrcReg (Single Source Transaction Request Register) - Программный запрос одиночного чтения Источника.

Каждому каналу соответствует определённый разряд запроса в этом регистре. Включение запроса одиночного чтения источника инициирует одиночное обращения DMA к Источнику. Если программный интерфейс запросов для источника канала N отключён, то соответствующие разряды этого регистра игнорируются.

Смещение	0x378
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.23. Описание полей регистра SglReqSrcReg**

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	SRC_SGLREQ_WE	Разрешение записи в разряды SRC_SGLREQ. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	SRC_SGLREQ	Запросы одиночных чтений Источника. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды происходит только, если установлены соответствующие разряды поля SRC_SGLREQ_WE.

#### 23.7.5.4 Регистр SglReqDstReg

Регистр SglReqDstReg (Single Destination Transaction Request Register) - Программный запрос одиночного обращения Приёмника.

Каждому каналу соответствует определённый разряд запроса в этом регистре. Включение запроса одиночной записи приёмника инициирует одиночное обращения DMA к Приёмнику. Если программный интерфейс запросов для приёмника канала N отключён, то соответствующие разряды этого регистра игнорируются.

Смещение	0x380
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.24. Описание полей регистра SglReqDstRe**

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	DST_SGLREQ_WE	Разрешение записи в разряды DST_SGLREQ. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	DST_SGLREQ	Запросы одиночных обращений Приёмника. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды

Разряды	Поле	Описание
		происходит только, если установлены соответствующие разряды поля DST_SGLREQ_WE.

### 23.7.5.5 Регистр LstSrcReg

Регистр LstSrcReg (Last Source Transaction Request Register) - Последнее обращение Источника. Если для запроса (пакета или одиночного обращения) установлен этот признак, после обработки этого запроса DMA завершит обработку Блока. Каждому каналу соответствует определённый разряд в этом регистре. Разряды этого регистра игнорируются, если программный интерфейс запросов для источника канала N отключён или если источник не управляет размером Блока.

Смещение	0x388
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.25. Описание полей регистра LstSrcReg

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	LSTSRC_WE	Разрешение записи в разряды LSTSRC. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	LSTSRC	Признак последнего обращения (пакета или одиночного обращения) Источника. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды происходит только, если установлены соответствующие разряды поля LSTSRC_WE. 0 - Обращение не является последним в Блоке. 1 - Последнее обращение в текущем Блоке.

### 23.7.5.6 Регистр LstDstReg

Регистр LstDstReg (Last Destination Transaction Request Register) - Последнее обращение к Приёмнику. Если для запроса (пакета или одиночного обращения) установлен этот признак, после обработки этого запроса DMA завершит обработку Блока. Каждому каналу соответствует определённый разряд в этом регистре. Разряды этого регистра игнорируются, если программный интерфейс запросов для приёмника канала N отключён или если приёмник не управляет размером Блока.

Смещение	0x390
Тип	RW

Смещение	0x390
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.26. Описание полей регистра LstDstReg

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	LSTDST_WE	Разрешение записи в разряды LSTDST. 0 - Запись запрещена. 1 - Запись разрешена. Поле доступно только по записи.
7:0	LSTDST	Признак последнего обращения (пакета или одиночного обращения) Приёмника. Разряды от 7 до нуля, соответствуют каналам от 7-го до нулевого. Запись в эти разряды происходит только, если установлены соответствующие разряды поля LSTDST_WE. 0 - Обращение не является последним в Блоке. 1 - Последнее обращение в текущем Блоке.

## 23.7.6 Регистры Конфигурации и Управления

### 23.7.6.1 Регистр DmaCfgReg

Регистр DmaCfgReg (DMA Configuration Register) - Конфигурационный регистр DMA.

Регистр используется для глобального разрешения работы DMA. Перед включением каналов необходимо установить глобальное разрешение работы.

Смещение	0x398
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.27. Описание полей регистра DmaCfgReg

Разряды	Поле	Описание
63:1	Undefined	Зарезервировано.
0	DMA_EN	Разрешения работы DMA. 0 - DMA отключён. 1 - DMA включён. Если во время работы любого из Каналов записать в этот разряд 0, то при чтении он будет возвращать 1 до тех пор, пока DMA не остановит активность на всех Каналах. После этого разряд будет возвращать 0.

### 23.7.6.2 Регистр ChEnReg

Регистр ChEnReg (Channel Enable Register) - Регистр включения каналов DMA.

Регистр позволяет определять доступные (неактивные) Каналы, а также разрешать/запрещать работу необходимых каналов. Все разряды регистра очищаются в ноль, когда глобальный разряд включения DMA (DmaCfgReg[0]) в нуле. Если глобальный

разряд включения DMA в нуле, записи в этот регистр будут игнорироваться и чтения всегда будут возвращать ноль.

Смещение	0x3A0
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

**Таблица 23.28. Описание полей регистра ChEnReg**

Разряды	Поле	Описание
63:16	Undefined	Зарезервировано.
15:8	CH_EN_WE	Разрешения записи в разряды разрешения работы каналов.
7:0	CH_EN	Разрешение/Запрет работы канала. 0 - Запрет работы. 1 - Разрешение работы. Каждому каналу соответствует один разряд поля CH_EN. Разряды поля CH_EN аппаратно сбрасываются в ноль после завершения последней АНВ транзакции к Приёмнику в DMA Пересылке. Опрашивая эти разряды, поможет определить доступные для работы каналы. Запись в разряды этого поля возможна только, если установлены соответствующие разряды CH_EN_WE. Например, запись значения 01X1 (шестн.) в этот регистр вызовет установку единицы в разряде ChEnReg[0], а остальные разряды ChEnReg[7:1] оставит незатронутыми. См. 1.6 «Выключение канала до завершения пересылки».

### 23.7.6.3 Регистр DmaIdReg

Регистр DmaIdReg (DMA ID Register) - ID регистр DMA.

Регистр доступен только по чтению. Возвращает аппаратно заданный идентификатор контроллера DMA.

Смещение	0x3a8
Тип	R
Состояние по сбросу	0хеее ???
Разрядность	64

**Таблица 23.29. Описание полей регистра DmaIdReg**

Разряды	Поле	Описание
63:32	Undefined	Зарезервировано.
31:0	DMA_ID	Аппаратно заданный ID DMA.

### 23.7.6.4 Регистр DmaTestReg

Регистр DmaTestReg (DMA Test Register) - Тестовый регистр DMA.

Регистр используется для перевода ведомого порта АНВ DMA в тестовый режим. В тестовом режиме значение, записанное в регистр DMA доступный по записи, совпадает со значением, которое возвращается последующими чтениями. В нормальном режиме значение, возвращаемое при чтении, зависит от состояния DMA и может не совпадать с записанным.

Смещение	0x3b0
Тип	RW
Состояние по сбросу	0x0
Разрядность	64

Таблица 23.30. Описание полей регистра DmaTestReg

Разряды	Поле	Описание
63:1	Undefined	Зарезервировано.
0	TEST_SLV_IF	Переводит конфигурационный (ведомый) порт АНВ в тестовый режим. В тестовом режиме при чтении регистров, доступных по записи, всегда возвращается значение, установленное предшествующей записью. Разряд не разрешает запись в регистры доступные только по чтению. 0 - Нормальный режим. 1 - Тестовый режим.

## 23.7.7 Регистры Аппаратных Параметров

Регистры аппаратных параметров - доступные только по чтению регистры, содержащие информацию об аппаратных параметрах DMA. О каждом из Каналов DMA можно узнать следующую информацию:

Таблица 23.31. Аппаратные параметры Каналов DMA

Параметр Канала x (x от 0 до 7)	Описание
Chx_FIFO_DEPTH[2:0]	Глубина FIFO канала в байтах. 0 = 8 байт 1 = 16 байт 2 = 32 байта 3 = 64 байта 4 = 128 байт 5 = 256 байт
CHx_SMS[2:0]	АНВ порт Источника. 0 = АНВ 1 1 = АНВ 2 2 = АНВ 3 3 = АНВ 4 4 = Жёстко не задан
CHx_LMS[2:0]	АНВ порт LLP. 0 = АНВ 1 1 = АНВ 2 2 = АНВ 3 3 = АНВ 4 4 = Жёстко не задан
CHx_DMS[2:0]	АНВ порт Приёмника. 0 = АНВ 1 1 = АНВ 2 2 = АНВ 3 3 = АНВ 4 4 = Жёстко не задан

Параметр Канала x (x от 0 до 7)	Описание
CHx_MAX_MULT_SIZE[2:0]	Максимальный размер Пакета DMA. 0 = 4 1 = 8 2 = 16 3 = 32 4 = 64 5 = 128 6 = 256 7 = Зарезервировано
Chx_FC[1:0]	Тип управления размером Блока. 0 = только DMA 1 = только Источник 2 = только Приёмник 3 = любое устройство Если выбран тип управления 3, тогда управляющее устройство назначается программно.
CHx_HC_LLP	Поддержка цепочек Блоков. 1 - Регистр LLP всегда в нуле. Передачи цепочками блоков не поддерживаются. Информация о статусах Источника и Приёмника не обрабатывается (см. STATx и DSTATx), т. е информация о статусе не считывается из периферии и не записывается в блок LLI. Отключена возможность обратной записи.
CHx_CTL_WB_EN	Поддержка обратной записи. 1 - Поддержка обратной записи реализована (см. CTLx, SSTATx, DSTATx).
CHx_MULTI_BLK_EN	Поддержка многоблочных Пересылок. 0 - Поддерживаются только одноблочные Пересылки. 1 - Многоблочные Пересылки поддерживаются
CHx_LOCK_EN	Поддержка блокировок Канала и Шины. 0 - Поддержка блокировки Канала и Шины не реализована 1 - Поддержка блокировки Канала и Шины реализована
CHx_SRC_GAT_EN	Поддержка Сбора Источника. 0 - Сбор Источника не поддерживается 1 - Сбор Источника поддерживается
CHx_DST_SCA_EN	Поддержка Разброса Приёмника. 0 - Разброса Приёмника не поддерживается 1 - Разброса Приёмника поддерживается
CHx_STAT_SRC	Поддержка обработки статуса Источника. 0 - чтение и обратная запись статуса не поддерживается 1 - чтение и обратная запись статуса поддерживается
CHx_STAT_DST	Поддержка обработки статуса Приёмника. 0 - чтение и обратная запись статуса не поддерживается 1 - чтение и обратная запись статуса поддерживается
CHx_STW	Разрядность передач Источника. 0 = Не задана 1 = 8 2 = 16 3 = 32 4 = 64 5 = 128 6 = 256 7 = Зарезервировано

Параметр Канала x (x от 0 до 7)	Описание
CHx_DTW	Разрядность передач Приёмника. 0 = Не задана 1 = 8 2 = 16 3 = 32 4 = 64 5 = 128 6 = 256 7 = Зарезервировано

### 23.7.7.1 Регистр DMA\_COMP\_PARAMS\_6

Регистр DMA\_COMP\_PARAMS\_6 – 6-ой регистр аппаратных параметров DMA. Содержит информацию об аппаратной конфигурация канала 7.

Смещение	0x3c8
Тип	R
Состояние по сбросу	Определяется настройками блока
Разрядность	64

Таблица 23.32. Описание полей регистра DMA\_COMP\_PARAMS\_6

Разряды	Поле	Описание
63	Reserved	Зарезервировано.
62:60	CH7_FIFO_DEPTH	Глубина FIFO канала в байтах.
59:57	CH7_SMS	АНВ порт Источника.
56:54	CH7_LMS	АНВ порт LLP.r
53:51	CH7_DMS	АНВ порт Приёмника.
50:48	CH7_MAX_MULT_SIZE	Максимальный размер Пакета.
47:46	CH7_FC	Тип управления размером Блока.
45	CH7_HC_LLP	Поддержка цепочек Блоков.
44	CH7_CTL_WB_EN	Поддержка обратной записи.
43	CH7_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
42	CH7_LOCK_EN	Поддержка блокировок Канала и Шиной.
41	CH7_SRC_GAT_EN	Поддержка Сбора Источника.
40	CH7_DST_SCA_EN	Поддержка Разброса Приёмника.
39	CH7_STAT_SRC	Поддержка обработки статуса Источника.
38	CH7_STAT_DST	Поддержка обработки статуса Приёмника.
37:35	CH7_STW	Разрядность передач Источника.
34:32	CH7_DTW	Разрядность передач Приёмника.
31:0	Reserved	Зарезервировано.

### 23.7.7.2 Регистр DMA\_COMP\_PARAMS\_5

Регистр DMA\_COMP\_PARAMS\_5 - 5-ый регистр аппаратных параметров DMA. Содержит информацию об аппаратной конфигурация каналов 6 и 5.

Смещение	0x3d0
Тип	R
Состояние по сбросу	Определяется настройками блока
Разрядность	64



**Таблица 23.33. Описание полей регистра DMA\_COMP\_PARAMS\_5**

Разряды	Поле	Описание
63	Reserved	Зарезервировано.
62:60	CH5_FIFO_DEPTH	Глубина FIFO канала в байтах.
59:57	CH5_SMS	АВВ порт Источника.
56:54	CH5_LMS	АВВ порт LLP.
53:51	CH5_DMS	АВВ порт Приёмника.
50:48	CH5_MAX_MULT_SIZE	Максимальный размер Пакета.
47:46	CH5_FC	Тип управления размером Блока.
45	CH5_HC_LLP	Поддержка цепочек Блоков.
44	CH5_CTL_WB_EN	Поддержка обратной записи.
43	CH5_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
42	CH5_LOCK_EN	Поддержка блокировок Канала и Шины.
41	CH5_SRC_GAT_EN	Поддержка Сбора Источника.
40	CH5_DST_SCA_EN	Поддержка Разброса Приёмника.
39	CH5_STAT_SRC	Поддержка обработки статуса Источника.
38	CH5_STAT_DST	Поддержка обработки статуса Приёмника.
37:35	CH5_STW	Разрядность передач Источника.
34:32	CH5_DTW	Разрядность передач Приёмника.
31	Reserved	Зарезервировано.
30:28	CH6_FIFO_DEPTH	Глубина FIFO канала в байтах.
27:25	CH6_SMS	АВВ порт Источника.
24:22	CH6_LMS	АВВ порт LLP.
21:19	CH6_DMS	АВВ порт Приёмника.
18:16	CH6_MAX_MULT_SIZE	Максимальный размер Пакета.
15:14	CH6_FC	Тип управления размером Блока.
13	CH6_HC_LLP	Поддержка цепочек Блоков.
12	CH6_CTL_WB_EN	Поддержка обратной записи.
11	CH6_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
10	CH6_LOCK_EN	Поддержка блокировок Канала и Шины.
9	CH6_SRC_GAT_EN	Поддержка Сбора Источника.
8	CH6_DST_SCA_EN	Поддержка Разброса Приёмника.
7	CH6_STAT_SRC	Поддержка обработки статуса Источника.
6	CH6_STAT_DST	Поддержка обработки статуса Приёмника.
5:3	CH6_STW	Разрядность передач Источника.
2:0	CH6_DTW	Разрядность передач Приёмника.

**23.7.7.3 Регистр DMA\_COMP\_PARAMS\_4**

Регистр DMA\_COMP\_PARAMS\_4 - 4-ый регистр аппаратных параметров DMA. Содержит информацию об аппаратной конфигурации каналов 4 и 3.

Смещение	0x3d8
Тип	R
Состояние по сбросу	Определяется настройками блока
Разрядность	64

**Таблица 23.34. Описание полей регистра DMA\_COMP\_PARAMS\_4**

Разряды	Поле	Описание
63	Reserved	Зарезервировано.
62:60	CH3_FIFO_DEPTH	Глубина FIFO канала в байтах.
59:57	CH3_SMS	АВВ порт Источника.
56:54	CH3_LMS	АВВ порт LLP.
53:51	CH3_DMS	АВВ порт Приёмника.

Разряды	Поле	Описание
50:48	CH3_MAX_MULT_SIZE	Максимальный размер Пакета.
47:46	CH3_FC	Тип управления размером Блока.
45	CH3_HC_LLP	Поддержка цепочек Блоков.
44	CH3_CTL_WB_EN	Поддержка обратной записи.
43	CH3_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
42	CH3_LOCK_EN	Поддержка блокировок Канала и Шины.
41	CH3_SRC_GAT_EN	Поддержка Сбора Источника.
40	CH3_DST_SCA_EN	Поддержка Разброса Приёмника.
39	CH3_STAT_SRC	Поддержка обработки статуса Источника.
38	CH3_STAT_DST	Поддержка обработки статуса Приёмника.
37:35	CH3_STW	Разрядность передач Источника.
34:32	CH3_DTW	Разрядность передач Приёмника.
31	Reserved	Зарезервировано.
30:28	CH4_FIFO_DEPTH	Глубина FIFO канала в байтах.
27:25	CH4_SMS	АНВ порт Источника.
24:22	CH4_LMS	АНВ порт LLP.
21:19	CH4_DMS	АНВ порт Приёмника.
18:16	CH4_MAX_MULT_SIZE	Максимальный размер Пакета.
15:14	CH4_FC	Тип управления размером Блока.
13	CH4_HC_LLP	Поддержка цепочек Блоков.
12	CH4_CTL_WB_EN	Поддержка обратной записи.
11	CH4_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
10	CH4_LOCK_EN	Поддержка блокировок Канала и Шины.
9	CH4_SRC_GAT_EN	Поддержка Сбора Источника.
8	CH4_DST_SCA_EN	Поддержка Разброса Приёмника.
7	CH4_STAT_SRC	Поддержка обработки статуса Источника.
6	CH4_STAT_DST	Поддержка обработки статуса Приёмника.
5:3	CH4_STW	Разрядность передач Источника.
2:0	CH4_DTW	Разрядность передач Приёмника.

### 23.7.7.4 Регистр DMA\_COMP\_PARAMS\_3

Регистр DMA\_COMP\_PARAMS\_3 - 3-ий регистр аппаратных параметров DMA. Содержит информацию об аппаратной конфигурации каналов 2 и 1.

Смещение	0x3e0
Тип	R
Состояние по сбросу	Определяется настройками блока
Разрядность	64

Таблица 23.35. Описание полей регистра DMA\_COMP\_PARAMS\_3

Разряды	Поле	Описание
63	Reserved	Зарезервировано.
62:60	CH1_FIFO_DEPTH	Глубина FIFO канала в байтах.
59:57	CH1_SMS	АНВ порт Источника.
56:54	CH1_LMS	АНВ порт LLP.
53:51	CH1_DMS	АНВ порт Приёмника.
50:48	CH1_MAX_MULT_SIZE	Максимальный размер Пакета.
47:46	CH1_FC	Тип управления размером Блока.
45	CH1_HC_LLP	Поддержка цепочек Блоков.
44	CH1_CTL_WB_EN	Поддержка обратной записи.
43	CH1_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
42	CH1_LOCK_EN	Поддержка блокировок Канала и Шины.
41	CH1_SRC_GAT_EN	Поддержка Сбора Источника.
40	CH1_DST_SCA_EN	Поддержка Разброса Приёмника.

Разряды	Поле	Описание
39	CH1_STAT_SRC	Поддержка обработки статуса Источника.
38	CH1_STAT_DST	Поддержка обработки статуса Приёмника.
37:35	CH1_STW	Разрядность передач Источника.
34:32	CH1_DTW	Разрядность передач Приёмника.
31	Reserved	Зарезервировано.
30:28	CH2_FIFO_DEPTH	Глубина FIFO канала в байтах.
27:25	CH2_SMS	АНВ порт Источника.
24:22	CH2_LMS	АНВ порт LLP.
21:19	CH2_DMS	АНВ порт Приёмника.
18:16	CH2_MAX_MULT_SIZE	Максимальный размер Пакета.
15:14	CH2_FC	Тип управления размером Блока.
13	CH2_HC_LLP	Поддержка цепочек Блоков.
12	CH2_CTL_WB_EN	Поддержка обратной записи.
11	CH2_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
10	CH2_LOCK_EN	Поддержка блокировок Канала и Шины.
9	CH2_SRC_GAT_EN	Поддержка Сбора Источника.
8	CH2_DST_SCA_EN	Поддержка Разброса Приёмника.
7	CH2_STAT_SRC	Поддержка обработки статуса Источника.
6	CH2_STAT_DST	Поддержка обработки статуса Приёмника.
5:3	CH2_STW	Разрядность передач Источника.
2:0	CH2_DTW	Разрядность передач Приёмника.

### 23.7.7.5 Регистр DMA\_COMP\_PARAMS\_2

Регистр DMA\_COMP\_PARAMS\_2 - 2-ой регистр аппаратных параметров DMA. Содержит информацию об аппаратной конфигурации канала 0 и типах многоблочных пересылок для всех каналов.

Смещение	0x3e8
Тип	R
Состояние по сбросу	Определяется настройками блока
Разрядность	64

Таблица 23.36. Описание полей регистра DMA\_COMP\_PARAMS\_2

Разряды	Поле	Описание
63:60	CH7_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 7. 0 = NO_HARDCODE Любые типы. 1 = CONT_RELOAD SARx - смежный. DAR и CTL - автоповтор. 2 = RELOAD_CONT SARx и CTLx - автоповтор. DARx - смежный. 3 = RELOAD_RELOAD SARx, DARx, и CTLx - автоповтор. 4 = CONT_LL SARx - смежный. DARx, CTLx, и LLPx загружаются из блоков LLI. 5 = RELOAD_LL SARx - автоповтор. DARx, CTLx, и LLPx загружаются из блоков LLI. 6 = LLP_CONT SARx, CTLx, и LLPx загружаются из блоков LLI. DARx - смежный. 7 = LLP_RELOAD SARx, CTLx, и LLPx загружаются из блоков LLI. DARx - автоповтор. 8 = LLP_LL SARx, DARx, CTLx, и LLPx загружаются из блоков LLI цепочки Блоков.
59:56	CH6_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 6.
55:52	CH5_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 5.
51:48	CH4_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 4.
47:44	CH3_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 3.
43:40	CH2_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 2.
39:36	CH1_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 1.
35:32	CH0_MULTI_BLK_TYPE	Тип многоблочных Пересылок канала 0.
31	Reserved	Зарезервировано.
30:28	CH0_FIFO_DEPTH	Глубина FIFO канала в байтах.
27:25	CH0_SMS	АНВ порт Источника.
24:22	CH0_LMS	АНВ порт LLP.
21:19	CH0_DMS	АНВ порт Приёмника.
18:16	CH0_MAX_MULT_SIZE	Максимальный размер Пакета.
15:14	CH0_FC	Тип управления размером Блока.
13	CH_HC_LL	Поддержка цепочек Блоков.
12	CH0_CTL_WB_EN	Поддержка обратной записи.
11	CH0_MULTI_BLK_EN	Поддержка многоблочных Пересылок.
10	CH0_LOCK_EN	Поддержка блокировок Канала и Шины.
9	CH0_SRC_GAT_EN	Поддержка Сбора Источника.
8	CH0_DST_SCA_EN	Поддержка Разброса Приёмника.
7	CH0_STAT_SRC	Поддержка обработки статуса Источника.
6	CH0_STAT_DST	Поддержка обработки статуса Приёмника.
5:3	CH0_STW	Разрядность передач Источника.
2:0	CH0_DTW	Разрядность передач Приёмника.

### 23.7.7.6 Регистр DMA\_COMP\_PARAMS\_1

Регистр DMA\_COMP\_PARAMS\_1 - 1-ый регистр аппаратных параметров DMA.

Смещение	0x3f0
Тип	R

Смещение	0x3f0
Состояние по сбросу	Определяется настройками блока
Разрядность	64

Таблица 23.37. Описание полей регистра DMA\_COMP\_PARAMS\_1

Разряды	Поле	Описание
63:62	Reserved	Зарезервировано.
61	STATIC_ENDIAN_SELECT	Способ задания порядка байтов в обменах АНВ. 0 - Для каждого порта АНВ порядок байтов определяется индивидуально отдельными аппаратными входами DMA. 1 - Порядок байтов для всех портов АНВ (как ведомых, так и ведущих) одинаков и жёстко задан при конфигурации.
60	ADD_ENCODED_PARAM	Определяет наличие регистров аппаратных параметров от 6-го до 1-го (DMA_COMP_PARAMS ..).
59:55	NUM_HS_INT	Количество аппаратных интерфейсов запросов (от 0 до 16).
54:53	M4_HDATA_WIDTH	Разрядность ведущего АНВ порта 4. 0 - 32 разряда 1 - 64 разряда 2 - 128 разрядов 3 - 256 разрядов
52:51	M3_HDATA_WIDTH	Разрядность ведущего АНВ порта 3.
50:49	M2_HDATA_WIDTH	Разрядность ведущего АНВ порта 2.
48:47	M1_HDATA_WIDTH	Разрядность ведущего АНВ порта 1.
46:45	S_HDATA_WIDTH	Разрядность конфигурационного АНВ порта.
44:43	NUM_MASTER_INT	Количество ведущих портов АНВ. 0 - 1 ведущий ... 3 - 4 ведущих
42:40	NUM_CHANNELS	Количество Каналов. 0 - 1 канал ... 7 - 8 каналов
39:36	Reserved	Зарезервировано.
35	MABRST	Разрешение программного ограничения размера пакетов АНВ. 0 - Максимальный размер пакета АНВ определяется размером FIFO Канала. DMA может заполнять и очищать FIFO за одно пакетное обращение. 1 - Максимальный размер пакета АНВ определяется настройками регистров Канала.
34:33	INTR_IO	Аппаратно заданный тип сигналов прерываний: 0 - ALL Для каждого канала выведены все 5 возможных прерываний (любому прерыванию каждого из каналов соответствует отдельный сигнал). 1 - TYPE Выведены 5 сигналов прерываний согласно типу (каждый из сигналов указывает на возникновение прерывание заданного типа в любом из каналов). Также выведен объединённый сигнал прерываний (указывает на наличие любого прерывания в любом канале). 2 - COMBINED Выведен только объединённый сигнал прерываний (указывает на наличие любого прерывания в любом канале). 3 - зарезервировано

Разряды	Поле	Описание
32	BIG_ENDIAN	Порядок байтов в обменах АНВ. 0 - Сначала передаются младшие байты (little-endian) 1 - Сначала передаются старшие байты (big-endian) Поле работает только при установленном в единицу поле STATIC_ENDIAN_SELECT.
31:28	CH7_MAX_BLK_SIZE	Максимальный размер Блока канала 7. Определяет максимально количество обращений разрядностью передач источника в Блоке. Если размером Блока управляет DMA, разрешается настраивать размер Блока, не превышающий значение этого параметра. Если размером Блока управляет периферия, то размер Блока может быть больше значения этого параметра. Параметр ограничивает размер границы Сбора/Разброса. 0x0 = 3 0x1 = 7 0x2 = 15 0x3 = 31 0x4 = 63 0x5 = 127 0x6 = 255 0x7 = 511 0x8 = 1023 0x9 = 2047 0xa = 4095
27:24	CH6_MAX_BLK_SIZE	Максимальный размер Блока канала 6.
23:20	CH5_MAX_BLK_SIZE	Максимальный размер Блока канала 5.
19:16	CH4_MAX_BLK_SIZE	Максимальный размер Блока канала 4.
15:12	CH3_MAX_BLK_SIZE	Максимальный размер Блока канала 3.
11:8	CH2_MAX_BLK_SIZE	Максимальный размер Блока канала 2.
7:4	CH1_MAX_BLK_SIZE	Максимальный размер Блока канала 1.
3:0	CH0_MAX_BLK_SIZE	Максимальный размер Блока канала 0.

### 23.7.8 Регистр DMA\_Component\_ID\_Register

Регистр DMA\_Component\_ID\_Register (DMA Component ID Register) - ID регистр компонента DMA.

Регистр доступен только по чтению. Регистр позволяет определить тип и версию компонента Designware.

Смещение	0x3f8
Тип	R
Состояние по сбросу	3231_372A__4457_1110
Разрядность	64

Таблица 23.38. Описание полей регистра DMA\_Component\_ID\_Register

Разряды	Поле	Описание
63:32	DMA_COMP_VERSION	Версия компонента.
31:0	DMA_COMP_TYPE	Тип компонента Designware. Значение по сбросу - 0x44571110.

## 24. RWC: СЧЕТЧИК РЕАЛЬНОГО ВРЕМЕНИ, WAKE-КОНТРОЛЛЕР

### 24.1 Введение

Блок RWC расположен в батарейном домене питания и имеет следующие возможности:

- Встроенные кварцевый осциллятор LFE и RC-осциллятор LFI на частоту 32 кГц
- Интегрированные конденсаторы для осциллятора LFE
- Формирование частоты 1 Гц с возможностью подстройки до точности менее 1 ppm
- 32-х разрядный счетчик реального времени
- Регистр ALARM для сигнализации о совпадении счетчика времени с заданным значением
- 32-х разрядный регистр общего назначения
- Пробуждение системы из малопотребляющих режимов по внешнему сигналу WKUP с настраиваемым уровнем срабатывания
- Управление входом в режим SHUTDOWN и выходом из него

### 24.2 Описание функционирования

#### 24.2.1 Структурная схема

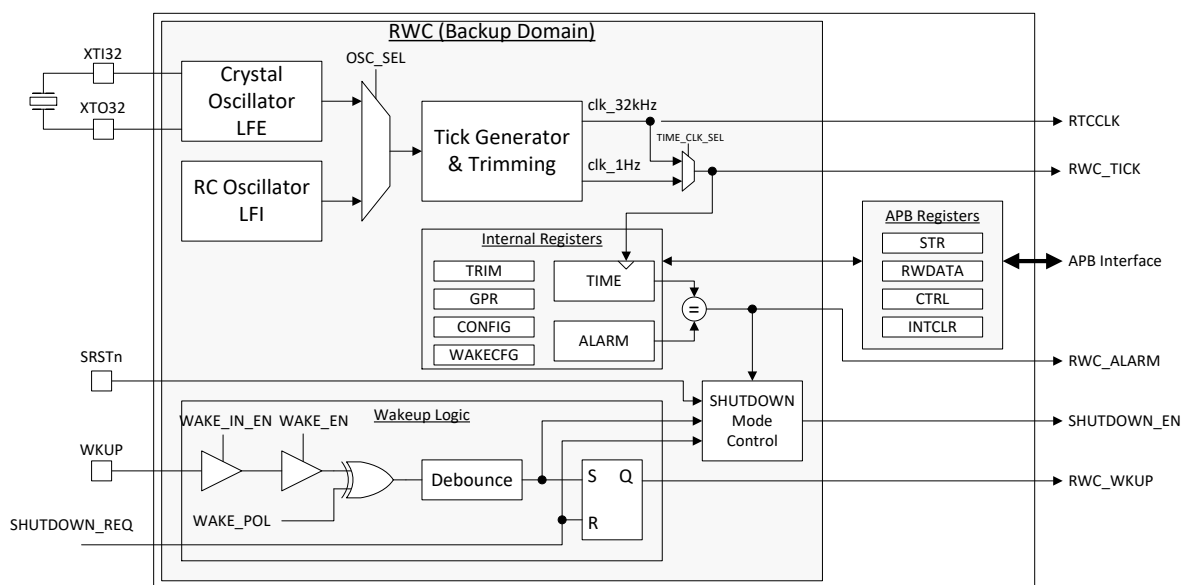


Рисунок 24.1. Структурная схема RWC

Компоненты RWC, расположенные в батарейном домене питания:

- Осцилляторы LFI, LFE
- Блок подстройки тактовой частоты счетчика времени
- Внутренние регистры RWC
- Контроллер внешнего прерывания WKUP



- Схема управления режимом SHUTDOWN

Регистры, доступные по интерфейсу APB расположены в отключаемом домене PD\_SYS и также относятся к блоку RWC.

В следующей таблице перечислены внешние сигналы и интерфейсы RWC.

**Таблица 24.1. Сигналы RWC**

Сигнал	Описание
XTI32	Вход осциллятора LFE для подключения внешнего кварцевого резонатора либо внешнего генератора тактового сигнала.
XTO32	Выход осциллятора LFE для подключения внешнего кварцевого резонатора.
SRSTn	Внешний системный сброс. Используется в блоке RWC для выхода из режима SHUTDOWN.
WKUP	Внешнее прерывание для пробуждения системы.
APB Interface	Интерфейс доступа к регистрам RWC.
RTCCLK	Выходной тактовый сигнал 32 кГц.
RWC_TICK	Выход секундного stroba. Используется в качестве прерывания NVIC.
RWC_ALARM	Прерывание, сигнализирующее совпадение значения счетчика времени с значением регистра ALARM.
RWC_WKUP	Прерывание по внешнему событию WKUP.
SHUTDOWN_REQ	Запрос перехода в режим SHUTDOWN.
SHUTDOWN_EN	Разрешение перехода в режим SHUTDOWN.

## 24.2.2 Доступ ко внутренним регистрам

Блок RWC содержит два набора регистров: регистры, доступные по интерфейсу APB и расположенные в домене PD\_SYS, и внутренние регистры с батарейным питанием.

Доступ ко внутренним регистрам выполняется через регистры RWC\_RWDATA и RWC\_CTRL по интерфейсу APB.

Для записи внутреннего регистра необходимо поместить записываемые данные в регистр RWC\_RWDATA, адрес регистра – в поле RSEL и записать значение 0x1 в поле CMD. Завершение записи подтверждается чтением CMD = 0.

Для чтения внутреннего регистра необходимо поместить адрес регистра в поле RSEL и записать значение 0x2 в поле CMD. Считанное значение регистра будет доступно в регистре RWC\_RWDATA после сброса поля CMD.

Перед выполнением новой команды чтения или записи необходимо убедиться, что CMD = 0 и SYNC\_DONE = 1.

## 24.2.3 Счетчик реального времени

RWC содержит счетчик реального времени, функционирующий в любом режиме работы микросхемы пока на вывод VBAT подается напряжение питания. Счетчик тактируется сигналом clk\_1Hz с периодом в одну секунду.



Для получения текущего значения счетчика времени можно считать внутренний регистр TIME. Во избежание считывания неверных данных во время обновления регистра TIME, чтение необходимо последовательно повторить два раза. Если считанные значения отличаются, необходимо выполнить чтение еще раз.

Альтернативный метод – использование теневого регистра RWC\_STR. После «холодного» сброса микросхемы или выхода из режима STANDBY запускается процесс синхронизации внутреннего регистра TIME и теневого регистра. По завершению синхронизации устанавливается флаг SYNC\_DONE в регистре RWC\_CTRL.

После синхронизации регистр RWC\_STR содержит актуальное значение времени, инкрементируемое одновременно с регистром TIME.

Запись регистра TIME рекомендуется производить не раньше, чем через ~1 мкс после секундного строба RWC\_TICK. После изменения регистра TIME необходимо запустить процесс синхронизации теневого регистра путем записи 1 в бит RESYNC.

#### 24.2.4 Подстройка частоты секундного строба

Для коррекции точности секундного счетчика предусмотрена подстройка частоты секундного строба clk\_1Hz. Подстройка не зависит от выбранного осциллятора LFI/LFE, однако больше применима при работе от осциллятора LFE за счет его более высокой точности.

Подстройка частоты сигнала clk\_1Hz выполняется с помощью параметра TRIM\_1HZ регистра RWC\_TRIM. Для применения параметра необходимо выполнить запись в регистр RWC\_TRIMLOAD (записываемые данные не важны).

Значение подстройки частоты определяется по следующей таблице.

**Таблица 24.2. Подстройка частоты осцилляторов**

TRIM_*	Коррекция, ppm	Описание
0x3FF	512.0	Увеличение длительности секунды на 1024 такта каждые 64 секунды
0x3FE	511.5	Увеличение длительности секунды на 1023 такта каждые 64 секунды
...	...	...
0x001	1.0	Увеличение длительности секунды на 2 такта каждые 64 секунды
0x000	0.5	Увеличение длительности секунды на 1 такт каждые 64 секунды
0x7FF	0.0	Нет подстройки, значение по умолчанию
0x7FE	-0.5	Уменьшение длительности секунды на 1 такт каждые 64 секунды
...	...	...
0x401	-511.0	Уменьшение длительности секунды на 1022 такта каждые 64 секунды
0x400	-511.5	Уменьшение длительности секунды на 1023 такта каждые 64 секунды

## 24.3 Внутренние регистры RWC

В следующей таблице перечислены внутренние регистры RWC.

**Таблица 24.3. Перечень внутренних регистров RWC**

Адрес (RSEL)	Название регистра	Описание	Тип доступа	Исходное состояние
0x0	-	Резерв	RO	0x0000 0000
0x1	TICKGEN	Регистр для тестирования	RW	0x0000 0000
0x2	TRIMLOAD	Запись в этот регистр применяет значение подстройки в регистре TRIM	WO	0x0000 0000
0x3	TIME	Регистр текущего значения счетчика времени	RW	0x0000 0000
0x4	ALARM	Регистр времени пробуждения	RW	0x0000 0000
0x5	TRIM	Регистр подстройки осцилляторов	RW	0x8401 9800
0x6	CONFIG	Конфигурационный регистр	RW	0x8000 6021
0x7	GPR	Регистр общего назначения	RW	0x0000 0000
0x8	WAKECFG	Регистр настройки контроллера пробуждения	RW	0x0000 0000
0x9-0xF	-	Резерв	RO	0x0000 0000

### 24.3.1 Регистр TICKGEN

**Таблица 24.4. Описание регистра TICKGEN**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:21	-	RO	0	Резерв
20:0	TICKGEN	RW	0	Поле предназначено для тестирования

### 24.3.2 Регистр TRIMLOAD

**Таблица 24.5. Описание регистра TRIMLOAD**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:1	-	RO	0	Резерв
0	TRIMLOAD	WO	0	Запись в этот регистр любого значения применяет значение подстройки в регистре TRIM

### 24.3.3 Регистр TIME

**Таблица 24.6. Описание регистра TIME**

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:0	TIME	RW	0	Запись устанавливает значение счетчика времени. Чтение возвращает текущее значение счетчика.

## 24.3.4 Регистр ALARM

Таблица 24.7. Описание регистра ALARM

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:0	ALARM_TIME	RW	0	Значение времени пробуждения, сравниваемое с регистром TIME

## 24.3.5 Регистр TRIM

Таблица 24.8. Описание регистра TRIM

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:27	-	RO	0x10	Резерв
26:25	WAKE_REASON	RO	0x2	Поле показывает причину выхода системы из режима SHUTDOWN: 0x0: пробуждение по регистру ALARM; 0x1: пробуждение по внешнему сигналу WKUP; 0x2: первичное включение питания; 0x3: резерв.
24:23	-	RO	0	Резерв
22	LFE_BYPASS	RW	0	Режим работы осциллятора LFE: 0x0: работа в режиме осциллятора совместно с внешним кварцевым резонатором; 0x1: работа в режиме буфера для внешнего КМОП сигнала на выводе XT132.
21:11	TRIM_LFI	RW	0x033	Подстройка частоты осциллятора LFI. Значение применяется только после записи регистра TRIMLOAD. После сброса значение автоматически обновляется из памяти OTP
10:0	TRIM_1HZ	RW	0x000	Подстройка частоты секундного строба clk_1Hz. Значение применяется только после записи регистра TRIMLOAD

## 24.3.6 Регистр CONFIG

Таблица 24.9. Описание регистра CONFIG

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31	-	RO	1	Резерв
30	SHUTDOWN_FORCE	RW	0	Установка этого бита приводит к принудительному переходу системы в режим SHUTDOWN. Не рекомендуется использовать.
29:16	-	RO	0	Резерв
15	WAKE_IN_EN	RW	0	Разрешение работы входа WKUP.
14	PL	RW	1	Бит устанавливается при первом включении питания. Сбрасывать нельзя.
13	PZ	RW	1	Бит устанавливается при первом включении питания. Сбрасывать нельзя.

Диапазон	Название	Тип доступа	Исходное состояние	Описание
12	ALARM_EN	RW	0	Разрешение прерывания RWC_ALARM по совпадению значений регистров TIME и ALARM.
11	-	RO	0	Резерв
10:6	CLKDIV	RW	0	Значение делителя тактового сигнала clk_32kHz и соответствующего выхода RTCCLK. Результирующая частота определяется таким образом: 0x0 – 0x15: 32 кГц / 2 <sup>CLKDIV</sup> ; 0x16 – 0x1F: резерв
5	-	RO	0	Резерв
4	OSC_SEL	RW	0	Выбор осциллятора: 0x0: RC-осциллятор LFI; 0x1: кварцевый осциллятор LFE
3:1	-	RW	0	Резерв
0	TIME_CLK_SEL	RW	1	Выбор сигнала для тактирования счетчика времени: 0x0: clk_32kHz; 0x1 clk_1Hz;

### 24.3.7 Регистр GPR

Регистр общего назначения. Сохраняет свое состояние в любом режиме работы микросхемы при наличии питания VBAT. Сброс регистра выполняется только при первичном включении питания VBAT.

Таблица 24.10. Описание регистра GPR

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:0	GPR	RW	0	Регистр общего назначения.

### 24.3.8 Регистр WAKECFG

Таблица 24.11. Описание регистра WAKECFG

Диапазон	Название	Тип доступа	Исходное состояние	Описание
31:17	-	RO	0	Резерв
16	WAKE_POL	RW	0	Полярность сигнала WKUP для генерирования прерывания: 0x0: активный высокий уровень; 0x1: активный низкий уровень
15:1	-	RO	0	Резерв
0	WAKE_EN	RW	0	Разрешение прерывания RWC WKUP

## 24.4 Регистры APB интерфейса

Доступ к регистрам возможен только 32-х разрядными словами.

## 24.4.1 Перечень регистров

Таблица 24.12. Регистры RWC

Условное обозначение	Название регистра	Исходное состояние	Тип доступа	Смещение
RWC_STR	Теневой регистр счетчика реального времени	0x0000 0000	RO	0x000
RWC_RWDATA	Регистр данных для чтения и записи внутренних регистров	0x0000 0000	RW	0x004
RWC_CTRL	Регистр управления и статуса	0x0000 0300	RW	0x008
RWC_INTCLR	Регистр сброса прерываний	0x0000 0000	WO	0x00C
RWC_INTEN	Регистр разрешения прерываний	0x0000 0000	RW	0x010
RWC_INTSTAT	Регистр статуса прерываний	0x0000 0000	RW	0x014

## 24.4.2 Регистр RWC\_STR

Таблица 24.13. Регистр RWC\_STR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	TIME	RO	0x0	Текущее значение счетчика реального времени.

## 24.4.3 Регистр RWC\_RWDATA

Таблица 24.14. Регистр RWC\_RWDATA

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA	RW	0x0	Значение, записываемое во внутренний регистр, либо считанное из внутреннего регистра.

## 24.4.4 Регистр RWC\_CTRL

Таблица 24.15. Регистр RWC\_CTRL

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	TICK_PULSE	RW	0x0	Используется для тестирования. Бит устанавливается по фронту сигнала RWC_TICK. Бит сбрасывается при записи 0. Запись 1 игнорируется.
30	-	RO	0x0	Резерв
29	REG_CLK_FORCE	RW	0x0	Используется для тестирования. Управление тактовым сигналом регистрового интерфейса RWC: 0x0: нормальная работа интерфейса; 0x1: установка уровня 1 на тактовом сигнале.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
28	CLK_DIS	RW	0x0	Управление выходом RTCCLK: 0x0: тактовый сигнал включен; 0x1: тактовый сигнал выключен.
27	-	RO	0x0	Резерв
26:25	WAKE_REASON	RO	0x2	После сброса в данное поле копируется значение WAKE_REASON из внутреннего регистра TRIM.
24:23	-	RO	0x0	Резерв
22	WKUP_INT_TYPE	RW	0x0	Тип сигнала (уровень/фронт) на входе WKUP для генерирования прерывания RWC_WKUP: 0x0: прерывание по уровню сигнала; 0x1: прерывание по фронту сигнала
21	WKUP_INT_POL	RW	0x0	Полярность сигнала на входе WKUP для генерирования прерывания RWC_WKUP: 0x0: прерывание по низкому уровню или спадающему фронту сигнала; 0x1: прерывание по высокому уровню или нарастающему фронту сигнала
20	WKUP_STATE	RO	0x0	Состояние входа WKUP
19:17	-	RO	0x0	Резерв
16	SYNC_DONE	RO	0x0	Флаг устанавливается при синхронизации значения теневого регистра RWC_STR с внутренним регистром TIME. Запись игнорируется.
15:11	-	RO	0x0	Резерв
10:8	PRESC	RW	0x3	Чтение и запись внутренних регистров RWC происходит по последовательному интерфейсу. Значение PRESC задает делитель для формирования тактовой частоты интерфейса: 0x0: 8 МГц; 0x1: 4 МГц; 0x2: 2 МГц; 0x3: 1 МГц; 0x4: 500 кГц; 0x5: 250 кГц; 0x6: 125 кГц; 0x7: 62.5 кГц Рекомендуемое значение частоты – 1 МГц.
7:4	RSEL	RW	0x0	Адрес внутреннего регистра RWC (Таблица 24.3).
3	RESYNC	WO	0x0	Запись 1 в это поле сбрасывает флаг SYNC_DONE, что приводит к повторной синхронизации значения теневого регистра RWC_TR. При чтении возвращается 0.
2	-	RO	0x0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
1:0	CMD	RW	0x0	Запись в это поле инициирует доступ к внутренним регистрам RWC: 0x1: запись данных из регистра RWC_RWDATA в регистр, определяемый полем RSEL; 0x2: чтение регистра, определяемого полем RSEL, в регистр RWC_RWDATA; 0x0, 0x3: резерв. Значение этого поля аппаратно сбрасывается при завершении доступа к внутреннему регистру RWC.

## 24.4.5 Регистр RWC\_INTCLR

Таблица 24.16. Регистр RWC\_INTCLR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0x0	Резерв
1	WKUP_CLR	WO	0x0	Запись 1 сбрасывает прерывание RWC_WKUP. Запись 0 игнорируется.
0	ALARM_CLR	WO	0x0	Запись 1 сбрасывает прерывание RWC_ALARM. Запись 0 игнорируется.

## 24.4.6 Регистр RWC\_INTEN

Таблица 24.17. Регистр RWC\_INTEN

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0x0	Резерв
1	WKUP_EN	RW	0x0	Разрешения прерывания RWC_WKUP
0	-	RO	0x0	Резерв

## 24.4.7 Регистр RWC\_INTSTAT

Таблица 24.18. Регистр RWC\_INTSTAT

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:2	-	RO	0x0	Резерв
1	WKUP_STAT	RW	0x0	Состояние прерывания RWC_WKUP. Бит может быть установлен программно для целей отладки.
0	ALARM_STAT	RO	0x0	Состояние прерывания RWC_ALARM.

## 25. MPC: КОНТРОЛЛЕР БЕЗОПАСНОСТИ ПАМЯТИ

### 25.1 Введение

Контроллеры MPC (Memory Protection Controller) предназначены для настройки уровня доступа (Secure/Non-secure) внутренней SRAM и Flash памяти, а также внешней памяти, подключаемой по интерфейсам QSPI и SMC.

Основные возможности MPC:

- настройка уровня доступа к блокам памяти;
- настройка ответа на нарушение безопасности;
- сбор отладочной информации при нарушении безопасности;
- программная блокировка доступа к памяти.

### 25.2 Описание функционирования

#### 25.2.1 Настройка доступа к памяти

Для настройки безопасности доступа память, подключенная к MPC, условно делится на блоки фиксированного размера. Размер блоков определяется конфигурацией конкретного контроллера MPC (Таблица 25.1). Количество блоков определяется общим размером памяти и размером одного блока.

**Таблица 25.1. Параметры MPC**

Тип памяти	Название MPC	Размер блока MPC	GATE_PRESENT
SRAMn, n = 0,1,2,3	SRAMn_MPC	256 байт	0
BKPSRAM	BKPSRAM_MPC	256 байт	1
Flash (основной раздел)	FLASH_MPC	8 Кбайт	1
Внешняя память QSPI	QSPI_MPC	64 Кбайт	1
Внешняя память SMC	SMC_MPC	64 Кбайт	1

В MPC реализована таблица LUT, содержащая N строк по 32 бита. Каждый бит таблицы содержит настройку безопасности доступа (Secure/Non-secure) для одного блока памяти. Если общее количество блоков меньше 32, то  $N = 1$ .

Для чтения всей таблицы LUT необходимо:

1. Включить автоинкремент адреса установкой бита AUTOINCR\_EN = 1 (при  $N > 1$ ).
2. Определить размер таблицы N из регистра MPC\_BLK\_MAX:  $N = \text{BLK\_MAX} + 1$ .
3. Установить начальный адрес таблицы 0x0 в регистре MPC\_BLK\_IDX.
4. Выполнить чтение регистра MPC\_BLK\_LUT N раз.



Для записи всей таблицы LUT необходимо:

1. Включить автоинкремент адреса установкой бита AUTOINCR\_EN = 1 (при  $N > 1$ ).
2. Определить размер таблицы N из регистра MPC\_BLK\_MAX:  $N = \text{BLK\_MAX} + 1$ .
3. Установить начальный адрес таблицы 0x0 в регистре MPC\_BLK\_IDX.
4. Выполнить запись настроек безопасности в регистр MPC\_BLK\_LUT N раз для заполнения таблицы.

Для чтения и последующего изменения одной строки таблицы LUT необходимо:

1. Выключить автоинкремент адреса сбросом бита AUTOINCR\_EN = 0.
2. Установить требуемый адрес таблицы в регистре MPC\_BLK\_IDX.
3. Считать строку таблицы из регистра MPC\_BLK\_LUT.
4. Записать модифицированную строку таблицы в регистр MPC\_BLK\_LUT.

При несовпадении атрибута безопасности входящей транзакции с настройкой безопасности адресуемого блока памяти контроллер MPC блокирует транзакцию и выполняет следующие действия:

- Если бит SEC\_RESP в регистре MPC\_CTRL установлен, то на интерфейсе АНВ возвращается ошибка шины (HRESP = 1).
- Если бит SEC\_RESP в регистре MPC\_CTRL сброшен, то транзакция обрабатывается как RAZ/WI: при чтении возвращаются нули, запись игнорируется.
- Устанавливается прерывание MPC, если это разрешено в регистре MPC\_INT\_EN. Прерывания от всех контроллеров MPC объединены в одну линию прерывания NVIC. Для определения, какой из блоков MPC вызвал прерывание, используется регистр SPCTR\_SECMPCINTSTAT.
- Если прерывание разрешено, то в регистры MPC\_INT\_INFO\* заносится информация о транзакции, вызвавшей нарушение безопасности.

## 25.2.2 Блокировка доступа к памяти

В контроллерах MPC, у которых параметр GATE\_PRESENT = 1 (Таблица 25.1), реализована функция программной блокировки доступа к памяти, подключенной к соответствующему контроллеру.

Для включения блокировки необходимо установить бит GATE\_REQ в регистре MPC\_CTRL и дождаться установки бита подтверждения GATE\_ACK. После этого любое обращение к памяти блокируется, т.е. при чтении возвращаются нули, а запись игнорируется.

Для выключения блокировки необходимо сбросить бит GATE\_REQ в регистре MPC\_CTRL и дождаться сброса бита подтверждения GATE\_ACK.

## 25.3 Регистры

### 25.3.1 Перечень регистров MPC

Перечень регистров MPC приведен в следующей таблице.

**Таблица 25.2. Регистры MPC**

Условное обозначение	Описание	Исходное состояние	Смещение
MPC_CTRL	Регистр конфигурации и управления	0x00000100	0x000
-	Резерв	0x00000000	0x004 – 0x00C
MPC_BLK_MAX	Регистр максимального адреса таблицы LUT	*	0x010
MPC_BLK_CFG	Регистр статуса и размера блоков	*	0x014
MPC_BLK_IDX	Регистр адреса для доступа к таблице LUT	0x00000000	0x018
MPC_BLK_LUT	Регистр для доступа к таблице LUT	0x00000000	0x01C
MPC_INT_STAT	Регистр статуса прерывания	0x00000000	0x020
MPC_INT_CLEAR	Регистр сброса прерывания	0x00000000	0x024
MPC_INT_EN	Регистр разрешения прерывания	0x00000000	0x028
MPC_INT_INFO1	Регистр отладочной информации 1	0x00000000	0x02C
MPC_INT_INFO2	Регистр отладочной информации 2	0x00000000	0x030
MPC_INT_SET	Регистр установки прерывания (для отладки)	0x00000000	0x034
-	Резерв	0x00000000	0x038 – 0xFFC
MPC_PIDR4	Регистр Peripheral ID 4	0x00000004	0xFD0
MPC_PIDR5	Регистр Peripheral ID 5	0x00000000	0xFD4
MPC_PIDR6	Регистр Peripheral ID 6	0x00000000	0xFD8
MPC_PIDR7	Регистр Peripheral ID 7	0x00000000	0xFDC
MPC_PIDR0	Регистр Peripheral ID 0	0x00000060	0xFE0
MPC_PIDR1	Регистр Peripheral ID 1	0x000000B8	0xFE4
MPC_PIDR2	Регистр Peripheral ID 2	0x0000000B	0xFE8
MPC_PIDR3	Регистр Peripheral ID 3	0x00000000	0xFEC
MPC_CIDR0	Регистр Component ID 0	0x0000000D	0xFF0
MPC_CIDR1	Регистр Component ID 1	0x000000F0	0xFF4
MPC_CIDR2	Регистр Component ID 2	0x00000005	0xFF8
MPC_CIDR3	Регистр Component ID 3	0x000000B1	0xFFC

### 25.3.2 Регистр MPC\_CTRL

Формат регистра MPC\_CTRL приведен в следующей таблице.

**Таблица 25.3. Поля регистра MPC\_CTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	SEC_LOCK	RW	0	При установке этого бита блокируется изменение регистров MPC_CTRL, MPC_BLK_LUT, MPC_INT_SET, MPC_INT_EN. После установки бита запись 0 игнорируется. Очистка бита возможна только по аппаратному сбросу.
30:9	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
8	AUTOINCR_EN	RO/RW	0/1	При BLK_MAX > 0 –настройка автоинкремента адреса таблицы при доступе к регистру MPC_BLK_LUT: 0x0: автоинкремент выключен; 0x1: автоинкремент включен. Исходное состояние – 1. Если BLK_MAX = 0, то бит имеет исходное состояние 0 и тип доступа RO.
7	GATE_ACK	RO	0	Установка этого бита означает подтверждение блокировки доступа к памяти
6	GATE_REQ	RW/RO	0	При значении параметра GATE_PRESENT = 1 поле имеет тип доступа RW. Запись 1 инициирует блокировку доступа к памяти через MPC. Запись 0 снимает блокировку. При заблокированном доступе при чтении памяти возвращаются нули, запись игнорируется. При значении параметра GATE_PRESENT = 0 поле имеет тип доступа RO. Функция блокировки доступа к памяти не реализована.
5	-	RO	0	Резерв
4	SEC_RESP	RW	0	Настройка ответа при нарушении безопасности: 0x0: RAZ-WI – чтение возвращает нули, запись игнорируется; 0x1: ошибка шины
3:0	-	RO	0	Резерв

### 25.3.3 Регистр MPC\_BLK\_MAX

Формат регистра MPC\_BLK\_MAX приведен в следующей таблице.

**Таблица 25.4. Поля регистра MPC\_BLK\_MAX**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	BLK_MAX	RO	*	Поле содержит максимальный адрес таблицы LUT, нумерация адреса начинается с 0. Исходное состояние определяется конфигурацией MPC.

### 25.3.4 Регистр MPC\_BLK\_CFG

Формат регистра MPC\_BLK\_CFG приведен в следующей таблице.

**Таблица 25.5. Поля регистра MPC\_BLK\_CFG**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	INIT_N_DONE	RO	1	Статус инициализации таблицы LUT: 0x0: инициализация завершена; 0x1: инициализация в процессе.
30:4	-	RO	0	Резерв

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
3:0	BLK_SIZE	RO	*	Размер блока памяти = $1 \ll (\text{BLK\_SIZE} + 5)$ : 0x0: 32 байта; 0x1: 64 байта; ... 0xF: 1 Мбайт. Исходное состояние определяется конфигурацией MPC.

### 25.3.5 Регистр MPC\_BLK\_IDX

Формат регистра MPC\_BLK\_IDX приведен в следующей таблице.

**Таблица 25.6. Поля регистра MPC\_BLK\_IDX**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:12	-	RO	0	Резерв
11:0	BLK_IDX	RW/RO	0	Текущий адрес таблицы LUT. При BLK_MAX > 0 имеет тип доступа RW. Поле также инкрементируется при доступе к регистру MPC_BLK_LUT при включенном автоинкременте адреса. При BLK_MAX = 0 тип доступа – RO.

### 25.3.6 Регистр MPC\_BLK\_LUT

Формат регистра MPC\_BLK\_LUT приведен в следующей таблице.

**Таблица 25.7. Поля регистра MPC\_BLK\_LUT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	BLK_LUT	RW	0	Регистр для записи/чтения строки таблицы LUT. Адрес строки определяется регистром MPC_BLK_IDX. Каждый бит $i$ ( $i = 0, \dots, 31$ ) отвечает за настройку доступа к блоку памяти с номером $(\text{BLK\_IDX} * 32 + i)$ : 0x0: разрешен только Secure доступ; 0x1: разрешен только Non-secure доступ

### 25.3.7 Регистр MPC\_INT\_STAT

Формат регистра MPC\_INT\_STAT приведен в следующей таблице.

**Таблица 25.8. Поля регистра MPC\_INT\_STAT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	INT_STAT	RO	0	Статус прерывания

### 25.3.8 Регистр MPC\_INT\_CLEAR

Формат регистра MPC\_INT\_CLEAR приведен в следующей таблице.

**Таблица 25.9. Поля регистра MPC\_INT\_CLEAR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	INT_CLEAR	WO	0	Запись 1 сбрасывает прерывание. Запись 0 игнорируется.

### 25.3.9 Регистр MPC\_INT\_EN

Формат регистра MPC\_INT\_EN приведен в следующей таблице.

**Таблица 25.10. Поля регистра MPC\_INT\_EN**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	INT_EN	RW	1	Разрешение прерывания: 0x0: прерывание запрещено; 0x1: прерывание разрешено.

### 25.3.10 Регистр MPC\_INT\_INFO1

Регистр предназначен для отладки нарушений безопасности при доступе к памяти. Информация в регистре обновляется только при установке прерывания.

Формат регистра MPC\_INT\_INFO1 приведен в следующей таблице.

**Таблица 25.11. Поля регистра MPC\_INT\_INFO1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	HADDR	RO	0	Адрес транзакции, вызвавшей нарушение безопасности и установку прерывания.

### 25.3.11 Регистр MPC\_INT\_INFO2

Регистр предназначен для отладки нарушений безопасности при доступе к памяти. Информация в регистре обновляется только при установке прерывания.

Формат регистра MPC\_INT\_INFO2 приведен в следующей таблице.

**Таблица 25.12. Поля регистра MPC\_INT\_INFO2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:18	-	RO	0	Резерв
17	CFG_NS	RO	0	Настройка безопасности блока памяти, при обращении к которому произошло нарушение безопасности.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
16	HNONSEC	RO	0	Атрибут безопасности транзакции, вызвавшей нарушение безопасности и установку прерывания
15:0	HMASTER	RO	0	Идентификатор master-устройства транзакции, вызвавшей нарушение безопасности и установку прерывания

### 25.3.12 Регистр MPC\_INT\_SET

Регистр предназначен для программной установки прерывания MPC для целей отладки.

Формат регистра MPC\_INT\_SET приведен в следующей таблице.

**Таблица 25.13. Поля регистра MPC\_INT\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв
0	INT_SET	WO	0	Запись 1 устанавливает прерывание MPC.

## 26. ОТП: ОДНОКРАТНО ПРОГРАММИРУЕМАЯ ПАМЯТЬ

### 26.1 Введение

Микросхема содержит блок однократно-программируемой (ОТП) памяти для хранения следующих данных:

- данные крипто-ускорителя CryptoCell;
- системная конфигурация;
- пользовательские данные.

ОТП память имеет следующие особенности:

- организация памяти – 256 слов по 40 бит;
- используемые методы коррекции ошибок: ECC, BIP, дублирование данных;
- объем пользовательских данных – 640 байт в single-cell режиме, 320 байт в double-cell режиме;
- поддержка выполнения инструкций из пользовательской области памяти в том числе в режиме execute-only;
- встроенный источник IPS для формирования необходимого напряжения питания.

Исходное состояние всех ячеек памяти – 0. При программировании (записи) памяти состояние ячейки может быть изменено на 1. После записи изменить состояние 1 на 0 невозможно.

### 26.2 Описание функционирования

#### 26.2.1 Структурная схема

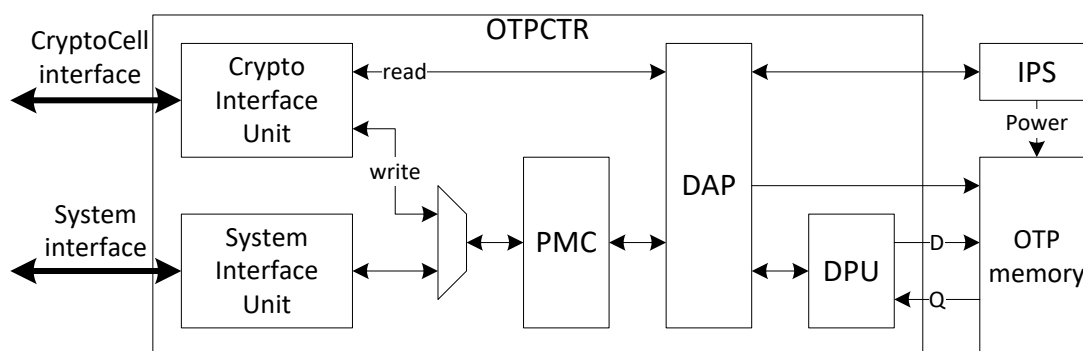


Рисунок 26.1. Структурная схема подсистемы ОТП

На Рисунок 26.1 показана структурная схема подсистемы ОТП памяти. Подсистема состоит из компонентов:

- память OTP memory.
- Интегрированный источник питания IPS. Генерирует необходимые уровни напряжения для чтения и программирования памяти.

- Контроллер ОTRCTR. Выполняет конфигурацию IPS, ОTR и предоставляет доступ к памяти. Состоит из под-блоков:
  - Интерфейс CryptoCell (Crypto Interface Unit): служит для обработки запросов чтения и записи памяти от блока CryptoCell. Содержит логику фильтрации на основе Life Cycle State (LCS).
  - Системный интерфейс (System Interface Unit): предоставляет доступ к регистрам PMC, DAP, DPU.
  - Device Access Port (DAP): содержит конфигурационные регистры IPS и ОTR.
  - Datapath Unit (DPU): содержит логику для выполнения низкоуровневых операций с памятью, а также схемы обнаружения и коррекции ошибок ECC, BRP.
  - Programmable Master Controller (PMC): реализует алгоритмы программирования и проверки памяти.

## 26.2.2 Интерфейсы

### 26.2.2.1 Интерфейс CryptoCell

Интерфейс CryptoCell предоставляет основной механизм для чтения и записи ОTR памяти. При этом блоки CryptoCell и Crypto Interface Unit совместно выполняют фильтрацию транзакций к памяти на основе LCS.

Настройка привилегированности и безопасности доступа, а также функции execute-only выполняется в регистре AO\_APB\_FILTERING блока CryptoCell.

Интерфейс предназначен только для чтения и записи памяти. Настройка всех параметров выполняется через системный интерфейс ОTRCTR. Процедуры чтения и записи описаны в пунктах 26.2.5 и 26.2.7 соответственно.

### 26.2.2.2 Интерфейс System

Системный интерфейс ОTRCTR предназначен для настройки временных параметров операций чтения и записи памяти, параметров источника питания IPS и режимов ОTR, а также для отладочного доступа к памяти.

Системный интерфейс ОTRCTR доступен только для привилегированных Secure обращений по базовому адресу 0x5003\_3000.

Доступ к содержимому памяти посредством системного интерфейса предназначен только для тестирования при LCS = CM и может быть заблокирован с помощью бита OTR\_SYS\_RW\_EN регистра CRYPTO\_HOST\_DCU\_EN0. При LCS = DM/SECURE/RMA бит OTR\_SYS\_RW\_EN всегда равен 0, и доступ к памяти через системный интерфейс ОTRCTR заблокирован.



## 26.2.3 Методы коррекции ошибок

### 26.2.3.1 ECC, BRP

Блок DPU в составе OTPCTR содержит схемы Error Correcting Code (ECC) и Bit Repair by Polarity (BRP). Данные схемы могут быть использованы для обнаружения и коррекции ошибок в пользовательской области OTP-памяти. Управление ECC и BRP производится битами ECCGEN, ECCDIS, BRPGEN, BRPDIS регистра OTPCTR\_DAP\_RQ2.

ECC и BRP не применяются для области памяти, используемой блоком CryptoCell.

### 26.2.3.2 Режим дублирования данных

OTP-память может использоваться в одном из двух режимов:

- single-cell – для хранения одного бита используется одна ячейка памяти;
- double-cell – для хранения одного бита используются две ячейки памяти.

Double-cell режим повышает надежность считываемых данных, но задействует вдвое больше памяти.

Область памяти, используемая блоком CryptoCell, а также системная конфигурация OTP\_CFG0/1 всегда считываются в double-cell режиме.

Системная конфигурация OTP\_CFG2-7 всегда должна считываться в single-cell режиме.

Пользовательская область памяти может использоваться в single-cell режиме либо в double-cell режиме. Режим включается параметром SHF\_REDUND в регистре OTPCTR\_DAP\_RQ0.

Запись всегда производится в single-cell режиме. При этом, если чтение будет выполняться в double-cell режиме, то каждый бит необходимо записать в две ячейки памяти. Для области данных CryptoCell и системной конфигурации OTP\_CFG0/1 дублирование записи через интерфейс CryptoCell выполняется аппаратно. Для пользовательской области памяти дублирование необходимо выполнить программно (см. пункт 26.2.7).

## 26.2.4 Структура памяти

В следующей таблице показана структура OTP-памяти с указанием фактических адресов памяти для каждой области (указаны адреса 32-х разрядных слов). Адресация по системному интерфейсу совпадает с фактической адресацией памяти. Также в таблице указаны:

- Значения LCS, при которых разрешены чтение и запись памяти по интерфейсу CryptoCell.
- Дублирование данных (REDUND): всегда включено (+), всегда выключено (-) или конфигурируется (Конф.).
- Использование ECC и BRP: всегда включено (+), всегда выключено (-) или конфигурируется (Конф.).

Неиспользуемые адреса зарезервированы. Чтение по ним возвращает нули, запись игнорируется.

**Таблица 26.1. Структура OTP памяти**

Адрес слова памяти	Назначение	LCS для чтения	LCS для записи	REDUND	ECC, BRP
0x00 - 0x26	CryptoCell	1	1	+	-
0x27	Системная конфигурация OTP_CFG0	Любое	CM	+	-
0x28 - 0x2B	CryptoCell	1	1	+	-
0x2C	Системная конфигурация OTP_CFG1	Любое	CM, DM <sup>4</sup>	+	+
0x2D - 0x7C	Первая часть пользовательской области (User area 1)	Любое	<sup>5</sup>	Конф. <sup>3</sup>	Конф. <sup>3</sup>
0x7D	Системная конфигурация OTP_CFG2	Любое	CM	-	+
0x7E	Системная конфигурация OTP_CFG3	Любое	CM	-	+
0x7F	Системная конфигурация OTP_CFG4	Любое	CM	-	+
0x80 - 0xAC	Резерв. Используется для дублирования данных по адресам 0x00 – 0x2C	CM <sup>2</sup>	CM <sup>2</sup>	-	-
0xAD - 0xFC	single-cell режим: вторая часть пользовательской области (User area 2). double-cell режим: дубликат первой части пользовательской области	Любое	<sup>5</sup>	-	Конф. <sup>3</sup>
0xFD	Системная конфигурация OTP_CFG5	Любое	CM	-	+
0xFE	Системная конфигурация OTP_CFG6	Любое	CM	-	+
0xFF	Системная конфигурация OTP_CFG7	Любое	CM	-	+

Примечания:

1. См. раздел 27 «CRYPTO: Крипто-акселератор CryptoCell».
2. Диапазон адресов памяти 0x80 - 0xAC доступен только через системный интерфейс OTPCTR при LCS = CM и OTP\_SYS\_RW\_EN = 1.
3. Использование REDUND, ECC и BRP определяется конфигурацией.
4. Альтернативный адрес загрузки может быть записан только один раз. Последующие операции записи игнорируются.
5. Запись пользовательской памяти возможна в любом состоянии LCS, если бит OTP\_USER\_WR\_LOCK = 0 (см. регистр CRYPTO\_HOST\_DCU\_EN0).

На следующем рисунке показано соответствие областей OTP памяти и адресов при обращении через интерфейс CryptoCell.

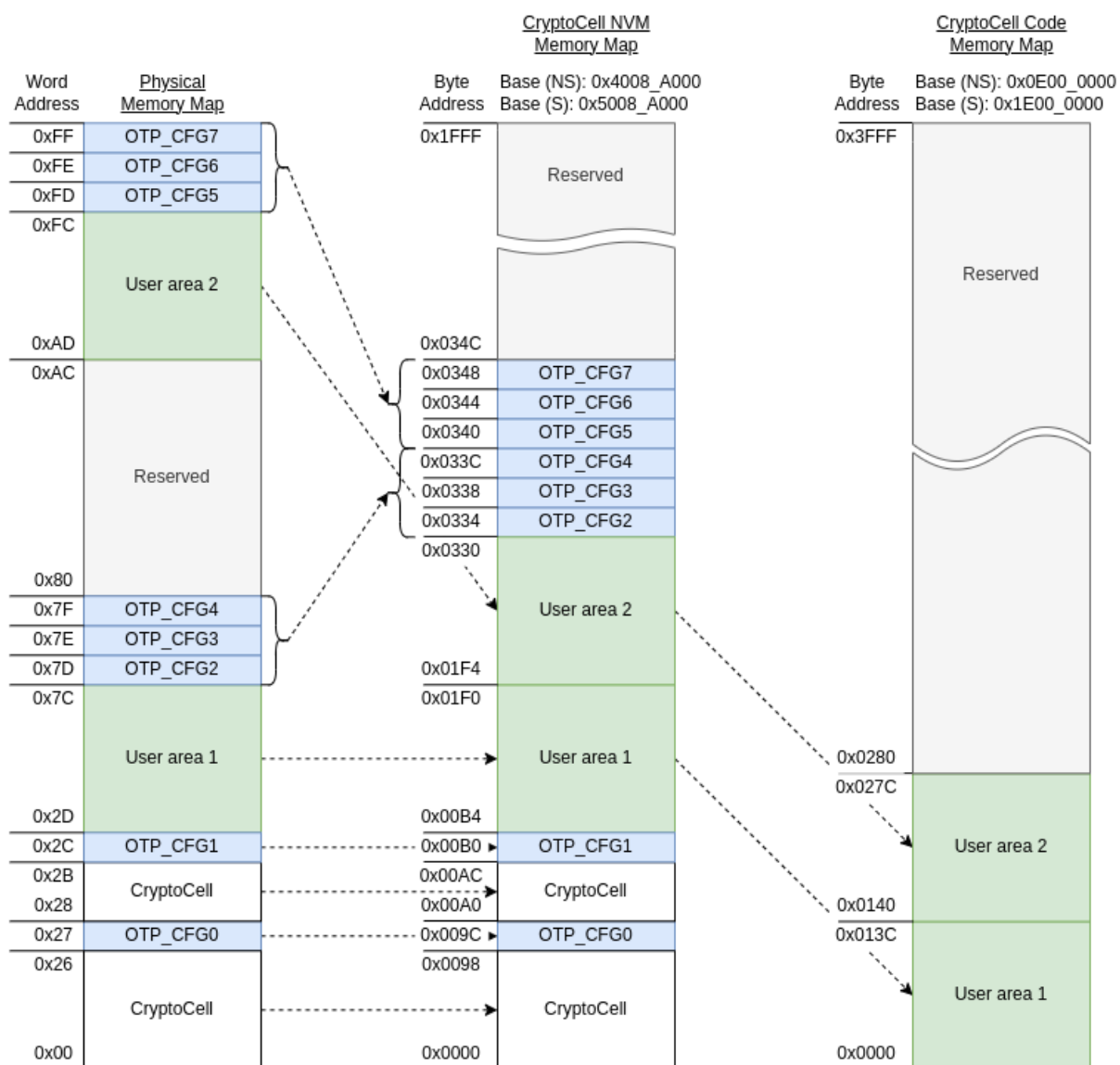


Рисунок 26.2. Адресация OTP памяти

## 26.2.5 Системная конфигурация

Память OTP содержит восемь 32-разрядных слов OTP\_CFG0 – OTP\_CFG7, предназначенных для хранения различных системных настроек.

Все параметры в каждом слове OTP\_CFG0 – OTP\_CFG7 должны записываться за одну транзакцию. Повторная запись не допускается.

В следующих таблицах приведены структура и назначение конфигурационных слов.

Таблица 2 Описание конфигурационного слова OTP\_CFG0

Номер бита	Название	Назначение
31:27	HFI_TRIM	Если APC_TRIM_DSBL=0, то после сброса данное значение автоматически загружается в регистр CLKCTR_HFITRIM

Номер бита	Название	Назначение
26:22	APC_VREF_VT	Если APC_TRIM_DSBL=0, то после сброса данное значение автоматически загружается в регистр PWRCTR_TRIM
21:19	APC_VREF_TT	Если APC_TRIM_DSBL=0, то после сброса данное значение автоматически загружается в регистр PWRCTR_TRIM
18:16	APC_VREF_IT	Если APC_TRIM_DSBL=0, то после сброса данное значение автоматически загружается в регистр PWRCTR_TRIM
15:8	RWC_LFI_TRIM	Если RWC_TRIM_DSBL=0, то после сброса данное значение автоматически загружается в регистр TRIM блока RWC
4	RWC_TRIM_DSBL	При установке бита отключается атозагрузка параметра RWC_LFI_TRIM в регистр TRIM блока RWC
3	APC_TRIM_DSBL	При установке бита отключается атозагрузка параметров APC_VREF_IT, APC_VREF_TT, APC_VREF_VT в регистр PWRCTR_TRIM и параметра HFI_TRIM в регистр CLKCTR_HFITRIM
2	FBIST_LOCK	Блокировка доступа к интерфейсу Flash BIST
1	GNSS_LOCK	Блокировка управления PPU домена PD_GNSS. При установке бита блокируется включение домена PD_GNSS
0	GMS_LOCK	Блокировка управления PPU домена PD_GMS. При установке бита блокируется включение домена PD_GMS

Таблица 3 Описание конфигурационного слова OTP\_CFG1

Номер бита	Название	Назначение
31:7	BOOT_ADDR	Адрес загрузки CPU при BOOT_ADDR_SEL=1
6:1	-	Резерв
0	BOOT_ADDR_SEL	При BOOT_ADDR_SEL=1 используется альтернативный адрес загрузки BOOT_ADDR. После установки бита BOOT_ADDR_SEL конфигурационное слово OTP_CFG1 блокируется для записи.

Таблица 4 Описание конфигурационного слова OTP\_CFG2

Номер бита	Название	Назначение
31:27	-	Резерв
26:16	JTM_TCAL	Параметр калибровки JTM
15:11	-	Резерв
10:0	JTM_WCAL	Параметр калибровки JTM

Таблица 5 Описание конфигурационного слова OTP\_CFG3

Номер бита	Название	Назначение
31:25	-	Резерв
24	JTM_CALS	Параметр калибровки JTM
23	-	Резерв
22:16	JTM_WTCALCON F	Параметр калибровки JTM
15:10	-	Резерв
9:0	JTM_WTCONF	Параметр калибровки JTM

Параметры калибровки JTM из OTP\_CFG2/3 автоматически загружаются в регистры JTM после сброса.

**Таблица 6 Описание конфигурационного слова OTP\_CFG4**

Номер бита	Название	Назначение
31:0	UID0	Уникальный идентификатор, биты [31:0]

**Таблица 7 Описание конфигурационного слова OTP\_CFG5**

Номер бита	Название	Назначение
31:0	UID1	Уникальный идентификатор, биты [64:32]

**Таблица 8 Описание конфигурационного слова OTP\_CFG6**

Номер бита	Название	Назначение
31:0	UID2	Уникальный идентификатор, биты [95:64]

**Таблица 9 Описание конфигурационного слова OTP\_CFG7**

Номер бита	Название	Назначение
31:0	UID3	Уникальный идентификатор, биты [127:96]

Уникальный идентификатор {UID3, UID2, UID1, UID0} доступен для чтения через регистры SYSINFO\_UID0-3.

## 26.2.6 Чтение

Для чтения OTP памяти предварительно необходимо выполнить настройки, перечисленные в следующей таблице.

**Таблица 26.10. Настройки OTPCTR для чтения памяти**

#	Параметр	Значение
1	IPS_EN	0x1
2	IPS_VRR	0x8
3	IPS_VRRTS	0x0
4	IPS_VRR_EN	0x1
5	SHF_IREF	0x2
6	SHF_SEL	0x1
7	SHF_CKDEL	0x00 или 0x1F
8	SHORT_READ_PW	Рассчитывается, исходя из длительности импульса tRP: 60 нс < tRP < 100 нс, при SHF_CKDEL = 0x00; 45 нс < tRP, при SHF_CKDEL = 0x1F;
9	SHF_REDUND, ECC_DIS, BRP_DIS	При чтении пользовательской области памяти установить в соответствии с используемыми методами коррекции ошибок

Примечания:

- после включения питания микросхемы и снятия сброса настройки из пунктов 1 - 8 уже соответствуют рекомендуемым, SHF\_REDUND = 0, ECC\_DIS = 0, BRP\_DIS = 0;
- после изменения параметра IPS\_VRR необходимо выждать минимум 7.0 мкс перед первым чтением памяти;

- после изменения параметра SHF\_IREF необходимо выждать минимум 670 нс перед первым чтением памяти.

Чтение OTP памяти через интерфейс CryptoCell возможно двумя способами:

- чтение всех разделов памяти через регистровый интерфейс блока CryptoCell по базовым адресам 0x4008\_A000 (Non-secure) и 0x5008\_A000 (Secure);
- чтение пользовательской части OTP памяти (в том числе исполнение программного кода) через интерфейс памяти блока CryptoCell по базовым адресам 0x0E00\_0000 (Non-secure) и 0x1E00\_0000 (Secure).

Поддерживаются только 32-х разрядные обращения.

## 26.2.7 Запись

Для записи OTP памяти предварительно необходимо выполнить настройку блоков DAP и PMC. В следующей таблице приведены рекомендуемые значения параметров.

**Таблица 26.11. Настройки OTPCTR для записи памяти**

#	Параметр	Значение
1	SHF_SEL	0x1
2	IPS_EN	0x1
3	IPS_VQQ	0x5
4	IPS_VPP	0x5
5	PROG_MODE0	0x4802
6	PROG_MODE1	0xE816
7	PROG_MODE2	0x7D12
8	PROG_MODE3	0x4C01
9	SHORT_READ_PW	Рассчитывается, исходя из длительности импульса tRP: 60 нс < tRP < 100 нс, при SHF_CKDEL = 0x00; 45 нс < tRP, при SHF_CKDEL = 0x1F;
10	LONG_READ_PW	0x0
11	BRP_CHECK_READ_PW, PRE_PROG_READ_PW, POST_PROG_READ_PW, POST_SOAK_READ_PW, COMPARE_READ_PW	0x0
12	READ_RECOVERY	0x0
13	PROG_RECOVERY	0x0
14	PROG_PW	Рассчитывается, исходя из длительности импульса tPP: 30 мкс < tPP < 40 мкс
15	SOAK_PW	Рассчитывается, исходя из длительности импульса tSPP: 120 мкс < tPP < 160 мкс
16	MODE_SETTING_TIME	Рассчитывается, исходя из задержки на установку режима: не менее 7.0 мкс

Запись OTP памяти через интерфейс CryptoCell осуществляется по базовым адресам 0x4008\_A000 (Non-secure) и 0x5008\_A000 (Secure). Поддерживаются только 32-х разрядные обращения.

Транзакция записи на шине АНВ завершается сразу, не дожидаясь завершения записи в OTP память, которая может занимать длительное время. Следующее обращение к памяти

(чтение или запись) должно выполняться только после завершения текущей операции записи. Для отслеживания статуса записи в память используется бит AIB\_FUSE\_PROG\_COMPLETED блока CryptoCell.

Статус выполненной операции записи доступен в поле CTRL\_STATUS регистра PMC\_CQCTRL.

Если записываемый адрес  $X$  пользовательской области памяти предполагается считывать в double-cell режиме, то необходимо выполнить запись того же слова данных по адресу  $(X + 0x80)$ , где  $X$  – адрес 32-х разрядного слова в памяти. Перед этим нужно убедиться, что проверочные биты BRP совпадают по обоим адресам.

### **26.2.8 Режим пониженного энергопотребления**

Для перевода ОТР памяти в режим пониженного энергопотребления необходимо установить бит SHF\_PD = 1. После этого чтение и запись памяти не доступны.

После выхода из данного режима путем записи SHF\_PD = 0 необходимо выждать не менее 670 нс перед выполнением чтения памяти.

## 26.3 Регистры

### 26.3.1 Перечень регистров OTPCTR

Перечень регистров OTPCTR приведен в следующей таблице.

**Таблица 26.12. Регистры OTPCTR**

Условное обозначение	Описание	Исходное состояние	Смещение
OTPCTR_DAP_DR	Регистр данных	0x00000000	0x000
OTPCTR_DAP_ER	Регистр проверочных данных ECC и BRP	0x00000000	0x020
OTPCTR_DAP_RQ0	Конфигурационный регистр RQ0, блок DAP	0x04094802	0x030
OTPCTR_DAP_RQ1	Конфигурационный регистр RQ1, блок DAP	0x011F0000	0x034
OTPCTR_DAP_RQ2	Конфигурационный регистр RQ2, блок DAP	0x00000008	0x038
OTPCTR_DAP_CQCTRL	Конфигурационный регистр CQCTRL, блок DAP	0x00000000	0x03C
OTPCTR_PMC_RQ0	Конфигурационный регистр RQ0, блок PMC	0x480F4802	0x0B0
OTPCTR_PMC_RQ1	Конфигурационный регистр RQ1, блок PMC	0x00004821	0x0B4
OTPCTR_PMC_RQ2	Конфигурационный регистр RQ2, блок PMC	0x02000422	0x0B8
OTPCTR_PMC_CQCTRL	Конфигурационный регистр CQCTRL, блок PMC	0x00800000	0x0BC
OTPCTR_DAP_CMD	Регистр инструкций, блок DAP	0x00000000	0x100
OTPCTR_PMC_CMD	Регистр инструкций, блок PMC	0x00000000	0x180

### 26.3.2 Регистр OTPCTR\_DAP\_DR

При  $OTP\_SYS\_RW\_EN = 1$  регистр предоставляет доступ ко внутреннему регистру DAP\_DR блока DAP.

При  $OTP\_SYS\_RW\_EN = 0$  запись в данный регистр игнорируется, при чтении возвращаются нули.

Формат регистра OTPCTR\_DAP\_DR приведен в следующей таблице.

**Таблица 26.13. Поля регистра OTPCTR\_DAP\_DR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	DATA	RW	0	Слово данных для программирования памяти либо результат чтения памяти

### 26.3.3 Регистр OTPCTR\_DAP\_ER

При  $OTP\_SYS\_RW\_EN = 1$  регистр предоставляет доступ ко внутреннему регистру DAP\_ER блока DAP.

При  $OTP\_SYS\_RW\_EN = 0$  запись в данный регистр игнорируется, при чтении возвращаются нули.

Формат регистра OTPCTR\_DAP\_ER приведен в следующей таблице.



Таблица 26.14. Поля регистра OTPCTR\_DAP\_ER

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:8	-	RO	0	Резерв
7:6	BRP	RW	0	Проверочные биты BRP
5:0	ECC	RW	0	Проверочные биты ECC

## 26.3.4 Регистр OTPCTR\_DAP\_RQ0

Конфигурационный регистр RQ0, блок DAP.

Формат регистра OTPCTR\_DAP\_RQ0 приведен в следующей таблице.

Таблица 26.15. Поля регистра OTPCTR\_DAP\_RQ0

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	IPS_VRRSWC	RW	0	Резерв (MRR[6]). Значение данного поля нельзя изменять
30	IPS_REF_BIAS_DIS	RW	0	Reference Bias Disable (MRR[30]). Значение данного поля нельзя изменять
29	IPS_EXT_CK_EN	RW	0	External Clock Enable (MRR[29]). Значение данного поля нельзя изменять
28:27	IPS_OSC_OUT	RW	0	Управление выходом OSC_OUT. Резерв. (MRR[28:27])
26	IPS_EN	RW	0x1	IPS Control Enable (MRR[0])
25	IPS_EXT_REF_EN	RW	0	External Reference Enable (MRR[25]). Значение данного поля нельзя изменять
24	IPS_VDD_DET_DIS	RW	0	Power Up Detector Disable (MRR[24]). Значение данного поля нельзя изменять
23	IPS_OSCVQQ	RW	0	Резерв (MRR[5]). Значение данного поля нельзя изменять
22	IPS_OSCVPP	RW	0	Резерв (MRR[4]). Значение данного поля нельзя изменять
21:19	IPS_VPP	RW	0x1	Настройка уровня напряжения VPP (MRR[21:19])
18:16	IPS_VQQ	RW	0x1	Настройка уровня напряжения VQQ (MRR[18:16])
15	IPS_CP_EN	RW	0	Включение схемы накачки заряда (MRR[2])
14	IPS_VRR_EN	RW	0x1	Включение регулятора напряжения VRR (MRR[1])
13:12	IPS_VRRTS	RW	0	Настройка Temperature Slope для напряжения VRR (MRR[13:12])
11:8	IPS_VRR	RW	0x8	Настройка уровня напряжения VRR (MRR[11:8])
7	IPS_LD_CP_EN	RW	0	Low Drive Charge Pump Enable (MRR[3]). Значение данного поля не рекомендуется изменять
6	-	RO	0	Резерв
5	SHF_REDUND	RW	0	Управление режимом дублирования данных для пользовательской области памяти (MR[4]): 0x0: single-cell режим, дублирование выключено; 0x1: double-cell режим, дублирование включено

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
4:0	SHF_IREF	RW	0x2	Настройка опорного тока для чтения (IREF, MR[15:11]). Значения 0x00 и 0x10 используются только для тестирования

### 26.3.5 Регистр OTPCTR\_DAP\_RQ1

Конфигурационный регистр RQ1, блок DAP.

Формат регистра OTPCTR\_DAP\_RQ1 приведен в следующей таблице.

**Таблица 26.16. Поля регистра OTPCTR\_DAP\_RQ1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31	-	RO	0	Резерв
30:28	IPS_RESERVED	RW	0	Резерв ({MRR[31], MRR[23:22]}). Значение данного поля нельзя изменять
27:25	IPS_VREFLVL	RW	0	Резерв ({MRR[15:14], MRR[7]}). Значение данного поля нельзя изменять
24	IPS_WE_CK	RW	0x1	Charge Pump Synch Input (MRR[26]). Значение данного поля не рекомендуется изменять
23:21	-	RO	0	Резерв
20:16	SHF_CKDEL	RW	0x1F	Управление режимом чтения (MR[10:6]): 0x00: внешнее управление; 0x1F: управление внутренним таймером; 0x01 - 0x1E: резерв
15:12	-	RO	0	Резерв
11	SHF_OTP_RTST	RW	0	OTP IRR Cell Current Test Mode (MR[3]). Используется только для тестирования
10:9	SHF_AROM	RW	0	Адрес тестовой строки памяти (MR[2:1]). Используется только для тестирования
8	SHF_ROMEN	RW	0	Включение тестовых строк памяти (MR0). Используется только для тестирования
7:0	-	RO	0	Резерв

### 26.3.6 Регистр OTPCTR\_DAP\_RQ2

Конфигурационный регистр RQ2, блок DAP.

Формат регистра OTPCTR\_DAP\_RQ2 приведен в следующей таблице.

**Таблица 26.17. Поля регистра OTPCTR\_DAP\_RQ2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:27	-	RO	0	Резерв
26:24	DPU_MBPC	RW	0	Управление количеством программируемых бит. Значение данного поля не рекомендуется изменять
23:22	DPU_MUXQ	RW	0	Управление мультиплексором данных. Значение данного поля не рекомендуется изменять

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
21	DPU_PASS	RW	0	Управление блоком Datapath Unit. Значение данного поля не рекомендуется изменять
20	DPU_BRPGEN	RW	0	Управление генерированием проверочных бит BRP при записи: 0x0: выключено; 0x1: включено
19	DPU_BRPDIS	RW	0	Управление коррекцией данных с помощью BRP при чтении: 0x0: коррекция выполняется; 0x1: коррекция не выполняется
18	DPU_ECCTST	RW	0	Проверка ECC. Используется для тестирования
17	DPU_ECCGEN	RW	0	Управление генерированием проверочных бит ECC при записи памяти: 0x0: выключено; 0x1: включено
16	DPU_ECCDIS	RW	0	Управление коррекцией данных с помощью ECC при чтении: 0x0: коррекция выполняется; 0x1: коррекция не выполняется
15:4	-	RO	0	Резерв
3	SHF_SEL	RW	0x1	Управление сигналом выбора памяти: 0x0: память деактивирована, чтение и запись не выполняются; 0x1: память активирована
2	-	RO	0	Резерв
1	SHF_PD	RW	0	Управление режимом памяти: 0x0: рабочий режим; 0x1: режим пониженного энергопотребления
0	-	RO	0	Резерв

### 26.3.7 Регистр OTPCTR\_DAP\_CQCTRL

При OTP\_SYS\_RW\_EN = 1 регистр предоставляет доступ ко внутренним регистрам DAP\_OAR, DAP\_FP и DAP\_DCSR блока DAP.

При OTP\_SYS\_RW\_EN = 0 запись в данный регистр игнорируется, при чтении возвращаются нули.

Формат регистра OTPCTR\_DAP\_CQCTRL приведен в следующей таблице.

**Таблица 26.18. Поля регистра OTPCTR\_DAP\_CQCTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:24	DAP_DCSR	RW	0	Поле предоставляет доступ ко внутреннему регистру DAP_DCSR блока DAP
23:16	DAP_FP	RO	0	Поле предоставляет доступ ко внутреннему регистру DAP_FP блока DAP
15:0	DAP_OAR	RW	0	Поле предоставляет доступ ко внутреннему регистру DAP_OAR блока DAP

### 26.3.8 Регистр OTPCTR\_PMC\_RQ0

Конфигурационный регистр RQ0, блок PMC.

Формат регистра OTPCTR\_PMC\_RQ0 приведен в следующей таблице.

**Таблица 26.19. Поля регистра OTPCTR\_PMC\_RQ0**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	PROG_MODE1	RW	0x480F	Режим блока DAP при выполнении фазы PGM операции PROG. Рекомендуемое значение - 0xE816
15:0	PROG_MODE0	RW	0x4802	Режим блока DAP, устанавливаемый при успешном завершении операции PROG. Рекомендуемое значение - 0x4802

### 26.3.9 Регистр OTPCTR\_PMC\_RQ1

Конфигурационный регистр RQ1, блок PMC.

Формат регистра OTPCTR\_PMC\_RQ1 приведен в следующей таблице.

**Таблица 26.20. Поля регистра OTPCTR\_PMC\_RQ1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	PROG_MODE3	RW	0	Режим блока DAP при выполнении фаз PROG_BRP и PROG_COMP. Рекомендуемое значение - 0x4C01
15:0	PROG_MODE2	RW	0x4821	Режим блока DAP при выполнении фазы VERIFY операции PROG. Рекомендуемое значение - 0x7D12

### 26.3.10 Регистр OTPCTR\_PMC\_RQ2

Конфигурационный регистр RQ2, блок PMC.

Формат регистра OTPCTR\_PMC\_RQ2 приведен в следующей таблице.

**Таблица 26.21. Поля регистра OTPCTR\_PMC\_RQ2**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:30	-	RO	0	Резерв
29:24	DAP_ADDR	RW	0x02	Идентификатор блока DAP = 0x2. Значение данного поля нельзя изменять
23:22	MODE_SETTING_TIME	RW	0	Время установки режима: 0x0: 128 тактов; 0x1: 256 тактов; 0x2: 512 тактов; 0x3: 1024 такта

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
21:19	SOAK_PW	RW	0	Длительность импульса при выполнении процедуры SOAK, tSPP: 0x0: 512 тактов; 0x1: 1024 такта; 0x2: 2048 тактов; 0x3: 4096 тактов; 0x4: 8192 такта; 0x5: 16384 такта; 0x6: 32768 тактов; 0x7: 65536 тактов
18:16	PROG_PW	RW	0	Длительность импульса программирования, tPP: 0x0: 64 такта; 0x1: 128 тактов; 0x2: 256 тактов; 0x3: 512 тактов; 0x4: 1024 такта; 0x5: 2048 тактов; 0x6: 4096 тактов; 0x7: 8192 такта
15:14	PROG_RECOVERY	RW	0	Время восстановления после программирования, tPR: 0x0: 64 такта; 0x1: 128 тактов; 0x2: 256 тактов; 0x3: 512 тактов
13	READ_RECOVERY	RW	0	Время восстановления после чтения, tRR: 0x0: 1 такт; 0x1: 2 такта
12	COMPARE_READ_PW	RW	0	Длительность импульса чтения при сравнении: 0x0: короткий импульс; 0x1: длинный импульс
11	POST_SOAK_READ_PW	RW	0	Длительность импульса чтения при выполнении процедуры SOAK: 0x0: короткий импульс; 0x1: длинный импульс
10	POST_PROG_READ_PW	RW	0x1	Длительность импульса чтения после программирования: 0x0: короткий импульс; 0x1: длинный импульс
9	PRE_PROG_READ_PW	RW	0	Длительность импульса чтения перед программированием: 0x0: короткий импульс; 0x1: длинный импульс
8	BRP_CHECK_READ_PW	RW	0	Длительность импульса чтения для проверки BRP: 0x0: короткий импульс; 0x1: длинный импульс
7:4	LONG_READ_PW	RW	0x2	Длительность короткого импульса чтения. Значение N данного поля соответствует импульсу длительностью $2N + 1$ тактов
3:0	SHORT_READ_PW	RW	0x2	Для операции PROG - длительность короткого импульса чтения. Значение N данного поля соответствует импульсу длительностью $N + 11$ тактов. Длительность импульса чтения при обращении через интерфейс CryptoCell. Значение N соответствует импульсу длительностью $N + 1$ тактов.

### 26.3.11 Регистр OTPCTR\_PMC\_CQCTRL

Конфигурационный регистр CQCTRL, блок PMC. Поля данного регистра не рекомендуется изменять.

Формат регистра OTPCTR\_PMC\_CQCTRL приведен в следующей таблице.

**Таблица 26.22. Поля регистра OTPCTR\_PMC\_CQCTRL**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:24	CTRL_STATUS	RW	0	При записи биты CTRL_STATUS[3:0] отвечают за выбор функции, выполняемой блоком PMC: 0x0 - 0x7: сброс функции; 0x8: BOOT; 0x9: BIST; 0xA: PROG; 0xB - 0xF: резерв Биты CTRL_STATUS[7:4] отвечают за выбор статусного флага. При записи данное поле должно быть равно 0. При чтении поле CTRL_STATUS отображает различную статусную информацию блока PMC. CTRL_STATUS[7:6] - статус выполняемой операции: 0x0: резерв; 0x1: операция выполнена; 0x2: PMC не выполняет операцию; 0x3: PMC занят выполнением операции. CTRL_STATUS[5:4] - результат выполнения операции: 0x0: успешно; 0x1: операция завершилась ошибкой - для PROG превышен лимит перезаписи при выполнении процедуры SOAK; 0x2: операция PROG завершилась ошибкой - несовпадение при сравнении. Для BIST и BOOT - резерв; 0x3: резерв. CTRL_STATUS[3:0] - младшие 4 бита счетчика soak (для PROG); младшие 4 бита счетчика адреса (для BIST); резерв (для BOOT).
23:16	DFSR	RO	0x80	Номер бита поля CTRL_STATUS для вывода статусного флага
15	-	RO	0	Резерв
14	CQ_AINC	RW	0	Инкремент адреса: 0x0: выключен; 0x1: включен
13:10	CQ_LIMITn	RW	0	Лимит счетчика перезаписи при выполнении процедуры SOAK
9:5	CQ_EXITn	RW	0	Точка выхода в алгоритме PROG
4:0	CQ_ENTRY	RW	0	Точка входа в алгоритме PROG

### 26.3.12 Регистр OTPCTR\_DAP\_CMD

При OTP\_SYS\_RW\_EN = 1 регистр позволяет запускать инструкции в блоке DAP. При OTP\_SYS\_RW\_EN = 0 регистр зарезервирован.

Формат регистра OTPCTR\_DAP\_CMD приведен в следующей таблице.

**Таблица 26.23. Поля регистра OTPCTR\_DAP\_CMD**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	-	RO	0	Резерв
6:0	INSTR	WO	0	Запись в это поле инициирует выполнение инструкции в блоке DAP. Коды инструкций описаны в дополнительной документации.

### 26.3.13 Регистр OTPCTR\_PMC\_CMD

При OTP\_SYS\_RW\_EN = 1 регистр позволяет запускать инструкции в блоке PMC. При OTP\_SYS\_RW\_EN = 0 регистр зарезервирован.

Формат регистра OTPCTR\_PMC\_CMD приведен в следующей таблице.

**Таблица 26.24. Поля регистра OTPCTR\_PMC\_CMD**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:7	-	RO	0	Резерв
6:0	INSTR	WO	0	Запись в это поле инициирует выполнение инструкции в блоке PMC: 0x1: инструкция START - запуск выбранной операции; 0x2: инструкция STOP - остановка запущенной операции; остальное: резерв.

## 27. CRYPTO: КРИПТО-АКСЕЛЕРАТОР CRYPTOCCELL

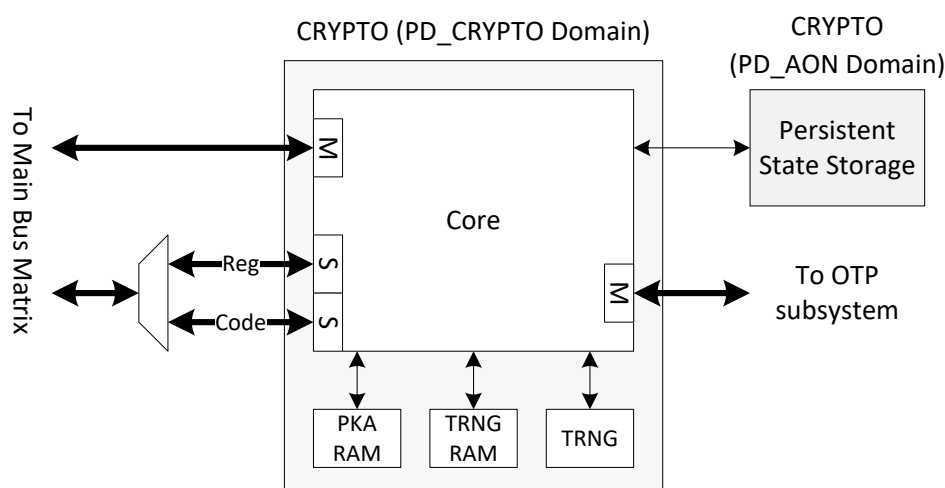
### 27.1 Введение

Блок CryptoCell предоставляет функции, необходимые для организации безопасности системы, в том числе:

- Аппаратное ускорение криптографических алгоритмов.
- Верификация образа программы при загрузке.
- Организация доверенной отладки.
- Генерация случайных чисел.
- Управление жизненным циклом микросхемы.

### 27.2 Функционирование

На рисунке показана структурная схема блока CryptoCell.



**Рисунок 27.1. Структурная схема блока CryptoCell**

Блок состоит из двух частей: одна часть расположена в отключаемом домене PD\_CRYPTO (вычислительное ядро, блоки памяти, генератор случайных чисел), вторая – в домене PD\_AON (хранение состояния блока).

### 27.3 Назначение битов DCU

В следующей таблице показано назначение битов регистров HOST\_DCU\_EN0, HOST\_DCU\_LOCK0 и AO\_ICV\_DCU\_RESTRICTION\_MASK0. В таблице также приведены значения битов после сброса для различных LCS. Серым цветом отмечены значения без возможности изменения в соответствующем LCS.



**Таблица 27.1. Назначение битов DCU**

Бит	Назначение	Life Cycle State			
		CM	DM	SECURE	RMA
1	OTP_SYS_RW_EN – разрешение доступа к OTP памяти через системный интерфейс контроллера OTPCTR	1	0	0	0
2	FLASH_SYS_RO_EN – включение режима Read-Only для системного раздела Flash памяти	0	1	1	1
3	FLASH_BIST_EN – включение блока BIST для тестирования Flash памяти	0	0	0	0
4	DBGEN	1	1	0	1
5	NIDEN	1	1	0	1
6	SPIDEN	1	1	0	1
7	SPNIDEN	1	1	0	1
8	OTP_USER_WR_LOCK – блокировка записи в пользовательскую область OTP памяти	0	0	1	1

## 27.4 Регистры

TBD

## 28. GMS: GMS CRYPTO CORE

### 28.1 Общая информация

GMS-Crypto – ускоритель отечественных криптоалгоритмов. Структурная схема показана на Рисунок 28.1.

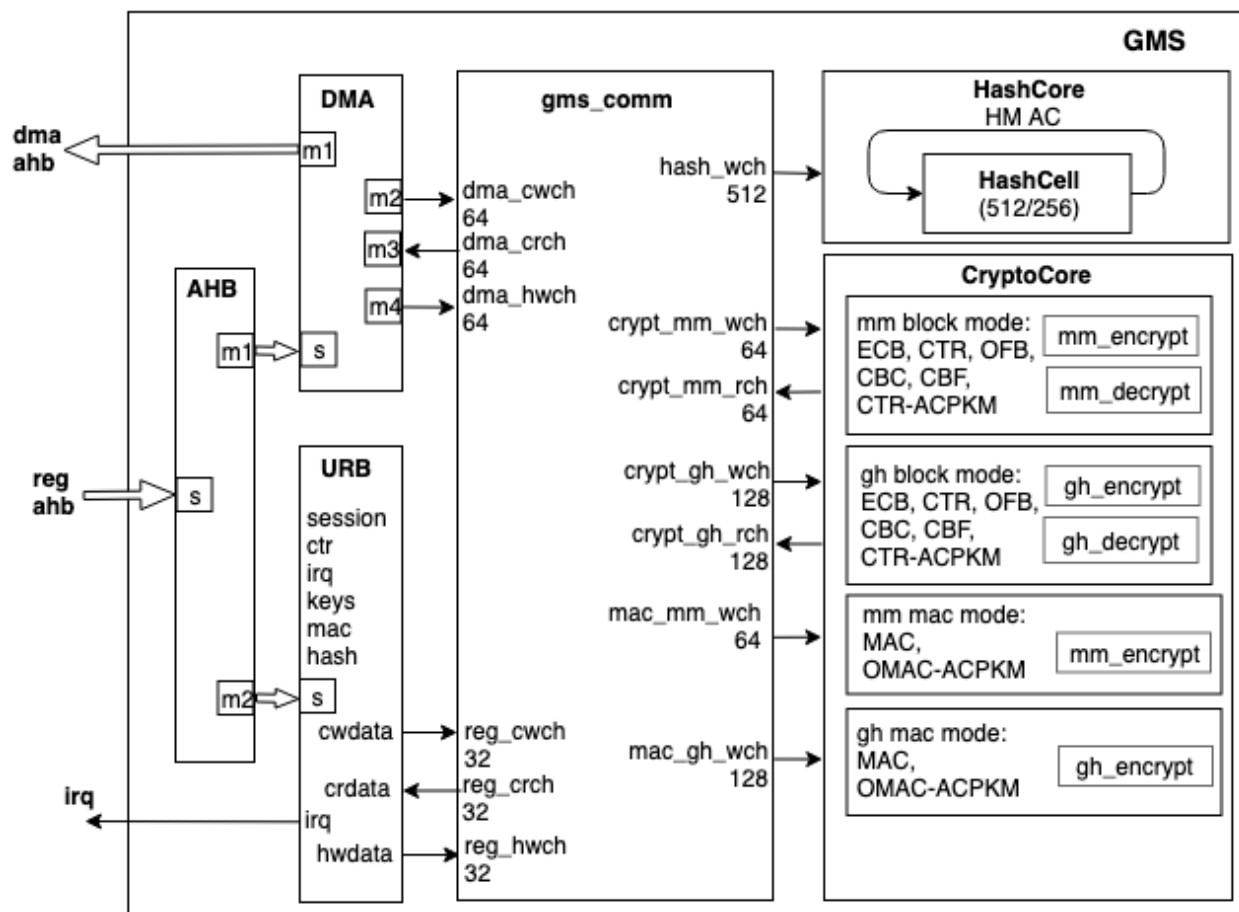


Рисунок 28.1. Структурная схема GMS\_Crypto

Ниже описаны назначение и основные параметры составных блоков GMS-Crypto:

**CryptoCore.** Ускоритель алгоритмов блочного шифрования и выработки имитовставки.

- Ускоритель поддерживает работу с базовыми алгоритмами “Кузнечик” и “Магма” (ГОСТ Р 34.12-2015).
- При работе с блочными шифрами могут быть использованы режимы из ГОСТ Р 34.13-2015: режим простой замены (ECB), режим счетчика (CTR), режим гаммирования с обратной связью по выходу (OFB), режим простой замены с зацеплением (CBC), режим гаммирования с обратной связью по шифротексту (CFB).
- Поддерживается режим работы CTR-АСРКМ рекомендации Р 1323565.1.017-2018.

- Вычисление имитовставки может происходить в режиме MAC из ГОСТ Р 34.13-2015 или в режиме ОMAC-АСРКМ из рекомендации Р 1323565.1.017-2018.
- Вычисление имитовставки MAC может происходить одновременно с блочным шифрованием/расшифрованием данных. При этом имитовставка будет всегда вычисляться на основе незашифрованных данных.
- Базовый алгоритм вычисления MAC выбирается независимо от базового алгоритма блочного шифрования. В качестве сессионного ключа может быть использован как сессионный ключ блочного шифрования, так и свой уникальный ключ.
- Вычисление имитовставки ОMAC-АСРКМ не может происходить одновременно с блочным шифрованием/расшифрованием.

#### **HashCore. Ускоритель алгоритмов вычисления хэш-функции.**

- Ускоритель производит вычисление хэш-функции по алгоритму “Стрибог” ГОСТ Р 34.11-2012.
- Ускоритель может работать в режиме вычисления кода аутентификации сообщения HMAC Рекомендация Р 50.1.113-2016.
- GMS-Crypto поддерживает режим девирсификации ключей, при котором результат работы HashCore может быть непосредственно использован в качестве сессионного ключа блочного шифрования и/или вычисления MAC.
- HashCore имеет свой входной канал данных. Т.е. GMS-Crypto не поддерживает одновременное вычисление хэш-функции и блочное шифрование/расшифрование на одном потоке данных.

#### **GMS-DMA.**

- Блок GMS-Crypto имеет встроенный DMA-контроллер.
- DMA имеет 4 мастер-интерфейса:
  - m1: dma\_ahb – внешний мастер интерфейс.
  - m2: dma\_cwch – интерфейс записи данных в CryptoCore
  - m3: dma\_crch – интерфейс чтения данных из CryptoCore
  - m4: dma\_hwch – интерфейс записи данных в HashCore

#### **URB. Универсальный блок регистров.**

- Набор регистров управления GMS-Crypto.

## **28.2 Адресное пространство**

Таблица 28.1 Адресное пространство gms\_crypto

Смещение	Конечный адрес	Описание
0x0000	0x0FFF	Регистры URB

0x1000	0x1FFF	Регистры PDMA
--------	--------	---------------

## 28.3 Регистры CRYPTO URB

Перечень регистров urb приведен в Таблица 28.2.

**Таблица 28.2. Перечень программно-доступных регистров urb**

Смещение	Условное обозначение	Название регистра	Исходное состояние	Тип доступа
0x000	gms_id	Регистр ревизии блока криптоакселератора gms_crypto.	00000015	RO
0x004	cryptCore_session_ctr0	Регистр управления сессией cryptCore.	00000000	RW
0x008	hashCore_session_ctr0	Регистр управления сессией hashCore.	00000000	RW
0x00C	session_ctrl	Дополнительный write_only регистр управления сессией.	00000000	writeOnce
0x02C	watermark	Регистр пороговых значений заполнения FIFO	00000444	RW
0x030	fifo_status	Регистр статуса FIFO	00000000	RO
0x034	irq_status	Регистр статуса прерывания	00000000	RO
0x038	irq_clr	Регистр сброса прерывания	00000000	writeOnce
0x03C	irq_mask	Регистр маски прерывания	000001F8	RW
0x100	ctx0_crypt_block_size_low	Младший регистр размера области данных для блочного шифрования/вычисления MAC.	00000000	RW
0x104	ctx0_crypt_block_size_high	Старший регистр размера области данных для блочного шифрования/вычисления MAC.	00000000	RW
0x108	ctx0_crypt_block_size_per_key	Регистр размера области данных для криптопреобразования на 1 ключ.	00000000	RW
0x10C	ctx0_ctracpkm_section_size	Регистр длины секции N CTR-АСРКМ.	00000000	RW
0x110	ctx0_omac_section_size	Регистр длины секции N ОМАС-АСРКМ.	00000000	RW
0x114	ctx0_ctracpkm_sec_cnt	Регистр счетчика обработанных байт секции CTR-АСРКМ	00000000	RW
0x118	ctx0_omac_key_change_cnt	Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе	00000000	RW
0x120	ctx0_crypt_key0	Регистр 0 сессионного ключа блочного шифрования.	00000000	RW
0x124	ctx0_crypt_key1	Регистр 1 сессионного ключа блочного шифрования.	00000000	RW
0x128	ctx0_crypt_key2	Регистр 2 сессионного ключа блочного шифрования.	00000000	RW
0x12C	ctx0_crypt_key3	Регистр 3 сессионного ключа блочного шифрования.	00000000	RW
0x130	ctx0_crypt_key4	Регистр 4 сессионного ключа блочного шифрования.	00000000	RW
0x134	ctx0_crypt_key5	Регистр 5 сессионного ключа блочного шифрования.	00000000	RW
0x138	ctx0_crypt_key6	Регистр 6 сессионного ключа блочного шифрования.	00000000	RW
0x13C	ctx0_crypt_key7	Регистр 7 сессионного ключа блочного шифрования.	00000000	RW

0x140	ctx0_iv0	Регистр 0 инициализирующей последовательности.	00000000	RW
0x144	ctx0_iv1	Регистр 1 инициализирующей последовательности.	00000000	RW
0x148	ctx0_iv2	Регистр 2 инициализирующей последовательности.	00000000	RW
0x14C	ctx0_iv3	Регистр 3 инициализирующей последовательности.	00000000	RW
0x150	ctx0_iv4	Регистр 4 инициализирующей последовательности.	00000000	RW
0x154	ctx0_iv5	Регистр 5 инициализирующей последовательности.	00000000	RW
0x158	ctx0_iv6	Регистр 6 инициализирующей последовательности.	00000000	RW
0x15C	ctx0_iv7	Регистр 7 инициализирующей последовательности.	00000000	RW
0x160	ctx0_iv8	Регистр 8 инициализирующей последовательности.	00000000	RW
0x164	ctx0_iv9	Регистр 9 инициализирующей последовательности.	00000000	RW
0x168	ctx0_iv10	Регистр 10 инициализирующей последовательности.	00000000	RW
0x16C	ctx0_iv11	Регистр 11 инициализирующей последовательности.	00000000	RW
0x170	ctx0_iv12	Регистр 12 инициализирующей последовательности.	00000000	RW
0x174	ctx0_iv13	Регистр 13 инициализирующей последовательности.	00000000	RW
0x178	ctx0_iv14	Регистр 14 инициализирующей последовательности.	00000000	RW
0x17C	ctx0_iv15	Регистр 15 инициализирующей последовательности.	00000000	RW
0x180	ctx0_mac_key0	Регистр 0 сессионного ключа вычисления имитовставки.	00000000	RW
0x184	ctx0_mac_key1	Регистр 1 сессионного ключа вычисления имитовставки.	00000000	RW
0x188	ctx0_mac_key2	Регистр 2 сессионного ключа вычисления имитовставки.	00000000	RW
0x18C	ctx0_mac_key3	Регистр 3 сессионного ключа вычисления имитовставки.	00000000	RW
0x190	ctx0_mac_key4	Регистр 4 сессионного ключа вычисления имитовставки.	00000000	RW
0x194	ctx0_mac_key5	Регистр 5 сессионного ключа вычисления имитовставки.	00000000	RW
0x198	ctx0_mac_key6	Регистр 6 сессионного ключа вычисления имитовставки.	00000000	RW
0x19C	ctx0_mac_key7	Регистр 7 сессионного ключа вычисления имитовставки.	00000000	RW
0x1A0	ctx0_mac0	Регистр 0 имитовставки mac	00000000	RW
0x1A4	ctx0_mac1	Регистр 1 имитовставки mac	00000000	RW
0x1A8	ctx0_mac2	Регистр 2 имитовставки mac	00000000	RW
0x1AC	ctx0_mac3	Регистр 3 имитовставки mac	00000000	RW
0x200	ctx0_hash_block_size_low	Младший регистр размера области данных для вычисления hash.	00000000	RW
0x204	ctx0_hash_block_size_high	Старший регистр размера области данных для hashCore.	00000000	RW
0x208	ctx0_hash_N_low	Регистр счетчика hash_N. Младшая часть	00000000	RW

0x20C	ctx0_hash_N_high	Регистр счетчика hash_N. Старшая часть.	00000000	RW
0x210	ctx0_hash0	Регистр 0 хэш-суммы.	00000000	RW
0x214	ctx0_hash1	Регистр 1 хэш-суммы.	00000000	RW
0x218	ctx0_hash2	Регистр 2 хэш-суммы.	00000000	RW
0x21C	ctx0_hash3	Регистр 3 хэш-суммы.	00000000	RW
0x220	ctx0_hash4	Регистр 4 хэш-суммы.	00000000	RW
0x224	ctx0_hash5	Регистр 5 хэш-суммы.	00000000	RW
0x228	ctx0_hash6	Регистр 6 хэш-суммы.	00000000	RW
0x22C	ctx0_hash7	Регистр 7 хэш-суммы.	00000000	RW
0x230	ctx0_hash8	Регистр 8 хэш-суммы.	00000000	RW
0x234	ctx0_hash9	Регистр 9 хэш-суммы.	00000000	RW
0x238	ctx0_hash10	Регистр 10 хэш-суммы.	00000000	RW
0x23C	ctx0_hash11	Регистр 11 хэш-суммы.	00000000	RW
0x240	ctx0_hash12	Регистр 12 хэш-суммы.	00000000	RW
0x244	ctx0_hash13	Регистр 13 хэш-суммы.	00000000	RW
0x248	ctx0_hash14	Регистр 14 хэш-суммы.	00000000	RW
0x24C	ctx0_hash15	Регистр 15 хэш-суммы.	00000000	RW
0x250	ctx0_hmac_key0	Регистр 0 ключа вычисления HMAC.	00000000	RW
0x254	ctx0_hmac_key1	Регистр 1 ключа вычисления HMAC.	00000000	RW
0x258	ctx0_hmac_key2	Регистр 2 ключа вычисления HMAC.	00000000	RW
0x25C	ctx0_hmac_key3	Регистр 3 ключа вычисления HMAC.	00000000	RW
0x260	ctx0_hmac_key4	Регистр 4 ключа вычисления HMAC.	00000000	RW
0x264	ctx0_hmac_key5	Регистр 5 ключа вычисления HMAC.	00000000	RW
0x268	ctx0_hmac_key6	Регистр 6 ключа вычисления HMAC.	00000000	RW
0x26C	ctx0_hmac_key7	Регистр 7 ключа вычисления HMAC.	00000000	RW
0x270	ctx0_hmac_key8	Регистр 8 ключа вычисления HMAC.	00000000	RW
0x274	ctx0_hmac_key9	Регистр 9 ключа вычисления HMAC.	00000000	RW
0x278	ctx0_hmac_key10	Регистр 10 ключа вычисления HMAC.	00000000	RW
0x27C	ctx0_hmac_key11	Регистр 11 ключа вычисления HMAC.	00000000	RW
0x280	ctx0_hmac_key12	Регистр 12 ключа вычисления HMAC.	00000000	RW
0x284	ctx0_hmac_key13	Регистр 13 ключа вычисления HMAC.	00000000	RW
0x288	ctx0_hmac_key14	Регистр 14 ключа вычисления HMAC.	00000000	RW
0x28C	ctx0_hmac_key15	Регистр 15 ключа вычисления HMAC.	00000000	RW
0x290	ctx0_hash_sig0	Регистр 0 значения переменной hash_sig	00000000	RW
0x294	ctx0_hash_sig1	Регистр 1 значения переменной hash_sig	00000000	RW
0x298	ctx0_hash_sig2	Регистр 2 значения переменной hash_sig	00000000	RW
0x29C	ctx0_hash_sig3	Регистр 3 значения переменной hash_sig	00000000	RW
0x2A0	ctx0_hash_sig4	Регистр 4 значения переменной hash_sig	00000000	RW
0x2A4	ctx0_hash_sig5	Регистр 5 значения переменной hash_sig	00000000	RW
0x2A8	ctx0_hash_sig6	Регистр 6 значения переменной hash_sig	00000000	RW
0x2AC	ctx0_hash_sig7	Регистр 7 значения переменной hash_sig	00000000	RW
0x2B0	ctx0_hash_sig8	Регистр 8 значения переменной hash_sig	00000000	RW
0x2B4	ctx0_hash_sig9	Регистр 9 значения переменной hash_sig	00000000	RW
0x2B8	ctx0_hash_sig10	Регистр 10 значения переменной hash_sig	00000000	RW
0x2BC	ctx0_hash_sig11	Регистр 11 значения переменной hash_sig	00000000	RW

0x2C0	ctx0_hash_sig12	Регистр 12 значения переменной hash_sig	00000000	RW
0x2C4	ctx0_hash_sig13	Регистр 13 значения переменной hash_sig	00000000	RW
0x2C8	ctx0_hash_sig14	Регистр 14 значения переменной hash_sig	00000000	RW
0x2CC	ctx0_hash_sig15	Регистр 15 значения переменной hash_sig	00000000	RW
0x300	ctx1_crypt_block_size_low	Младший регистр размера области данных для блочного шифрования/вычисления MAC.	00000000	RW
0x304	ctx1_crypt_block_size_high	Старший регистр размера области данных для блочного шифрования/вычисления MAC.	00000000	RW
0x308	ctx1_crypt_block_size_per_key	Регистр размера области данных для криптопреобразования на 1 ключ.	00000000	RW
0x30C	ctx1_ctracpkm_section_size	Регистр длины секции N CTR-АСРКМ.	00000000	RW
0x310	ctx1_omac_section_size	Регистр длины секции N ОМАС-АСРКМ.	00000000	RW
0x314	ctx1_ctracpkm_sec_cnt	Регистр счетчика обработанных байт секции CTR-АСРКМ	00000000	RW
0x318	ctx1_omac_key_change_cnt	Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе	00000000	RW
0x320	ctx1_crypt_key0	Регистр 0 сессионного ключа блочного шифрования.	00000000	RW
0x324	ctx1_crypt_key1	Регистр 1 сессионного ключа блочного шифрования.	00000000	RW
0x328	ctx1_crypt_key2	Регистр 2 сессионного ключа блочного шифрования.	00000000	RW
0x32C	ctx1_crypt_key3	Регистр 3 сессионного ключа блочного шифрования.	00000000	RW
0x330	ctx1_crypt_key4	Регистр 4 сессионного ключа блочного шифрования.	00000000	RW
0x334	ctx1_crypt_key5	Регистр 5 сессионного ключа блочного шифрования.	00000000	RW
0x338	ctx1_crypt_key6	Регистр 6 сессионного ключа блочного шифрования.	00000000	RW
0x33C	ctx1_crypt_key7	Регистр 7 сессионного ключа блочного шифрования.	00000000	RW
0x340	ctx1_iv0	Регистр 0 инициализирующей последовательности.	00000000	RW
0x344	ctx1_iv1	Регистр 1 инициализирующей последовательности.	00000000	RW
0x348	ctx1_iv2	Регистр 2 инициализирующей последовательности.	00000000	RW
0x34C	ctx1_iv3	Регистр 3 инициализирующей последовательности.	00000000	RW
0x350	ctx1_iv4	Регистр 4 инициализирующей последовательности.	00000000	RW
0x354	ctx1_iv5	Регистр 5 инициализирующей последовательности.	00000000	RW
0x358	ctx1_iv6	Регистр 6 инициализирующей последовательности.	00000000	RW
0x35C	ctx1_iv7	Регистр 7 инициализирующей последовательности.	00000000	RW
0x360	ctx1_iv8	Регистр 8 инициализирующей последовательности.	00000000	RW



0x364	ctx1_iv9	Регистр 9 инициализирующей последовательности.	00000000	RW
0x368	ctx1_iv10	Регистр 10 инициализирующей последовательности.	00000000	RW
0x36C	ctx1_iv11	Регистр 11 инициализирующей последовательности.	00000000	RW
0x370	ctx1_iv12	Регистр 12 инициализирующей последовательности.	00000000	RW
0x374	ctx1_iv13	Регистр 13 инициализирующей последовательности.	00000000	RW
0x378	ctx1_iv14	Регистр 14 инициализирующей последовательности.	00000000	RW
0x37C	ctx1_iv15	Регистр 15 инициализирующей последовательности.	00000000	RW
0x380	ctx1_mac_key0	Регистр 0 сессионного ключа вычисления имитовставки.	00000000	RW
0x384	ctx1_mac_key1	Регистр 1 сессионного ключа вычисления имитовставки.	00000000	RW
0x388	ctx1_mac_key2	Регистр 2 сессионного ключа вычисления имитовставки.	00000000	RW
0x38C	ctx1_mac_key3	Регистр 3 сессионного ключа вычисления имитовставки.	00000000	RW
0x390	ctx1_mac_key4	Регистр 4 сессионного ключа вычисления имитовставки.	00000000	RW
0x394	ctx1_mac_key5	Регистр 5 сессионного ключа вычисления имитовставки.	00000000	RW
0x398	ctx1_mac_key6	Регистр 6 сессионного ключа вычисления имитовставки.	00000000	RW
0x39C	ctx1_mac_key7	Регистр 7 сессионного ключа вычисления имитовставки.	00000000	RW
0x3A0	ctx1_mac0	Регистр 0 имитовставки mac	00000000	RW
0x3A4	ctx1_mac1	Регистр 1 имитовставки mac	00000000	RW
0x3A8	ctx1_mac2	Регистр 2 имитовставки mac	00000000	RW
0x3AC	ctx1_mac3	Регистр 3 имитовставки mac	00000000	RW
0x400	ctx1_hash_block_size_low	Младший регистр размера области данных для вычисления hash.	00000000	RW
0x404	ctx1_hash_block_size_high	Старший регистр размера области данных для hashCore.	00000000	RW
0x408	ctx1_hash_N_low	Регистр счетчика hash_N. Младшая часть	00000000	RW
0x40C	ctx1_hash_N_high	Регистр счетчика hash_N. Старшая часть.	00000000	RW
0x410	ctx1_hash0	Регистр 0 хэш-суммы.	00000000	RW
0x414	ctx1_hash1	Регистр 1 хэш-суммы.	00000000	RW
0x418	ctx1_hash2	Регистр 2 хэш-суммы.	00000000	RW
0x41C	ctx1_hash3	Регистр 3 хэш-суммы.	00000000	RW
0x420	ctx1_hash4	Регистр 4 хэш-суммы.	00000000	RW
0x424	ctx1_hash5	Регистр 5 хэш-суммы.	00000000	RW
0x428	ctx1_hash6	Регистр 6 хэш-суммы.	00000000	RW
0x42C	ctx1_hash7	Регистр 7 хэш-суммы.	00000000	RW
0x430	ctx1_hash8	Регистр 8 хэш-суммы.	00000000	RW
0x434	ctx1_hash9	Регистр 9 хэш-суммы.	00000000	RW
0x438	ctx1_hash10	Регистр 10 хэш-суммы.	00000000	RW
0x43C	ctx1_hash11	Регистр 11 хэш-суммы.	00000000	RW
0x440	ctx1_hash12	Регистр 12 хэш-суммы.	00000000	RW



0x444	ctx1_hash13	Регистр 13 хэш-суммы.	00000000	RW
0x448	ctx1_hash14	Регистр 14 хэш-суммы.	00000000	RW
0x44C	ctx1_hash15	Регистр 15 хэш-суммы.	00000000	RW
0x450	ctx1_hmac_key0	Регистр 0 ключа вычисления HMAC.	00000000	RW
0x454	ctx1_hmac_key1	Регистр 1 ключа вычисления HMAC.	00000000	RW
0x458	ctx1_hmac_key2	Регистр 2 ключа вычисления HMAC.	00000000	RW
0x45C	ctx1_hmac_key3	Регистр 3 ключа вычисления HMAC.	00000000	RW
0x460	ctx1_hmac_key4	Регистр 4 ключа вычисления HMAC.	00000000	RW
0x464	ctx1_hmac_key5	Регистр 5 ключа вычисления HMAC.	00000000	RW
0x468	ctx1_hmac_key6	Регистр 6 ключа вычисления HMAC.	00000000	RW
0x46C	ctx1_hmac_key7	Регистр 7 ключа вычисления HMAC.	00000000	RW
0x470	ctx1_hmac_key8	Регистр 8 ключа вычисления HMAC.	00000000	RW
0x474	ctx1_hmac_key9	Регистр 9 ключа вычисления HMAC.	00000000	RW
0x478	ctx1_hmac_key10	Регистр 10 ключа вычисления HMAC.	00000000	RW
0x47C	ctx1_hmac_key11	Регистр 11 ключа вычисления HMAC.	00000000	RW
0x480	ctx1_hmac_key12	Регистр 12 ключа вычисления HMAC.	00000000	RW
0x484	ctx1_hmac_key13	Регистр 13 ключа вычисления HMAC.	00000000	RW
0x488	ctx1_hmac_key14	Регистр 14 ключа вычисления HMAC.	00000000	RW
0x48C	ctx1_hmac_key15	Регистр 15 ключа вычисления HMAC.	00000000	RW
0x490	ctx1_hash_sig0	Регистр 0 значения переменной hash_sig	00000000	RW
0x494	ctx1_hash_sig1	Регистр 1 значения переменной hash_sig	00000000	RW
0x498	ctx1_hash_sig2	Регистр 2 значения переменной hash_sig	00000000	RW
0x49C	ctx1_hash_sig3	Регистр 3 значения переменной hash_sig	00000000	RW
0x4A0	ctx1_hash_sig4	Регистр 4 значения переменной hash_sig	00000000	RW
0x4A4	ctx1_hash_sig5	Регистр 5 значения переменной hash_sig	00000000	RW
0x4A8	ctx1_hash_sig6	Регистр 6 значения переменной hash_sig	00000000	RW
0x4AC	ctx1_hash_sig7	Регистр 7 значения переменной hash_sig	00000000	RW
0x4B0	ctx1_hash_sig8	Регистр 8 значения переменной hash_sig	00000000	RW
0x4B4	ctx1_hash_sig9	Регистр 9 значения переменной hash_sig	00000000	RW
0x4B8	ctx1_hash_sig10	Регистр 10 значения переменной hash_sig	00000000	RW
0x4BC	ctx1_hash_sig11	Регистр 11 значения переменной hash_sig	00000000	RW
0x4C0	ctx1_hash_sig12	Регистр 12 значения переменной hash_sig	00000000	RW
0x4C4	ctx1_hash_sig13	Регистр 13 значения переменной hash_sig	00000000	RW
0x4C8	ctx1_hash_sig14	Регистр 14 значения переменной hash_sig	00000000	RW
0x4CC	ctx1_hash_sig15	Регистр 15 значения переменной hash_sig	00000000	RW
0x500	wcdata	Регистр записи данных в cryptoCore	00000000	WO
0x540	rcdata	Регистр чтения данных из cryptoCore	00000000	RO
0x580	whdata	Регистр записи данных в hashCore	00000000	WO
0x600	soft_reset	Программный сброс gms.	00000001	writeOnce

### 28.3.1 Регистр gms\_id

Смещение: 0x000

Регистр ревизии блока криптоакселератора gms\_crypto.

Формат регистра gms\_id приведен в следующей таблице:

**Таблица 28.3. Поля регистра gms\_id**

Разряды	Название	Описание	Сброс
31:0	ID	Регистр ревизии блока криптоакселератора gms_crypto.	0

## 28.3.2 Регистр cryptCore\_session\_ctr0

Смещение: 0x004

Регистр определяет параметры сессии ядра CryptCore и управляет инициализацией сессии.

Формат регистра cryptCore\_session\_ctr0 приведен в следующей таблице:

**Таблица 28.4. Поля регистра cryptCore\_session\_ctr0**

Разряды	Название	Описание	Сброс
31:24	-	Резерв	0
23	lock_ctx	Регистр блокировки активного контекста. Если в момент старта сессии lock==1, следующие регистры и поля будут недоступны для записи и чтения до момента завершения сессии cryptCore или ее принудительной остановки: - cryptCore_session_ctr0 - ctx*_crypt_block_size_[low/high] - ctx*_crypt_block_size_per_key - ctx*_ctracpkm_section_size - ctx*_omac_section_size - ctx*_ctracpkm_sec_cnt - ctx*_omac_key_change_cnt - ctx*_crypt_key[0..7] - ctx*_iv[0..15] - ctx*_mac_key[0..7] - ctx*_mac[0..3] где * - активный контекст сессии. Также будет невозможен flush активного контекста.	0
22	ctx	Регистр активного контекста пререквизитов ядра cryptCore.  0: ctx0. 1: ctx1.	0
21	endian_output	Регистр преобразования потока данных little-endian => big-endian для выходного потока данных из cryptCore.  0: Преобразование выключено. 1: Преобразование включено.	0

Разряды	Название	Описание	Сброс
20	endian_input	Регистр преобразования потока данных little-endian => big-endian для входного потока данных в cryptoCore.  0: Преобразование выключено. 1: Преобразование включено.	0
19	mac_finalize_resolve	Поле разрешения финализации MAC. Если поле установлено в 1, то при достижении 0 счетчиком mac_block_cnt, произойдет финализация текущей MAC-сессии.	0
18	crypt_key_div_enable	Поле включения поддержки диверсификации ключей.  Если это поле установлено в 1, сессия cryptCore будет останавливаться каждый раз, когда криптоакселератор завершает обработку объема данных, указанного в регистре crypt_block_size_per_key.  Для возобновления сессии необходимо записать 1 в session_ctr1.resume_crypt. После записи произойдет перевычисление раундовых ключей блока шифрования и блока вычисления имитовставки.	0
17	enable_dma_read	Поле включения gms-dma для канала чтения cryptCore:  0: Чтение данных из cryptoCore будет осуществляться через программный регистр rdata.  1: Чтение данных из cryptoCore осуществляется через интерфейс master3 gms-dma.	0
16	enable_dma_write	Поле включения gms-dma для канала записи cryptCore:  0: Запись данных в cryptoCore будет осуществляться через программный регистр wdata.  1: Запись данных в cryptoCore будет осуществляться через интерфейс master2 gms-dma.	0
15	mac_key_source	Поле источника ключа вычисления имитовставки:  0: Регистры crypt_keyx 1: Регистры mac_keyx	0
14	omac_new_session	При старте сессии в режиме вычисления ОМАС-АСРКМ (mac_mode==1), этот регистр определяет, произойдет старт новой сессии ОМАС-АСРКМ или продолжится вычисление инициализированной ранее сессии.  0: Возобновление старой сессии. 1: Старт новой сессии.	0

Разряды	Название	Описание	Сброс
13	mac_mode	Выбор режима выработки имитовставки:  0: MAC. ГОСТ 34.13-2015 1: OMAC-АСРКМ. Рекомендация Р 1323565.1.017-2018  Криптоакселератор не поддерживает одновременное вычисление имитовставки по стандарту OMAC-АСРКМ и работу с блочным шифрованием.	0
12	mac_ghmm	Выбор базового алгоритма вычисления имитовставки по ГОСТ Р 34.12-2015:  0: Кузнечик 1: Магма	0
11:8	crypt_feedback_reg_size	Разрядность регистра зацепления: 0x0: n 0x1: n*2 0x2: n*3 0x3: n*4 n - размер блока базового шифра (128 для Кузнечика, 64 для магмы)	0
7:5	crypt_mode	Выбор режима блочного шифрования.  ГОСТ Р 34.13-2015:  3'b000: ECB. Режим простой замены. 3'b001: CTR. Режим счетчика. 3'b010: OFB. Режим гаммирования с обратной связью по выходу. 3'b011: CBC. Режим простой замены с зацеплением. 3'b100: CFB. Режим гаммирования с обратной связью по шифротексту.  Рекомендация Р 1323565.1.017-2018:  3'b101: CTR-АСРКМ. 3'b110: Резерв 3'b111: Резерв	0
4	crypt_ghmm	Выбор базового алгоритма шифрования по ГОСТ Р 34.12-2015:  0: Кузнечик 1: Магма	0
3	crypt_ed	Режим направления шифрования:  0: зашифрование 1: расшифрование	0
2	enable_mac	Включение вычисления имитовставки в cryptCore.  Если значение этого поля 1, то в сессии будет задействован блок вычисления имитовставки.	0
1	enable_crypt	Включение блочного шифрования в cryptCore.  Если значение этого поля 1, то в сессии будет задействован блок шифрования.	0

Разряды	Название	Описание	Сброс
0	start	Старт сессии cryptoCore.  Запись 1 в это поле приведет старту сессии работы блока cryptoCore (блочные шифры + MAC)	0

### 28.3.3 Регистр hashCore\_session\_ctr0

Смещение: 0x008

Регистр определяет параметры сессии ядра hashCore и управляет инициализацией сессии.

Формат регистра hashCore\_session\_ctr0 приведен в следующей таблице:

**Таблица 28.5. Поля регистра hashCore\_session\_ctr0**

Разряды	Название	Описание	Сброс
31:10	-	Резерв	0
9	lock_ctx	Регистр блокировки активного контекста. Если в момент старта сессии lock==1, следующие регистры и поля будут недоступны для записи и чтения до момента завершения сессии hashCore или ее принудительной остановки: - hashCore_session_ctr0 - ctx*_hash_block_size_[low/high] - ctx*_hash[0..15] - ctx*_hmac_key[0..15] - ctx*_hash_sig[0..15] - ctx*_hash_N_[low/high] где * - активный контекст сессии hashCore. Также будет невозможен flush активного контекста.	0
8	ctx	Регистр активного контекста пререквизитов ядра hashCore.  0: ctx0. 1: ctx1.	0
7	endian	Регистр преобразования потока данных little-endian => big-endian для потока данных в hashCore.  0: Преобразование выключено 1: Преобразование включено	0
6	hash_finalize_resolve	Поле разрешения финализации hash. Если поле установлено в 1, то при достижении 0 счетчиком hash_block_cnt, произойдет финализация текущей hash-сессии.	0

Разряды	Название	Описание	Сброс
5	enable_hash_dma	Поле включения gms-dma для канала записи в hashCore:  0: Запись данных в hashCore будет осуществляться через программный регистр whdata.  1: Запись данных в hashCore будет осуществляться через интерфейс master4 gms-dma.	0
4:3	hash_dst	Поле выбора месторасположения результатов работы hashCore.  0x0: Регистры hashx 0x1: Регистры crypt_keyx активного контекста cryptCore. 0x2: Регистры mac_keyx активного контекста cryptCore. 0x3: Резерв	0
2	hash_size	Поле выбора длины хэш-кода для функции хеширования или кода аутентификации HMAC.  0: 512-бит 1: 256-бит	0
1	hash_mode	Поле режима работы вычислителя hash:  0: Вычисление хэш-функции от входных данных. Стандарт ГОСТ Р 34.11-2012  1: Вычисление кода аутентификации сообщения HMAC Рекомендация Р 50.1.113-2016	0
0	start_hashCore	Старт сессии hashCore.  Запись 1 в это поле приведет старту сессии работы блока hashCore (hash/HMAC)	0

### 28.3.4 Регистр session\_ctrl

Смещение: 0x00C

Второй регистр управления сессией.

Регистр позволяет управлять сессией не перезаписывая ее основные параметры.

Формат регистра session\_ctrl приведен в следующей таблице:

**Таблица 28.6. Поля регистра session\_ctrl**

Разряды	Название	Описание	Сброс
31:12	-	Резерв	0
11	powerdown_req	Запрос на выключение питания.  Запись 1 в поле приведет к установке в 1 сигнала разрешения выключения питания powerdown_rdy после корректного завершения текущей сессии.	0

Разряды	Название	Описание	Сброс
10	flush_hashCore_ctx1	Поле сброса пререквезитов ctx1 сессии hashCore. Запись 1 в это поле приводит к сбросу в 0 регистров: <ul style="list-style-type: none"> <li>- ctx1_hash_block_size_[low/high]</li> <li>- ctx1_hash[0..15]</li> <li>- ctx1_hmac_key[0..15]</li> <li>- ctx1_hash_sig[0..15]</li> <li>- ctx1_hash_N_[low/high]</li> </ul>	0
9	flush_hashCore_ctx0	Поле сброса пререквезитов ctx0 сессии hashCore. Запись 1 в это поле приводит к сбросу в 0 регистров: <ul style="list-style-type: none"> <li>- ctx0_hash_block_size_[low/high]</li> <li>- ctx0_hash[0..15]</li> <li>- ctx0_hmac_key[0..15]</li> <li>- ctx0_hash_sig[0..15]</li> <li>- ctx0_hash_N_[low/high]</li> </ul>	0
8	flush_cryptCore_ctx1	Поле сброса пререквезитов ctx1 сессии cryptoCore. Запись 1 в это поле приводит к сбросу в 0 регистров: <ul style="list-style-type: none"> <li>- ctx1_crypt_block_size_[low/high]</li> <li>- ctx1_crypt_block_size_per_key</li> <li>- ctx1_ctracpkm_section_size</li> <li>- ctx1_omac_section_size</li> <li>- ctx1_ctracpkm_sec_cnt</li> <li>- ctx1_omac_key_change_cnt</li> <li>- ctx1_crypt_key[0..7]</li> <li>- ctx1_iv[0..15]</li> <li>- ctx1_mac_key[0..7]</li> <li>- ctx1_mac[0..3]</li> </ul>	0
7	flush_cryptCore_ctx0	Поле сброса пререквезитов ctx0 сессии cryptoCore. Запись 1 в это поле приводит к сбросу в 0 регистров: <ul style="list-style-type: none"> <li>- ctx0_crypt_block_size_[low/high]</li> <li>- ctx0_crypt_block_size_per_key</li> <li>- ctx0_ctracpkm_section_size</li> <li>- ctx0_omac_section_size</li> <li>- ctx0_ctracpkm_sec_cnt</li> <li>- ctx0_omac_key_change_cnt</li> <li>- ctx0_crypt_key[0..7]</li> <li>- ctx0_iv[0..15]</li> <li>- ctx0_mac_key[0..7]</li> <li>- ctx0_mac[0..3]</li> </ul>	0
6	busy_hashCore	Поле статуса занятости блока.  Переход 0 => 1 просиходит после записи 1 в поле start_hashCore или записи 1 в поле resume_hashCore регистра session_ctr1.  Переход 1 => 0 происходит в случаях: 1) После записи 1 в поле stop_hashCore и завершении обработки текущего базового блока данных. 2) При финализации сессии hashCore.	0

Разряды	Название	Описание	Сброс
5	busy_cryptCore	<p>Поле статуса занятости блока.</p> <p>Переход 0 =&gt; 1 просиходит после записи 1 в поле start_cryptCore или записи 1 в поле resume_cryptCore регистра session_ctr1.</p> <p>Переход 1 =&gt; 0 происходит в случаях:            1) После записи 1 в поле stop_cryptCore и завершении обработки текущего базового блока данных.            2) При финализации сессии cryptCore.            3) В случае, когда акселератор останавливается в ожидании смены мастер ключа (key_div_enable==1).</p>	0
4	-		?
3	resume_hash	<p>Поле возобновления сессии hashCore.</p> <p>Поле дублирует session_ctr0.start_hashCore и позволяет запускать сессию не перезаписывая ее параметры.</p> <p>Поле доступно для записи только при busy_hashCore==0.</p>	0
2	resume_crypt	<p>Поле возобновления сессии.</p> <p>Поле дублирует session_ctr0.start_cryptCore и позволяет запускать сессию не перезаписывая ее параметры.</p> <p>Поле доступно для записи только при busy_cryptCore==0.</p>	0
1	stop_hash	<p>Поле остановки сессии.</p> <p>Запись 1 в это поле приведет к остановке сессии hashCore, если поле lock==0.</p> <p>Для возобновления сессии необходимо произвести запись 1 в поле resume hash.</p>	0
0	stop_crypt	<p>Поле остановки сессии.</p> <p>Запись 1 в это поле приведет к остановке сессии cryptoCore, если поле lock==0.</p> <p>Для возобновления сессии необходимо произвести запись 1 в поле resume crypt.</p>	0

### 28.3.5 Регистр watermark

Смещение: 0x02C

Регистр пороговых значений заполнения FIFO

Формат регистра watermark приведен в следующей таблице:



**Таблица 28.7. Поля регистра watermark**

Разряды	Название	Описание	Сброс
31:12	-	Резерв	0
11:8	wrhFIFO	Если число 64-битных слов, хранящихся в wh_fifo, становится меньше либо равно указанному значению, то это приведет к формированию прерывания fifo_wrh_under_wm	4'h4
7:4	rdcFIFO	Если число 64-битных слов, хранящихся в rc_fifo, становится больше либо равно указанному значению, то это приведет к формированию прерывания fifo_rdc_over_wm	4'h4
3:0	wrcFIFO	Если число 64-битных слов, хранящихся в wc_fifo, становится меньше либо равно указанному значению, то это приведет к формированию прерывания fifo_wrc_under_wm	4'h4

### 28.3.6 Регистр fifo\_status

Смещение: 0x030

Регистр статуса FIFO

Формат регистра fifo\_status приведен в следующей таблице:

**Таблица 28.8. Поля регистра fifo\_status**

Разряды	Название	Описание	Сброс
31:22	-	Резерв	0
21:18	wh_fifo_status	Число 64-х битных слов, хранящихся в wh_fifo	0
17	wh_fifo_full	Признак заполнения wh_fifo	0
16	wh_fifo_empty	Признак отсутствия данных в wh_fifo	0
15:14	-	Резерв	0
13:10	rc_fifo_status	Число 64-х битных слов, хранящихся в rc_fifo	0
9	rc_fifo_full	Признак заполнения rc_fifo	0
8	rc_fifo_empty		0
7:6	-	Резерв	0
5:2	wc_fifo_status	Число 64-х битных слов, хранящихся в wc_fifo	0
1	wc_fifo_full	Признак заполнения wc_fifo	0
0	wc_fifo_empty	Признак отсутствия данных в wc_fifo	0

### 28.3.7 Регистр irq\_status

Смещение: 0x034

Регистр статуса прерываний

Формат регистра irq\_status приведен в следующей таблице:

**Таблица 28.9. Поля регистра irq\_status**

Разряды	Название	Описание	Сброс
31:20	-	Резерв	0

Разряды	Название	Описание	Сброс
19	cfg_err_hks	Ошибка конфигурации сессии hks:  Возникает совпадении условий: hash_size==0 hmac_key_dst != 0  При установке этого прерывания старт сессии hashCore будет отменен.	0
18	cfg_err_hkd	Ошибка конфигурации сессии hkd:  Зарезервированное значение поля session_ctr0.hmac_key_dst.  При установке этого прерывания старт сессии hashCore будет отменен.	0
17	cfg_err_cm	Ошибка конфигурации сессии cm:  Зарезервированное значение поля session_ctr0.crypt_mode.  При установке этого прерывания старт сессии cryptoCore будет отменен.	0
16	cfg_err_opc	Ошибка конфигурации сессии opc:  Включение режима вычисления имитовставки ОМАС-АСРКМ параллельно с блочным шифрованием: enable_crypt==1 enable_mac==1 mac_mode==1  При установке этого прерывания старт сессии cryptoCore будет отменен.	0
15	-	Резерв	0
14	hwch_write_err	Ошибка записи. Запись данных в полный буфер hwch fifo	0
13	cwch_write_err	Ошибка записи. Запись данных в полный буфер cwch fifo	0
12	crch_read_err	Ошибка чтения. Чтение из пустого буфера crch fifo	0
11	pdma_hwch_read	Попытка чтения через интерфейс записи в hashCore dma/m4.	0
10	pdma_crch_write	Попытка записи через интерфейс чтения в cryptoCore dma/m3.	0
9	pdma_cwch_read	Попытка чтения через интерфейс записи в cryptoCore dma/m2.	0
8	fifo_wrh_under_wm	Прерывание устанавливается, когда заполненность wrhFIFO ниже границы watermark.wrhFIFO.	0
7	fifo_wrh_empty	Прерывание устанавливается, когда wrhFIFO пуст.	0
6	fifo_wrc_under_wm	Прерывание устанавливается, когда заполненность wrcFIFO ниже границы watermark.wrcFIFO.	0
5	fifo_wrc_empty	Прерывание устанавливается, когда wrcFIFO пуст.	0
4	fifo_rdc_over_wm	Прерывание устанавливается, когда rdcFIFO выше границы watermark.rdcFIFO.	0

Разряды	Название	Описание	Сброс
3	fifo_rdc_full	Прерывание устанавливается, когда rdcFIFO полон.	0
2	eoh	Прерывание завершения сессии hashCore:  Прерывание устанавливается при в момент финализации сессии hashCore.	0
1	eok	Прерывание завершения работы cryptoCore на текущем ключе:  Прерывание устанавливается при включенной поддержке диверсификации ключей (session_ctr0.crypt_key_div_enable == 1) и завершении обработки последней транзакции на текущем ключе.	0
0	eoc	Прерывание завершения сессии cryptoCore:  Прерывание устанавливается при в момент финализации сессии cryptoCore.	0

### 28.3.8 Регистр irq\_clr

Смещение: 0x038

Регистр сброса прерываний. Запись 1 в поле сбрасывает соответствующее прерывание.

Формат регистра irq\_clr приведен в следующей таблице:

**Таблица 28.10. Поля регистра irq\_clr**

Разряды	Название	Описание	Сброс
31:20	-	Резерв	0
19	ccfg_err_hks		0
18	ccfg_err_hkd		0
17	ccfg_err_cm		0
16	ccfg_err_opc		0
15	-	Резерв	0
14	chwch_write_err		0
13	ccwch_write_err		0
12	ccrch_read_err		0
11	cpdma_hwch_read		0
10	cpdma_crch_write		0
9	cpdma_cwch_read		0
8:3	-	Резерв	0
2	ceoh		0
1	ceok		0
0	ceoc		0

### 28.3.9 Регистр irq\_mask

Смещение: 0x03C

Регистр маскирования прерываний. Запись 1 в поле запрещает формирование соответствующего прерывания.

Формат регистра `irq_mask` приведен в следующей таблице:

**Таблица 28.11. Поля регистра `irq_mask`**

Разряды	Название	Описание	Сброс
31:20	-	Резерв	0
19	<code>mcfg_err_hks</code>		0
18	<code>mcfg_err_hkd</code>		0
17	<code>mcfg_err_cm</code>		0
16	<code>mcfg_err_opc</code>		0
15	-	Резерв	0
14	<code>mhwch_write_err</code>		0
13	<code>mcwch_write_err</code>		0
12	<code>mcrch_read_err</code>		0
11:9	-	Резерв	0
8	<code>mfifo_wrh_under_wm</code>		1
7	<code>mfifo_wrh_empty</code>		1
6	<code>mfifo_wrc_under_wm</code>		1
5	<code>mfifo_wrc_empty</code>		1
4	<code>mfifo_rdc_over_wm</code>		1
3	<code>mfifo_rdc_full</code>		1
2	<code>meoh</code>		0
1	<code>meok</code>		0
0	<code>meoc</code>		0

### 28.3.10 Регистр `ctx0_crypt_block_size_low`

Смещение: `0x100`

Регистр принимает количество байт данных области блочного шифрования/вычисления MAC.

При старте или возобновлении сессии, в счетчик `crypt_block_cnt` записывается значение.  
`crypt_block_cnt = {crypt_block_size_high, crypt_block_size_low}`.

После чего, счетчик `crypt_block_cnt` декрементируется с каждой новой транзакцией в вычислитель `hash`.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание `eoc`.
2. Если `session_ctr0.crypt_finalize_resolve==1`, происходит финализация MAC-сессии.

Если сессия остановлена записью 1 в поле `session_ctr1.stop_crypt`, либо при возникновении прерывания `eok`, текущее значение счетчика будет доступно через регистры `crypt_block_size_*`.

Формат регистра `ctx0_crypt_block_size_low` приведен в следующей таблице:

**Таблица 28.12. Поля регистра `ctx0_crypt_block_size_low`**

Разряды	Название	Описание	Сброс
31:0	size	Младший регистр размера области данных для блочного шифрования/вычисления MAC.	0

### 28.3.11 Регистр `ctx0_crypt_block_size_high`

Смещение: 0x104

Регистр принимает количество байт данных области блочного шифрования/вычисления MAC.

При старте или возобновлении сессии, в счетчик `crypt_block_cnt` записывается значение.  
`crypt_block_cnt = {crypt_block_size_high, crypt_block_size_low}`.

После чего, счетчик `crypt_block_cnt` декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание `eoc`.
2. Если `session_ctr0.crypt_finalize_resolve==1`, происходит финализация MAC-сессии.

Если сессия остановлена записью 1 в поле `session_ctr1.stop_crypt`, либо при возникновении прерывания `eoc`, текущее значение счетчика будет доступно через регистры `crypt_block_size_*`.

Формат регистра `ctx0_crypt_block_size_high` приведен в следующей таблице:

**Таблица 28.13. Поля регистра `ctx0_crypt_block_size_high`**

Разряды	Название	Описание	Сброс
31:0	size	Старший регистр размера области данных для блочного шифрования/вычисления MAC.	0

### 28.3.12 Регистр `ctx0_crypt_block_size_per_key`

Смещение: 0x108

Регистр принимает количество байт данных области криптопреобразования на 1 ключ.

Если при старте сессии включена поддержка диверсификации ключа (`session_ctr0.crypt_key_div_enable==1`). То при старте или возобновлении сессии, в счетчик `crypt_key_cnt` записывается значение этого регистра. После чего, счетчик `crypt_key_cnt`

декрементируется с каждой новой транзакцией в вычислитель crypt.

По достижении счетчиком нуля, происходит следующее:

1. Вычислитель останавливается и прекращает принимать входные данные.
2. Устанавливается прерывание eok.

Если сессия остановлена записью 1 в поле session\_ctrl.stop\_crypt, текущее значение счетчика будет доступно через этот регистр.

Формат регистра ctx0\_crypt\_block\_size\_per\_key приведен в следующей таблице:

**Таблица 28.14. Поля регистра ctx0\_crypt\_block\_size\_per\_key**

Разряды	Название	Описание	Сброс
31:0	size	Регистр количества байт данных криптопреобразования на 1 ключ.	0

### 28.3.13 Регистр ctx0\_ctracpkm\_section\_size

Смещение: 0x10C

Регистр длины секции CTR-АСРКМ.

В режиме ОМАС-АСРКМ через этот регистр задается параметр T\* - частота смены ключей.

Формат регистра ctx0\_ctracpkm\_section\_size приведен в следующей таблице:

**Таблица 28.15. Поля регистра ctx0\_ctracpkm\_section\_size**

Разряды	Название	Описание	Сброс
31:0	size	CTR-АСРКМ: Регистр длины секции АСРКМ N в байтах. ОМАС-АСРКМ: Регистр частоты смены ключей T* в байтах.	0

### 28.3.14 Регистр ctx0\_omac\_section\_size

Смещение: 0x110

Регистр размера секции N ОМАС.

Формат регистра ctx0\_omac\_section\_size приведен в следующей таблице:

**Таблица 28.16. Поля регистра ctx0\_omac\_section\_size**

Разряды	Название	Описание	Сброс
31:0	size	Регистр размера секции N ОМАС в байтах	0

### 28.3.15 Регистр ctx0\_ctracpkm\_sec\_cnt

Смещение: 0x114

Регистр счетчика обработанных байт секции CTR-АСРКМ

Формат регистра ctx0\_ctracpkm\_sec\_cnt приведен в следующей таблице:

**Таблица 28.17. Поля регистра ctx0\_ctracpkm\_sec\_cnt**

Разряды	Название	Описание	Сброс
31:0	cnt	Регистр счетчика обработанных байт секции CTR-АСРКМ	0

### 28.3.16 Регистр ctx0\_omac\_key\_change\_cnt

Смещение: 0x118

Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе

Формат регистра ctx0\_omac\_key\_change\_cnt приведен в следующей таблице:

**Таблица 28.18. Поля регистра ctx0\_omac\_key\_change\_cnt**

Разряды	Название	Описание	Сброс
31:0	cnt	Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе	0

### 28.3.17 Регистр ctx0\_crypt\_key0

Смещение: 0x120

Регистр принимает биты crypt\_key[31:0] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[31:0]. В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[31:0] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key0 приведен в следующей таблице:

**Таблица 28.19. Поля регистра ctx0\_crypt\_key0**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[31:0] сессионного ключа блочного шифра.	0

### 28.3.18 Регистр ctx0\_crypt\_key1

Смещение: 0x124

Регистр принимает биты crypt\_key[63:32] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[63:32].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[63:32] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key1 приведен в следующей таблице:

**Таблица 28.20. Поля регистра ctx0\_crypt\_key1**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[63:32] сессионного ключа блочного шифра.	0

### 28.3.19 Регистр ctx0\_crypt\_key2

Смещение: 0x128

Регистр принимает биты crypt\_key[95:64] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[95:64].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[95:64] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key2 приведен в следующей таблице:

**Таблица 28.21. Поля регистра ctx0\_crypt\_key2**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[95:64] сессионного ключа блочного шифра.	0

### 28.3.20 Регистр ctx0\_crypt\_key3

Смещение: 0x12C

Регистр принимает биты crypt\_key[127:96] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[127:96].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[127:96] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key3 приведен в следующей таблице:



**Таблица 28.22. Поля регистра ctx0\_crypt\_key3**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[127:96] сессионного ключа блочного шифра.	0

**28.3.21 Регистр ctx0\_crypt\_key4**

Смещение:0x130

Регистр принимает биты crypt\_key[159:128] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[159:128].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[159:128] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key4 приведен в следующей таблице:

**Таблица 28.23. Поля регистра ctx0\_crypt\_key4**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[159:128] сессионного ключа блочного шифра.	0

**28.3.22 Регистр ctx0\_crypt\_key5**

Смещение:0x134

Регистр принимает биты crypt\_key[191:160] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[191:160].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[191:160] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key5 приведен в следующей таблице:

**Таблица 28.24. Поля регистра ctx0\_crypt\_key5**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[191:160] сессионного ключа блочного шифра.	0

**28.3.23 Регистр ctx0\_crypt\_key6**

Смещение:0x138

Регистр принимает биты crypt\_key[223:192] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[223:192].  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[223:192] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key6 приведен в следующей таблице:

**Таблица 28.25. Поля регистра ctx0\_crypt\_key6**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[223:192] сессионного ключа блочного шифра.	0

### 28.3.24 Регистр ctx0\_crypt\_key7

Смещение:0x13C

Регистр принимает биты crypt\_key[255:224] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[255:224].  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[255:224] текущего итерационного ключа.

Формат регистра ctx0\_crypt\_key7 приведен в следующей таблице:

**Таблица 28.26. Поля регистра ctx0\_crypt\_key7**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[255:224] сессионного ключа блочного шифра.	0

### 28.3.25 Регистр ctx0\_iv0

Смещение:0x140

Регистр хранит биты IV[31:0] инициализирующей последовательности.

Формат регистра ctx0\_iv0 приведен в следующей таблице:

**Таблица 28.27. Поля регистра ctx0\_iv0**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[31:0] инициализирующей последовательности.	0

### 28.3.26 Регистр ctx0\_iv1

Смещение:0x144

Регистр хранит биты IV[63:32] инициализирующей последовательности.

Формат регистра ctx0\_iv1 приведен в следующей таблице:

**Таблица 28.28. Поля регистра ctx0\_iv1**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[63:32] инициализирующей последовательности.	0

### 28.3.27 Регистр ctx0\_iv2

Смещение:0x148

Регистр хранит биты IV[95:64] инициализирующей последовательности.

Формат регистра ctx0\_iv2 приведен в следующей таблице:

**Таблица 28.29. Поля регистра ctx0\_iv2**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[95:64] инициализирующей последовательности.	0

### 28.3.28 Регистр ctx0\_iv3

Смещение:0x14C

Регистр хранит биты IV[127:96] инициализирующей последовательности.

Формат регистра ctx0\_iv3 приведен в следующей таблице:

**Таблица 28.30. Поля регистра ctx0\_iv3**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[127:96] инициализирующей последовательности.	0

### 28.3.29 Регистр ctx0\_iv4

Смещение:0x150

Регистр принимает биты IV[159:128] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[31:0] текущего дополнительного секционного ключа.

Формат регистра ctx0\_iv4 приведен в следующей таблице:

**Таблица 28.31. Поля регистра ctx0\_iv4**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[159:128] инициализирующей последовательности.	0

**28.3.30 Регистр ctx0\_iv5**

Смещение:0x154

Регистр принимает биты IV[191:160] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[63:32] текущего дополнительного секционного ключа.

Формат регистра ctx0\_iv5 приведен в следующей таблице:

**Таблица 28.32. Поля регистра ctx0\_iv5**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[191:160] инициализирующей последовательности.	0

**28.3.31 Регистр ctx0\_iv6**

Смещение:0x158

Регистр принимает биты IV[223:192] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[95:64] текущего дополнительного секционного ключа.

Формат регистра ctx0\_iv6 приведен в следующей таблице:

**Таблица 28.33. Поля регистра ctx0\_iv6**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[223:192] инициализирующей последовательности.	0

**28.3.32 Регистр ctx0\_iv7**

Смещение:0x15C

Регистр принимает биты IV[255:224] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[127:96] текущего дополнительного секционного ключа.

Формат регистра ctx0\_iv7 приведен в следующей таблице:

**Таблица 28.34. Поля регистра ctx0\_iv7**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[255:224] инициализирующей последовательности.	0

### 28.3.33 Регистр ctx0\_iv8

Смещение:0x160

Регистр принимает биты IV[287:256] инициализирующей последовательности.

Формат регистра ctx0\_iv8 приведен в следующей таблице:

**Таблица 28.35. Поля регистра ctx0\_iv8**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[287:256] инициализирующей последовательности.	0

### 28.3.34 Регистр ctx0\_iv9

Смещение:0x164

Регистр принимает биты IV[319:288] инициализирующей последовательности.

Формат регистра ctx0\_iv9 приведен в следующей таблице:

**Таблица 28.36. Поля регистра ctx0\_iv9**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[319:288] инициализирующей последовательности.	0

### 28.3.35 Регистр ctx0\_iv10

Смещение:0x168

Регистр принимает биты IV[351:320] инициализирующей последовательности.

Формат регистра ctx0\_iv10 приведен в следующей таблице:

**Таблица 28.37. Поля регистра ctx0\_iv10**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[351:320] инициализирующей последовательности.	0

### 28.3.36 Регистр ctx0\_iv11

Смещение: 0x16C

Регистр принимает биты IV[383:352] инициализирующей последовательности.

Формат регистра ctx0\_iv11 приведен в следующей таблице:

**Таблица 28.38. Поля регистра ctx0\_iv11**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[383:352] инициализирующей последовательности.	0

### 28.3.37 Регистр ctx0\_iv12

Смещение: 0x170

Регистр принимает биты IV[415:384] инициализирующей последовательности.

Формат регистра ctx0\_iv12 приведен в следующей таблице:

**Таблица 28.39. Поля регистра ctx0\_iv12**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[415:384] инициализирующей последовательности.	0

### 28.3.38 Регистр ctx0\_iv13

Смещение: 0x174

Регистр принимает биты IV[447:416] инициализирующей последовательности.

Формат регистра ctx0\_iv13 приведен в следующей таблице:

**Таблица 28.40. Поля регистра ctx0\_iv13**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[447:416] инициализирующей последовательности.	0

### 28.3.39 Регистр ctx0\_iv14

Смещение: 0x178

Регистр принимает биты IV[479:448] инициализирующей последовательности.

Формат регистра ctx0\_iv14 приведен в следующей таблице:

**Таблица 28.41. Поля регистра ctx0\_iv14**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[479:448] инициализирующей последовательности.	0

**28.3.40 Регистр ctx0\_iv15**

Смещение:0x17C

Регистр принимает биты IV[511:480] инициализирующей последовательности.

Формат регистра ctx0\_iv15 приведен в следующей таблице:

**Таблица 28.42. Поля регистра ctx0\_iv15**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[511:480] инициализирующей последовательности.	0

**28.3.41 Регистр ctx0\_mac\_key0**

Смещение:0x180

Регистр содержит биты mac[31:0] имитовставки.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[31:0] текущего секционного ключа.

Формат регистра ctx0\_mac\_key0 приведен в следующей таблице:

**Таблица 28.43. Поля регистра ctx0\_mac\_key0**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты mac[31:0] имитовставки	0

**28.3.42 Регистр ctx0\_mac\_key1**

Смещение:0x184

Регистр содержит биты hash[63:32] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[63:32] текущего секционного ключа.

Формат регистра ctx0\_mac\_key1 приведен в следующей таблице:

**Таблица 28.44. Поля регистра ctx0\_mac\_key1**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[63:32] текущего значения хэш-суммы.	0

**28.3.43 Регистр ctx0\_mac\_key2**

Смещение:0x188

Регистр содержит биты hash[95:64] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[95:64] текущего секционного ключа.

Формат регистра ctx0\_mac\_key2 приведен в следующей таблице:

**Таблица 28.45. Поля регистра ctx0\_mac\_key2**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[95:64] текущего значения хэш-суммы.	0

**28.3.44 Регистр ctx0\_mac\_key3**

Смещение:0x18C

Регистр содержит биты hash[127:96] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[127:96] текущего секционного ключа.

Формат регистра ctx0\_mac\_key3 приведен в следующей таблице:

**Таблица 28.46. Поля регистра ctx0\_mac\_key3**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[127:96] текущего значения хэш-суммы.	0

**28.3.45 Регистр ctx0\_mac\_key4**

Смещение:0x190

Регистр содержит биты hash[159:128] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[159:128] текущего секционного ключа.



Формат регистра ctx0\_mac\_key4 приведен в следующей таблице:

**Таблица 28.47. Поля регистра ctx0\_mac\_key4**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[159:128] текущего значения хэш-суммы.	0

### 28.3.46 Регистр ctx0\_mac\_key5

Смещение:0x194

Регистр содержит биты hash[191:160] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[191:160] текущего секционного ключа.

Формат регистра ctx0\_mac\_key5 приведен в следующей таблице:

**Таблица 28.48. Поля регистра ctx0\_mac\_key5**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[191:160] текущего значения хэш-суммы.	0

### 28.3.47 Регистр ctx0\_mac\_key6

Смещение:0x198

Регистр содержит биты hash[223:192] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[223:192] текущего секционного ключа.

Формат регистра ctx0\_mac\_key6 приведен в следующей таблице:

**Таблица 28.49. Поля регистра ctx0\_mac\_key6**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[223:192] текущего значения хэш-суммы.	0

### 28.3.48 Регистр ctx0\_mac\_key7

Смещение:0x19C

Регистр содержит биты hash[255:224] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[255:224] текущего секционного ключа.

Формат регистра ctx0\_mac\_key7 приведен в следующей таблице:

**Таблица 28.50. Поля регистра ctx0\_mac\_key7**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[255:224] текущего значения хэш-суммы.	0

### 28.3.49 Регистр ctx0\_mac0

Смещение: 0x1A0

Регистр содержит биты mac[31:0] имитовставки

Формат регистра ctx0\_mac0 приведен в следующей таблице:

**Таблица 28.51. Поля регистра ctx0\_mac0**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[31:0] имитовставки	0

### 28.3.50 Регистр ctx0\_mac1

Смещение: 0x1A4

Регистр содержит биты mac[63:32] имитовставки

Формат регистра ctx0\_mac1 приведен в следующей таблице:

**Таблица 28.52. Поля регистра ctx0\_mac1**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[63:32] имитовставки	0

### 28.3.51 Регистр ctx0\_mac2

Смещение: 0x1A8

Регистр содержит биты mac[95:64] имитовставки

Формат регистра ctx0\_mac2 приведен в следующей таблице:

**Таблица 28.53. Поля регистра ctx0\_mac2**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[95:64] имитовставки	0

### 28.3.52 Регистр ctx0\_mac3

Смещение: 0x1AC

Регистр содержит биты mac[127:96] имитовставки

Формат регистра ctx0\_mac3 приведен в следующей таблице:

**Таблица 28.54. Поля регистра ctx0\_mac3**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[127:96] имитовставки	0

### 28.3.53 Регистр ctx0\_hash\_block\_size\_low

Смещение: 0x200

Регистр принимает количество байт данных области для вычисления hash.

При старте или возобновлении сессии, в счетчик hash\_block\_cnt записывается значение.  
hash\_block\_cnt = {hash\_block\_size\_high, hash\_block\_size\_low}.

После чего, счетчик hash\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eoh.
2. Если session\_ctr0.hash\_finalize\_resolve==1, происходит финализация hash-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_hash, текущее значение счетчика будет доступно через регистры hash\_block\_size\_\*.

При включенной финализации, необходимо программно реализовать паддинг последнего блока данных и дополнить блок нулями до 64 байт.

Если размер последнего блока уже равен 64 байтам, то паддинг необходимо реализовать в дополнительном 64-байтном блоке.

В этом случае, необходимо заранее добавить 64 байта к значению hash\_block\_size

Формат регистра ctx0\_hash\_block\_size\_low приведен в следующей таблице:

**Таблица 28.55. Поля регистра ctx0\_hash\_block\_size\_low**

Разряды	Название	Описание	Сброс
31:0	size	Младший регистр размера области hashCore.	0

### 28.3.54 Регистр ctx0\_hash\_block\_size\_high

Смещение: 0x204

Регистр принимает количество байт данных области для вычисления hash.

При старте или возобновлении сессии, в счетчик hash\_block\_cnt записывается значение.  
hash\_block\_cnt = {hash\_block\_size\_high, hash\_block\_size\_low}.

После чего, счетчик hash\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eoh.
2. Если session\_ctr0.hash\_finalize\_resolve==1, происходит финализация hash-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_hash, текущее значение счетчика будет доступно через регистры hash\_block\_size\_\*.

При включенной финализации, необходимо программно реализовать паддинг последнего блока данных и дополнить блок нулями до 64 байт.

Если размер последнего блока уже равен 64 байтам, то паддинг необходимо реализовать в дополнительном 64-байтном блоке.

В этом случае, необходимо заранее добавить 64 байта к значению hash\_block\_size

Формат регистра ctx0\_hash\_block\_size\_high приведен в следующей таблице:

**Таблица 28.56. Поля регистра ctx0\_hash\_block\_size\_high**

Разряды	Название	Описание	Сброс
31:0	size	Старший регистр размера области для вычисления hash.	0

### 28.3.55 Регистр ctx0\_hash\_N\_low

Смещение: 0x208

Счетчик объема данных, обработанных hashCore в текущей сессии.

Формат регистра ctx0\_hash\_N\_low приведен в следующей таблице:

**Таблица 28.57. Поля регистра ctx0\_hash\_N\_low**

Разряды	Название	Описание	Сброс
31:0	N		0

### 28.3.56 Регистр ctx0\_hash\_N\_high

Смещение:0x20C

Счетчик объема данных, обработанных hashCore в текущей сессии.

Формат регистра ctx0\_hash\_N\_high приведен в следующей таблице:

**Таблица 28.58. Поля регистра ctx0\_hash\_N\_high**

Разряды	Название	Описание	Сброс
31:0	N		0

### 28.3.57 Регистр ctx0\_hash0

Смещение:0x210

Регистр содержит биты hash[31:0] текущего значения хэш-суммы.

Формат регистра ctx0\_hash0 приведен в следующей таблице:

**Таблица 28.59. Поля регистра ctx0\_hash0**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[31:0] текущего значения хэш-суммы.	0

### 28.3.58 Регистр ctx0\_hash1

Смещение:0x214

Регистр содержит биты hash[63:32] текущего значения хэш-суммы.

Формат регистра ctx0\_hash1 приведен в следующей таблице:

**Таблица 28.60. Поля регистра ctx0\_hash1**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[63:32] текущего значения хэш-суммы.	0

### 28.3.59 Регистр ctx0\_hash2

Смещение:0x218

Регистр содержит биты hash[95:64] текущего значения хэш-суммы.

Формат регистра ctx0\_hash2 приведен в следующей таблице:

**Таблица 28.61. Поля регистра ctx0\_hash2**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[95:64] текущего значения хэш-суммы.	0

**28.3.60 Регистр ctx0\_hash3**

Смещение:0x21C

Регистр содержит биты hash[127:96] текущего значения хэш-суммы.

Формат регистра ctx0\_hash3 приведен в следующей таблице:

**Таблица 28.62. Поля регистра ctx0\_hash3**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[127:96] текущего значения хэш-суммы.	0

**28.3.61 Регистр ctx0\_hash4**

Смещение:0x220

Регистр содержит биты hash[159:128] текущего значения хэш-суммы.

Формат регистра ctx0\_hash4 приведен в следующей таблице:

**Таблица 28.63. Поля регистра ctx0\_hash4**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[159:128] текущего значения хэш-суммы.	0

**28.3.62 Регистр ctx0\_hash5**

Смещение:0x224

Регистр содержит биты hash[191:160] текущего значения хэш-суммы.

Формат регистра ctx0\_hash5 приведен в следующей таблице:

**Таблица 28.64. Поля регистра ctx0\_hash5**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[191:160] текущего значения хэш-суммы.	0

### 28.3.63 Регистр ctx0\_hash6

Смещение:0x228

Регистр содержит биты hash[223:192] текущего значения хэш-суммы.

Формат регистра ctx0\_hash6 приведен в следующей таблице:

**Таблица 28.65. Поля регистра ctx0\_hash6**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[223:192] текущего значения хэш-суммы.	0

### 28.3.64 Регистр ctx0\_hash7

Смещение:0x22C

Регистр содержит биты hash[255:224] текущего значения хэш-суммы.

Формат регистра ctx0\_hash7 приведен в следующей таблице:

**Таблица 28.66. Поля регистра ctx0\_hash7**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[255:224] текущего значения хэш-суммы.	0

### 28.3.65 Регистр ctx0\_hash8

Смещение:0x230

Регистр содержит биты hash[287:256] текущего значения хэш-суммы.

Формат регистра ctx0\_hash8 приведен в следующей таблице:

**Таблица 28.67. Поля регистра ctx0\_hash8**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[287:256] текущего значения хэш-суммы.	0

### 28.3.66 Регистр ctx0\_hash9

Смещение:0x234

Регистр содержит биты hash[319:288] текущего значения хэш-суммы.

Формат регистра ctx0\_hash9 приведен в следующей таблице:

**Таблица 28.68. Поля регистра ctx0\_hash9**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[319:288] текущего значения хэш-суммы.	0

**28.3.67 Регистр ctx0\_hash10**

Смещение:0x238

Регистр содержит биты hash[351:320] текущего значения хэш-суммы.

Формат регистра ctx0\_hash10 приведен в следующей таблице:

**Таблица 28.69. Поля регистра ctx0\_hash10**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[351:320] текущего значения хэш-суммы.	0

**28.3.68 Регистр ctx0\_hash11**

Смещение:0x23C

Регистр содержит биты hash[383:352] текущего значения хэш-суммы.

Формат регистра ctx0\_hash11 приведен в следующей таблице:

**Таблица 28.70. Поля регистра ctx0\_hash11**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[383:352] текущего значения хэш-суммы.	0

**28.3.69 Регистр ctx0\_hash12**

Смещение:0x240

Регистр содержит биты hash[415:384] текущего значения хэш-суммы.

Формат регистра ctx0\_hash12 приведен в следующей таблице:

**Таблица 28.71. Поля регистра ctx0\_hash12**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[415:384] текущего значения хэш-суммы.	0



### 28.3.70 Регистр ctx0\_hash13

Смещение:0x244

Регистр содержит биты hash[447:416] текущего значения хэш-суммы.

Формат регистра ctx0\_hash13 приведен в следующей таблице:

**Таблица 28.72. Поля регистра ctx0\_hash13**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[447:416] текущего значения хэш-суммы.	0

### 28.3.71 Регистр ctx0\_hash14

Смещение:0x248

Регистр содержит биты hash[479:448] текущего значения хэш-суммы.

Формат регистра ctx0\_hash14 приведен в следующей таблице:

**Таблица 28.73. Поля регистра ctx0\_hash14**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[479:448] текущего значения хэш-суммы.	0

### 28.3.72 Регистр ctx0\_hash15

Смещение:0x24C

Регистр содержит биты hash[511:480] текущего значения хэш-суммы.

Формат регистра ctx0\_hash15 приведен в следующей таблице:

**Таблица 28.74. Поля регистра ctx0\_hash15**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[511:480] текущего значения хэш-суммы.	0

### 28.3.73 Регистр ctx0\_hmac\_key0

Смещение:0x250

Формат регистра ctx0\_hmac\_key0 приведен в следующей таблице:

**Таблица 28.75. Поля регистра ctx0\_hmac\_key0**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.74 Регистр ctx0\_hmac\_key1**

Смещение:0x254

Формат регистра ctx0\_hmac\_key1 приведен в следующей таблице:

**Таблица 28.76. Поля регистра ctx0\_hmac\_key1**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.75 Регистр ctx0\_hmac\_key2**

Смещение:0x258

Формат регистра ctx0\_hmac\_key2 приведен в следующей таблице:

**Таблица 28.77. Поля регистра ctx0\_hmac\_key2**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.76 Регистр ctx0\_hmac\_key3**

Смещение:0x25C

Формат регистра ctx0\_hmac\_key3 приведен в следующей таблице:

**Таблица 28.78. Поля регистра ctx0\_hmac\_key3**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.77 Регистр ctx0\_hmac\_key4**

Смещение:0x260

Формат регистра ctx0\_hmac\_key4 приведен в следующей таблице:

**Таблица 28.79. Поля регистра ctx0\_hmac\_key4**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.78 Регистр ctx0\_hmac\_key5

Смещение:0x264

Формат регистра ctx0\_hmac\_key5 приведен в следующей таблице:

**Таблица 28.80. Поля регистра ctx0\_hmac\_key5**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.79 Регистр ctx0\_hmac\_key6

Смещение:0x268

Формат регистра ctx0\_hmac\_key6 приведен в следующей таблице:

**Таблица 28.81. Поля регистра ctx0\_hmac\_key6**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.80 Регистр ctx0\_hmac\_key7

Смещение:0x26C

Формат регистра ctx0\_hmac\_key7 приведен в следующей таблице:

**Таблица 28.82. Поля регистра ctx0\_hmac\_key7**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.81 Регистр ctx0\_hmac\_key8

Смещение:0x270

Формат регистра ctx0\_hmac\_key8 приведен в следующей таблице:

**Таблица 28.83. Поля регистра ctx0\_hmac\_key8**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.82 Регистр ctx0\_hmac\_key9

Смещение:0x274

Формат регистра ctx0\_hmac\_key9 приведен в следующей таблице:

**Таблица 28.84. Поля регистра ctx0\_hmac\_key9**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.83 Регистр ctx0\_hmac\_key10

Смещение:0x278

Формат регистра ctx0\_hmac\_key10 приведен в следующей таблице:

**Таблица 28.85. Поля регистра ctx0\_hmac\_key10**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.84 Регистр ctx0\_hmac\_key11

Смещение:0x27C

Формат регистра ctx0\_hmac\_key11 приведен в следующей таблице:

**Таблица 28.86. Поля регистра ctx0\_hmac\_key11**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.85 Регистр ctx0\_hmac\_key12

Смещение:0x280

Формат регистра ctx0\_hmac\_key12 приведен в следующей таблице:

**Таблица 28.87. Поля регистра ctx0\_hmac\_key12**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.86 Регистр ctx0\_hmac\_key13

Смещение:0x284

Формат регистра ctx0\_hmac\_key13 приведен в следующей таблице:

**Таблица 28.88. Поля регистра ctx0\_hmac\_key13**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.87 Регистр ctx0\_hmac\_key14

Смещение:0x288

Формат регистра ctx0\_hmac\_key14 приведен в следующей таблице:

**Таблица 28.89. Поля регистра ctx0\_hmac\_key14**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.88 Регистр ctx0\_hmac\_key15

Смещение:0x28C

Формат регистра ctx0\_hmac\_key15 приведен в следующей таблице:

**Таблица 28.90. Поля регистра ctx0\_hmac\_key15**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.89 Регистр ctx0\_hash\_sig0**

Смещение:0x290

Формат регистра ctx0\_hash\_sig0 приведен в следующей таблице:

**Таблица 28.91. Поля регистра ctx0\_hash\_sig0**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.90 Регистр ctx0\_hash\_sig1**

Смещение:0x294

Формат регистра ctx0\_hash\_sig1 приведен в следующей таблице:

**Таблица 28.92. Поля регистра ctx0\_hash\_sig1**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.91 Регистр ctx0\_hash\_sig2**

Смещение:0x298

Формат регистра ctx0\_hash\_sig2 приведен в следующей таблице:

**Таблица 28.93. Поля регистра ctx0\_hash\_sig2**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.92 Регистр ctx0\_hash\_sig3**

Смещение:0x29C

Формат регистра ctx0\_hash\_sig3 приведен в следующей таблице:

**Таблица 28.94. Поля регистра ctx0\_hash\_sig3**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.93 Регистр ctx0\_hash\_sig4

Смещение:0x2A0

Формат регистра ctx0\_hash\_sig4 приведен в следующей таблице:

**Таблица 28.95. Поля регистра ctx0\_hash\_sig4**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.94 Регистр ctx0\_hash\_sig5

Смещение:0x2A4

Формат регистра ctx0\_hash\_sig5 приведен в следующей таблице:

**Таблица 28.96. Поля регистра ctx0\_hash\_sig5**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.95 Регистр ctx0\_hash\_sig6

Смещение:0x2A8

Формат регистра ctx0\_hash\_sig6 приведен в следующей таблице:

**Таблица 28.97. Поля регистра ctx0\_hash\_sig6**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.96 Регистр ctx0\_hash\_sig7

Смещение:0x2AC

Формат регистра ctx0\_hash\_sig7 приведен в следующей таблице:

**Таблица 28.98. Поля регистра ctx0\_hash\_sig7**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.97 Регистр ctx0\_hash\_sig8

Смещение:0x2B0

Формат регистра ctx0\_hash\_sig8 приведен в следующей таблице:

**Таблица 28.99. Поля регистра ctx0\_hash\_sig8**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.98 Регистр ctx0\_hash\_sig9

Смещение:0x2B4

Формат регистра ctx0\_hash\_sig9 приведен в следующей таблице:

**Таблица 28.100. Поля регистра ctx0\_hash\_sig9**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.99 Регистр ctx0\_hash\_sig10

Смещение:0x2B8

Формат регистра ctx0\_hash\_sig10 приведен в следующей таблице:

**Таблица 28.101. Поля регистра ctx0\_hash\_sig10**

Разряды	Название	Описание	Сброс
31:0	sig		0



### 28.3.100 Регистр ctx0\_hash\_sig11

Смещение:0x2BC

Формат регистра ctx0\_hash\_sig11 приведен в следующей таблице:

**Таблица 28.102. Поля регистра ctx0\_hash\_sig11**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.101 Регистр ctx0\_hash\_sig12

Смещение:0x2C0

Формат регистра ctx0\_hash\_sig12 приведен в следующей таблице:

**Таблица 28.103. Поля регистра ctx0\_hash\_sig12**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.102 Регистр ctx0\_hash\_sig13

Смещение:0x2C4

Формат регистра ctx0\_hash\_sig13 приведен в следующей таблице:

**Таблица 28.104. Поля регистра ctx0\_hash\_sig13**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.103 Регистр ctx0\_hash\_sig14

Смещение:0x2C8

Формат регистра ctx0\_hash\_sig14 приведен в следующей таблице:

**Таблица 28.105. Поля регистра ctx0\_hash\_sig14**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.104 Регистр ctx0\_hash\_sig15**

Смещение:0x2CC

Формат регистра ctx0\_hash\_sig15 приведен в следующей таблице:

**Таблица 28.106. Поля регистра ctx0\_hash\_sig15**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.105 Регистр ctx1\_crypt\_block\_size\_low**

Смещение:0x300

Регистр принимает количество байт данных области блочного шифрования/вычисления MAC.

При старте или возобновлении сессии, в счетчик crypt\_block\_cnt записывается значение.  
crypt\_block\_cnt = {crypt\_block\_size\_high, crypt\_block\_size\_low}.

После чего, счетчик crypt\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eos.
2. Если session\_ctr0.crypt\_finalize\_resolve==1, происходит финализация MAC-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_crypt, либо при возникновении прерывания eok, текущее значение счетчика будет доступно через регистры crypt\_block\_size\_\*.

Формат регистра ctx1\_crypt\_block\_size\_low приведен в следующей таблице:

**Таблица 28.107. Поля регистра ctx1\_crypt\_block\_size\_low**

Разряды	Название	Описание	Сброс
31:0	size	Младший регистр размера области данных для блочного шифрования/вычисления MAC.	0

### 28.3.106 Регистр ctx1\_crypt\_block\_size\_high

Смещение:0x304

Регистр принимает количество байт данных области блочного шифрования/вычисления MAC.

При старте или возобновлении сессии, в счетчик crypt\_block\_cnt записывается значение.  
crypt\_block\_cnt = {crypt\_block\_size\_high, crypt\_block\_size\_low}.

После чего, счетчик crypt\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eoc.
2. Если session\_ctr0.crypt\_finalize\_resolve==1, происходит финализация MAC-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_crypt, либо при возникновении прерывания eok, текущее значение счетчика будет доступно через регистры crypt\_block\_size\_\*.

Формат регистра ctx1\_crypt\_block\_size\_high приведен в следующей таблице:

**Таблица 28.108. Поля регистра ctx1\_crypt\_block\_size\_high**

Разряды	Название	Описание	Сброс
31:0	size	Старший регистр размера области данных для блочного шифрования/вычисления MAC.	0

### 28.3.107 Регистр ctx1\_crypt\_block\_size\_per\_key

Смещение:0x308

Регистр принимает количество байт данных области криптопреобразования на 1 ключ.

Если при старте сессии включена поддержка диверсификации ключа (session\_ctr0.crypt\_key\_div\_enable==1). То при старте или возобновлении сессии, в счетчик crypt\_key\_cnt записывается значение этого регистра. После чего, счетчик crypt\_key\_cnt декрементируется с каждой новой транзакцией в вычислитель crypt.

По достижении счетчиком нуля, происходит следующее:

1. Вычислитель останавливается и прекращает принимать входные данные.
2. Устанавливается прерывание eok.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_crypt, текущее значение счетчика будет доступно через этот регистр.

Формат регистра ctx1\_crypt\_block\_size\_per\_key приведен в следующей таблице:

**Таблица 28.109. Поля регистра ctx1\_crypt\_block\_size\_per\_key**

Разряды	Название	Описание	Сброс
31:0	size	Регистр количества байт данных криптопреобразования на 1 ключ.	0

### 28.3.108 Регистр ctx1\_ctracpkm\_section\_size

Смещение: 0x30C

Регистр длины секции CTR-АСРКМ.

В режиме ОМАС-АСРКМ через этот регистр задается параметр T\* - частота смены ключей.

Формат регистра ctx1\_ctracpkm\_section\_size приведен в следующей таблице:

**Таблица 28.110. Поля регистра ctx1\_ctracpkm\_section\_size**

Разряды	Название	Описание	Сброс
31:0	size	CTR-АСРКМ: Регистр длины секции АСРКМ N в байтах. ОМАС-АСРКМ: Регистр частоты смены ключей T* в байтах.	0

### 28.3.109 Регистр ctx1\_omac\_section\_size

Смещение: 0x310

Регистр размера секции N ОМАС.

Формат регистра ctx1\_omac\_section\_size приведен в следующей таблице:

**Таблица 28.111. Поля регистра ctx1\_omac\_section\_size**

Разряды	Название	Описание	Сброс
31:0	size	Регистр размера секции N ОМАС в байтах	0

### 28.3.110 Регистр ctx1\_ctracpkm\_sec\_cnt

Смещение: 0x314

Регистр счетчика обработанных байт секции CTR-АСРКМ

Формат регистра ctx1\_ctracpkm\_sec\_cnt приведен в следующей таблице:

**Таблица 28.112. Поля регистра ctx1\_ctracpkm\_sec\_cnt**

Разряды	Название	Описание	Сброс
31:0	cnt	Регистр счетчика обработанных байт секции CTR-АСРКМ	0

**28.3.111 Регистр ctx1\_omac\_key\_change\_cnt**

Смещение:0x318

Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе

Формат регистра ctx1\_omac\_key\_change\_cnt приведен в следующей таблице:

**Таблица 28.113. Поля регистра ctx1\_omac\_key\_change\_cnt**

Разряды	Название	Описание	Сброс
31:0	cnt	Регистр счетчика обработанных слов данных в режиме ОМАС-АСРКМ на текущем мастер-ключе	0

**28.3.112 Регистр ctx1\_crypt\_key0**

Смещение:0x320

Регистр принимает биты crypt\_key[31:0] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[31:0].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[31:0] текущего итерационного ключа.

Формат регистра ctx1\_crypt\_key0 приведен в следующей таблице:

**Таблица 28.114. Поля регистра ctx1\_crypt\_key0**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[31:0] сессионного ключа блочного шифра.	0

**28.3.113 Регистр ctx1\_crypt\_key1**

Смещение:0x324

Регистр принимает биты crypt\_key[63:32] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[63:32].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты `key_iter[63:32]` текущего итерационного ключа.

Формат регистра `ctx1_crypt_key1` приведен в следующей таблице:

**Таблица 28.115. Поля регистра `ctx1_crypt_key1`**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты <code>crypt_key[63:32]</code> сессионного ключа блочного шифра.	0

### 28.3.114 Регистр `ctx1_crypt_key2`

Смещение: 0x328

Регистр принимает биты `crypt_key[95:64]` сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа `master_key[95:64]`.  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты `key_iter[95:64]` текущего итерационного ключа.

Формат регистра `ctx1_crypt_key2` приведен в следующей таблице:

**Таблица 28.116. Поля регистра `ctx1_crypt_key2`**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты <code>crypt_key[95:64]</code> сессионного ключа блочного шифра.	0

### 28.3.115 Регистр `ctx1_crypt_key3`

Смещение: 0x32C

Регистр принимает биты `crypt_key[127:96]` сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа `master_key[127:96]`.  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты `key_iter[127:96]` текущего итерационного ключа.

Формат регистра `ctx1_crypt_key3` приведен в следующей таблице:

**Таблица 28.117. Поля регистра `ctx1_crypt_key3`**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты <code>crypt_key[127:96]</code> сессионного ключа блочного шифра.	0

### 28.3.116 Регистр ctx1\_crypt\_key4

Смещение:0x330

Регистр принимает биты crypt\_key[159:128] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[159:128].  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[159:128] текущего итерационного ключа.

Формат регистра ctx1\_crypt\_key4 приведен в следующей таблице:

**Таблица 28.118. Поля регистра ctx1\_crypt\_key4**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[159:128] сессионного ключа блочного шифра.	0

### 28.3.117 Регистр ctx1\_crypt\_key5

Смещение:0x334

Регистр принимает биты crypt\_key[191:160] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[191:160].  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[191:160] текущего итерационного ключа.

Формат регистра ctx1\_crypt\_key5 приведен в следующей таблице:

**Таблица 28.119. Поля регистра ctx1\_crypt\_key5**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[191:160] сессионного ключа блочного шифра.	0

### 28.3.118 Регистр ctx1\_crypt\_key6

Смещение:0x338

Регистр принимает биты crypt\_key[223:192] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[223:192].  
В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[223:192] текущего итерационного ключа.

Формат регистра ctx1\_crypt\_key6 приведен в следующей таблице:

**Таблица 28.120. Поля регистра ctx1\_crypt\_key6**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[223:192] сессионного ключа блочного шифра.	0

**28.3.119 Регистр ctx1\_crypt\_key7**

Смещение:0x33C

Регистр принимает биты crypt\_key[255:224] сессионного ключа блочного шифра.

В режиме ОМАС-АСРКМ регистр принимает биты мастер ключа master\_key[255:224].

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_iter[255:224] текущего итерационного ключа.

Формат регистра ctx1\_crypt\_key7 приведен в следующей таблице:

**Таблица 28.121. Поля регистра ctx1\_crypt\_key7**

Разряды	Название	Описание	Сброс
31:0	key	Регистр принимает биты crypt_key[255:224] сессионного ключа блочного шифра.	0

**28.3.120 Регистр ctx1\_iv0**

Смещение:0x340

Регистр хранит биты IV[31:0] инициализирующей последовательности.

Формат регистра ctx1\_iv0 приведен в следующей таблице:

**Таблица 28.122. Поля регистра ctx1\_iv0**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[31:0] инициализирующей последовательности.	0

**28.3.121 Регистр ctx1\_iv1**

Смещение:0x344

Регистр хранит биты IV[63:32] инициализирующей последовательности.

Формат регистра ctx1\_iv1 приведен в следующей таблице:



**Таблица 28.123. Поля регистра ctx1\_iv1**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[63:32] инициализирующей последовательности.	0

**28.3.122 Регистр ctx1\_iv2**

Смещение:0x348

Регистр хранит биты IV[95:64] инициализирующей последовательности.

Формат регистра ctx1\_iv2 приведен в следующей таблице:

**Таблица 28.124. Поля регистра ctx1\_iv2**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[95:64] инициализирующей последовательности.	0

**28.3.123 Регистр ctx1\_iv3**

Смещение:0x34C

Регистр хранит биты IV[127:96] инициализирующей последовательности.

Формат регистра ctx1\_iv3 приведен в следующей таблице:

**Таблица 28.125. Поля регистра ctx1\_iv3**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр хранит биты IV[127:96] инициализирующей последовательности.	0

**28.3.124 Регистр ctx1\_iv4**

Смещение:0x350

Регистр принимает биты IV[159:128] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[31:0] текущего дополнительного секционного ключа.

Формат регистра ctx1\_iv4 приведен в следующей таблице:

**Таблица 28.126. Поля регистра ctx1\_iv4**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[159:128] инициализирующей последовательности.	0

### 28.3.125 Регистр ctx1\_iv5

Смещение: 0x354

Регистр принимает биты IV[191:160] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[63:32] текущего дополнительного секционного ключа.

Формат регистра ctx1\_iv5 приведен в следующей таблице:

**Таблица 28.127. Поля регистра ctx1\_iv5**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[191:160] инициализирующей последовательности.	0

### 28.3.126 Регистр ctx1\_iv6

Смещение: 0x358

Регистр принимает биты IV[223:192] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[95:64] текущего дополнительного секционного ключа.

Формат регистра ctx1\_iv6 приведен в следующей таблице:

**Таблица 28.128. Поля регистра ctx1\_iv6**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[223:192] инициализирующей последовательности.	0

### 28.3.127 Регистр ctx1\_iv7

Смещение: 0x35C

Регистр принимает биты IV[255:224] инициализирующей последовательности.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key1\_sec[127:96] текущего дополнительного секционного ключа.

Формат регистра ctx1\_iv7 приведен в следующей таблице:

**Таблица 28.129. Поля регистра ctx1\_iv7**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[255:224] инициализирующей последовательности.	0

**28.3.128 Регистр ctx1\_iv8**

Смещение:0x360

Регистр принимает биты IV[287:256] инициализирующей последовательности.

Формат регистра ctx1\_iv8 приведен в следующей таблице:

**Таблица 28.130. Поля регистра ctx1\_iv8**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[287:256] инициализирующей последовательности.	0

**28.3.129 Регистр ctx1\_iv9**

Смещение:0x364

Регистр принимает биты IV[319:288] инициализирующей последовательности.

Формат регистра ctx1\_iv9 приведен в следующей таблице:

**Таблица 28.131. Поля регистра ctx1\_iv9**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[319:288] инициализирующей последовательности.	0

**28.3.130 Регистр ctx1\_iv10**

Смещение:0x368

Регистр принимает биты IV[351:320] инициализирующей последовательности.

Формат регистра ctx1\_iv10 приведен в следующей таблице:

**Таблица 28.132. Поля регистра ctx1\_iv10**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[351:320] инициализирующей последовательности.	0

### 28.3.131 Регистр ctx1\_iv11

Смещение: 0x36C

Регистр принимает биты IV[383:352] инициализирующей последовательности.

Формат регистра ctx1\_iv11 приведен в следующей таблице:

**Таблица 28.133. Поля регистра ctx1\_iv11**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[383:352] инициализирующей последовательности.	0

### 28.3.132 Регистр ctx1\_iv12

Смещение: 0x370

Регистр принимает биты IV[415:384] инициализирующей последовательности.

Формат регистра ctx1\_iv12 приведен в следующей таблице:

**Таблица 28.134. Поля регистра ctx1\_iv12**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[415:384] инициализирующей последовательности.	0

### 28.3.133 Регистр ctx1\_iv13

Смещение: 0x374

Регистр принимает биты IV[447:416] инициализирующей последовательности.

Формат регистра ctx1\_iv13 приведен в следующей таблице:

**Таблица 28.135. Поля регистра ctx1\_iv13**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[447:416] инициализирующей последовательности.	0

### 28.3.134 Регистр ctx1\_iv14

Смещение: 0x378

Регистр принимает биты IV[479:448] инициализирующей последовательности.

Формат регистра ctx1\_iv14 приведен в следующей таблице:

**Таблица 28.136. Поля регистра ctx1\_iv14**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[479:448] инициализирующей последовательности.	0

**28.3.135 Регистр ctx1\_iv15**

Смещение:0x37C

Регистр принимает биты IV[511:480] инициализирующей последовательности.

Формат регистра ctx1\_iv15 приведен в следующей таблице:

**Таблица 28.137. Поля регистра ctx1\_iv15**

Разряды	Название	Описание	Сброс
31:0	iv	Регистр принимает биты IV[511:480] инициализирующей последовательности.	0

**28.3.136 Регистр ctx1\_mac\_key0**

Смещение:0x380

Регистр содержит биты mac[31:0] имитовставки.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[31:0] текущего секционного ключа.

Формат регистра ctx1\_mac\_key0 приведен в следующей таблице:

**Таблица 28.138. Поля регистра ctx1\_mac\_key0**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты mac[31:0] имитовставки	0

**28.3.137 Регистр ctx1\_mac\_key1**

Смещение:0x384

Регистр содержит биты hash[63:32] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[63:32] текущего секционного ключа.

Формат регистра ctx1\_mac\_key1 приведен в следующей таблице:

**Таблица 28.139. Поля регистра ctx1\_mac\_key1**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[63:32] текущего значения хэш-суммы.	0

**28.3.138 Регистр ctx1\_mac\_key2**

Смещение:0x388

Регистр содержит биты hash[95:64] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[95:64] текущего секционного ключа.

Формат регистра ctx1\_mac\_key2 приведен в следующей таблице:

**Таблица 28.140. Поля регистра ctx1\_mac\_key2**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[95:64] текущего значения хэш-суммы.	0

**28.3.139 Регистр ctx1\_mac\_key3**

Смещение:0x38C

Регистр содержит биты hash[127:96] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[127:96] текущего секционного ключа.

Формат регистра ctx1\_mac\_key3 приведен в следующей таблице:

**Таблица 28.141. Поля регистра ctx1\_mac\_key3**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[127:96] текущего значения хэш-суммы.	0

**28.3.140 Регистр ctx1\_mac\_key4**

Смещение:0x390

Регистр содержит биты hash[159:128] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[159:128] текущего секционного ключа.

Формат регистра ctx1\_mac\_key4 приведен в следующей таблице:

**Таблица 28.142. Поля регистра ctx1\_mac\_key4**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[159:128] текущего значения хэш-суммы.	0

### 28.3.141 Регистр ctx1\_mac\_key5

Смещение:0x394

Регистр содержит биты hash[191:160] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[191:160] текущего секционного ключа.

Формат регистра ctx1\_mac\_key5 приведен в следующей таблице:

**Таблица 28.143. Поля регистра ctx1\_mac\_key5**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[191:160] текущего значения хэш-суммы.	0

### 28.3.142 Регистр ctx1\_mac\_key6

Смещение:0x398

Регистр содержит биты hash[223:192] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[223:192] текущего секционного ключа.

Формат регистра ctx1\_mac\_key6 приведен в следующей таблице:

**Таблица 28.144. Поля регистра ctx1\_mac\_key6**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[223:192] текущего значения хэш-суммы.	0

### 28.3.143 Регистр ctx1\_mac\_key7

Смещение:0x39C

Регистр содержит биты hash[255:224] текущего значения хэш-суммы.

В процессе криптопреобразования ОМАС-АСРКМ, в регистр отображаются биты key\_sec[255:224] текущего секционного ключа.

Формат регистра ctx1\_mac\_key7 приведен в следующей таблице:

**Таблица 28.145. Поля регистра ctx1\_mac\_key7**

Разряды	Название	Описание	Сброс
31:0	key	Регистр содержит биты hash[255:224] текущего значения хэш-суммы.	0

### 28.3.144 Регистр ctx1\_mac0

Смещение:0x3A0

Регистр содержит биты mac[31:0] имитовставки

Формат регистра ctx1\_mac0 приведен в следующей таблице:

**Таблица 28.146. Поля регистра ctx1\_mac0**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[31:0] имитовставки	0

### 28.3.145 Регистр ctx1\_mac1

Смещение:0x3A4

Регистр содержит биты mac[63:32] имитовставки

Формат регистра ctx1\_mac1 приведен в следующей таблице:

**Таблица 28.147. Поля регистра ctx1\_mac1**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[63:32] имитовставки	0

### 28.3.146 Регистр ctx1\_mac2

Смещение:0x3A8

Регистр содержит биты mac[95:64] имитовставки

Формат регистра ctx1\_mac2 приведен в следующей таблице:

**Таблица 28.148. Поля регистра ctx1\_mac2**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[95:64] имитовставки	0



### 28.3.147 Регистр ctx1\_mac3

Смещение: 0x3AC

Регистр содержит биты mac[127:96] имитовставки

Формат регистра ctx1\_mac3 приведен в следующей таблице:

**Таблица 28.149. Поля регистра ctx1\_mac3**

Разряды	Название	Описание	Сброс
31:0	mac	Регистр содержит биты mac[127:96] имитовставки	0

### 28.3.148 Регистр ctx1\_hash\_block\_size\_low

Смещение: 0x400

Регистр принимает количество байт данных области для вычисления hash.

При старте или возобновлении сессии, в счетчик hash\_block\_cnt записывается значение.  
hash\_block\_cnt = {hash\_block\_size\_high, hash\_block\_size\_low}.

После чего, счетчик hash\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eoh.
2. Если session\_ctr0.hash\_finalize\_resolve==1, происходит финализация hash-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_hash, текущее значение счетчика будет доступно через регистры hash\_block\_size\_\*.

При включенной финализации, необходимо программно реализовать паддинг последнего блока данных и дополнить блок нулями до 64 байт.

Если размер последнего блока уже равен 64 байтам, то паддинг необходимо реализовать в дополнительном 64-байтном блоке.

В этом случае, необходимо заранее добавить 64 байта к значению hash\_block\_size

Формат регистра ctx1\_hash\_block\_size\_low приведен в следующей таблице:

**Таблица 28.150. Поля регистра ctx1\_hash\_block\_size\_low**

Разряды	Название	Описание	Сброс
31:0	size	Младший регистр размера области hashCore.	0

### 28.3.149 Регистр ctx1\_hash\_block\_size\_high

Смещение: 0x404

Регистр принимает количество байт данных области для вычисления hash.

При старте или возобновлении сессии, в счетчик hash\_block\_cnt записывается значение.  
hash\_block\_cnt = {hash\_block\_size\_high, hash\_block\_size\_low}.

После чего, счетчик hash\_block\_cnt декрементируется с каждой новой транзакцией в вычислитель hash.

По достижении счетчиком нуля, происходит следующее:

1. Устанавливается прерывание eoh.
2. Если session\_ctr0.hash\_finalize\_resolve==1, происходит финализация hash-сессии.

Если сессия остановлена записью 1 в поле session\_ctr1.stop\_hash, текущее значение счетчика будет доступно через регистры hash\_block\_size\_\*.

При включенной финализации, необходимо программно реализовать паддинг последнего блока данных и дополнить блок нулями до 64 байт.

Если размер последнего блока уже равен 64 байтам, то паддинг необходимо реализовать в дополнительном 64-байтном блоке.

В этом случае, необходимо заранее добавить 64 байта к значению hash\_block\_size

Формат регистра ctx1\_hash\_block\_size\_high приведен в следующей таблице:

**Таблица 28.151. Поля регистра ctx1\_hash\_block\_size\_high**

Разряды	Название	Описание	Сброс
31:0	size	Старший регистр размера области для вычисления hash.	0

### 28.3.150 Регистр ctx1\_hash\_N\_low

Смещение: 0x408

Счетчик объема данных, обработанных hashCore в текущей сессии.

Формат регистра ctx1\_hash\_N\_low приведен в следующей таблице:

**Таблица 28.152. Поля регистра ctx1\_hash\_N\_low**

Разряды	Название	Описание	Сброс
31:0	N		0

### 28.3.151 Регистр ctx1\_hash\_N\_high

Смещение: 0x40C

Счетчик объема данных, обработанных hashCore в текущей сессии.

Формат регистра ctx1\_hash\_N\_high приведен в следующей таблице:

**Таблица 28.153. Поля регистра ctx1\_hash\_N\_high**

Разряды	Название	Описание	Сброс
31:0	N		0

### 28.3.152 Регистр ctx1\_hash0

Смещение: 0x410

Регистр содержит биты hash[31:0] текущего значения хэш-суммы.

Формат регистра ctx1\_hash0 приведен в следующей таблице:

**Таблица 28.154. Поля регистра ctx1\_hash0**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[31:0] текущего значения хэш-суммы.	0

### 28.3.153 Регистр ctx1\_hash1

Смещение: 0x414

Регистр содержит биты hash[63:32] текущего значения хэш-суммы.

Формат регистра ctx1\_hash1 приведен в следующей таблице:

**Таблица 28.155. Поля регистра ctx1\_hash1**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[63:32] текущего значения хэш-суммы.	0

### 28.3.154 Регистр ctx1\_hash2

Смещение: 0x418

Регистр содержит биты hash[95:64] текущего значения хэш-суммы.

Формат регистра ctx1\_hash2 приведен в следующей таблице:

**Таблица 28.156. Поля регистра ctx1\_hash2**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[95:64] текущего значения хэш-суммы.	0

**28.3.155 Регистр ctx1\_hash3**

Смещение:0x41C

Регистр содержит биты hash[127:96] текущего значения хэш-суммы.

Формат регистра ctx1\_hash3 приведен в следующей таблице:

**Таблица 28.157. Поля регистра ctx1\_hash3**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[127:96] текущего значения хэш-суммы.	0

**28.3.156 Регистр ctx1\_hash4**

Смещение:0x420

Регистр содержит биты hash[159:128] текущего значения хэш-суммы.

Формат регистра ctx1\_hash4 приведен в следующей таблице:

**Таблица 28.158. Поля регистра ctx1\_hash4**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[159:128] текущего значения хэш-суммы.	0

**28.3.157 Регистр ctx1\_hash5**

Смещение:0x424

Регистр содержит биты hash[191:160] текущего значения хэш-суммы.

Формат регистра ctx1\_hash5 приведен в следующей таблице:

**Таблица 28.159. Поля регистра ctx1\_hash5**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[191:160] текущего значения хэш-суммы.	0

### 28.3.158 Регистр ctx1\_hash6

Смещение:0x428

Регистр содержит биты hash[223:192] текущего значения хэш-суммы.

Формат регистра ctx1\_hash6 приведен в следующей таблице:

**Таблица 28.160. Поля регистра ctx1\_hash6**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[223:192] текущего значения хэш-суммы.	0

### 28.3.159 Регистр ctx1\_hash7

Смещение:0x42C

Регистр содержит биты hash[255:224] текущего значения хэш-суммы.

Формат регистра ctx1\_hash7 приведен в следующей таблице:

**Таблица 28.161. Поля регистра ctx1\_hash7**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[255:224] текущего значения хэш-суммы.	0

### 28.3.160 Регистр ctx1\_hash8

Смещение:0x430

Регистр содержит биты hash[287:256] текущего значения хэш-суммы.

Формат регистра ctx1\_hash8 приведен в следующей таблице:

**Таблица 28.162. Поля регистра ctx1\_hash8**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[287:256] текущего значения хэш-суммы.	0

### 28.3.161 Регистр ctx1\_hash9

Смещение:0x434

Регистр содержит биты hash[319:288] текущего значения хэш-суммы.

Формат регистра ctx1\_hash9 приведен в следующей таблице:

**Таблица 28.163. Поля регистра ctx1\_hash9**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[319:288] текущего значения хэш-суммы.	0

**28.3.162 Регистр ctx1\_hash10**

Смещение:0x438

Регистр содержит биты hash[351:320] текущего значения хэш-суммы.

Формат регистра ctx1\_hash10 приведен в следующей таблице:

**Таблица 28.164. Поля регистра ctx1\_hash10**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[351:320] текущего значения хэш-суммы.	0

**28.3.163 Регистр ctx1\_hash11**

Смещение:0x43C

Регистр содержит биты hash[383:352] текущего значения хэш-суммы.

Формат регистра ctx1\_hash11 приведен в следующей таблице:

**Таблица 28.165. Поля регистра ctx1\_hash11**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[383:352] текущего значения хэш-суммы.	0

**28.3.164 Регистр ctx1\_hash12**

Смещение:0x440

Регистр содержит биты hash[415:384] текущего значения хэш-суммы.

Формат регистра ctx1\_hash12 приведен в следующей таблице:

**Таблица 28.166. Поля регистра ctx1\_hash12**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[415:384] текущего значения хэш-суммы.	0

### 28.3.165 Регистр ctx1\_hash13

Смещение:0x444

Регистр содержит биты hash[447:416] текущего значения хэш-суммы.

Формат регистра ctx1\_hash13 приведен в следующей таблице:

**Таблица 28.167. Поля регистра ctx1\_hash13**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[447:416] текущего значения хэш-суммы.	0

### 28.3.166 Регистр ctx1\_hash14

Смещение:0x448

Регистр содержит биты hash[479:448] текущего значения хэш-суммы.

Формат регистра ctx1\_hash14 приведен в следующей таблице:

**Таблица 28.168. Поля регистра ctx1\_hash14**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[479:448] текущего значения хэш-суммы.	0

### 28.3.167 Регистр ctx1\_hash15

Смещение:0x44C

Регистр содержит биты hash[511:480] текущего значения хэш-суммы.

Формат регистра ctx1\_hash15 приведен в следующей таблице:

**Таблица 28.169. Поля регистра ctx1\_hash15**

Разряды	Название	Описание	Сброс
31:0	hash	Регистр содержит биты hash[511:480] текущего значения хэш-суммы.	0

### 28.3.168 Регистр ctx1\_hmac\_key0

Смещение:0x450

Формат регистра ctx1\_hmac\_key0 приведен в следующей таблице:

**Таблица 28.170. Поля регистра ctx1\_hmac\_key0**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.169 Регистр ctx1\_hmac\_key1**

Смещение:0x454

Формат регистра ctx1\_hmac\_key1 приведен в следующей таблице:

**Таблица 28.171. Поля регистра ctx1\_hmac\_key1**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.170 Регистр ctx1\_hmac\_key2**

Смещение:0x458

Формат регистра ctx1\_hmac\_key2 приведен в следующей таблице:

**Таблица 28.172. Поля регистра ctx1\_hmac\_key2**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.171 Регистр ctx1\_hmac\_key3**

Смещение:0x45C

Формат регистра ctx1\_hmac\_key3 приведен в следующей таблице:

**Таблица 28.173. Поля регистра ctx1\_hmac\_key3**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.172 Регистр ctx1\_hmac\_key4**

Смещение:0x460



Формат регистра ctx1\_hmac\_key4 приведен в следующей таблице:

**Таблица 28.174. Поля регистра ctx1\_hmac\_key4**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.173 Регистр ctx1\_hmac\_key5

Смещение:0x464

Формат регистра ctx1\_hmac\_key5 приведен в следующей таблице:

**Таблица 28.175. Поля регистра ctx1\_hmac\_key5**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.174 Регистр ctx1\_hmac\_key6

Смещение:0x468

Формат регистра ctx1\_hmac\_key6 приведен в следующей таблице:

**Таблица 28.176. Поля регистра ctx1\_hmac\_key6**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.175 Регистр ctx1\_hmac\_key7

Смещение:0x46C

Формат регистра ctx1\_hmac\_key7 приведен в следующей таблице:

**Таблица 28.177. Поля регистра ctx1\_hmac\_key7**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.176 Регистр ctx1\_hmac\_key8

Смещение:0x470

Формат регистра ctx1\_hmac\_key8 приведен в следующей таблице:

**Таблица 28.178. Поля регистра ctx1\_hmac\_key8**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.177 Регистр ctx1\_hmac\_key9

Смещение:0x474

Формат регистра ctx1\_hmac\_key9 приведен в следующей таблице:

**Таблица 28.179. Поля регистра ctx1\_hmac\_key9**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.178 Регистр ctx1\_hmac\_key10

Смещение:0x478

Формат регистра ctx1\_hmac\_key10 приведен в следующей таблице:

**Таблица 28.180. Поля регистра ctx1\_hmac\_key10**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.179 Регистр ctx1\_hmac\_key11

Смещение:0x47C

Формат регистра ctx1\_hmac\_key11 приведен в следующей таблице:

**Таблица 28.181. Поля регистра ctx1\_hmac\_key11**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.180 Регистр ctx1\_hmac\_key12

Смещение:0x480

Формат регистра ctx1\_hmac\_key12 приведен в следующей таблице:

**Таблица 28.182. Поля регистра ctx1\_hmac\_key12**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.181 Регистр ctx1\_hmac\_key13

Смещение:0x484

Формат регистра ctx1\_hmac\_key13 приведен в следующей таблице:

**Таблица 28.183. Поля регистра ctx1\_hmac\_key13**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.182 Регистр ctx1\_hmac\_key14

Смещение:0x488

Формат регистра ctx1\_hmac\_key14 приведен в следующей таблице:

**Таблица 28.184. Поля регистра ctx1\_hmac\_key14**

Разряды	Название	Описание	Сброс
31:0	key		0

### 28.3.183 Регистр ctx1\_hmac\_key15

Смещение:0x48C

Формат регистра ctx1\_hmac\_key15 приведен в следующей таблице:

**Таблица 28.185. Поля регистра ctx1\_hmac\_key15**

Разряды	Название	Описание	Сброс
31:0	key		0

**28.3.184 Регистр ctx1\_hash\_sig0**

Смещение:0x490

Формат регистра ctx1\_hash\_sig0 приведен в следующей таблице:

**Таблица 28.186. Поля регистра ctx1\_hash\_sig0**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.185 Регистр ctx1\_hash\_sig1**

Смещение:0x494

Формат регистра ctx1\_hash\_sig1 приведен в следующей таблице:

**Таблица 28.187. Поля регистра ctx1\_hash\_sig1**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.186 Регистр ctx1\_hash\_sig2**

Смещение:0x498

Формат регистра ctx1\_hash\_sig2 приведен в следующей таблице:

**Таблица 28.188. Поля регистра ctx1\_hash\_sig2**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.187 Регистр ctx1\_hash\_sig3**

Смещение:0x49C

Формат регистра ctx1\_hash\_sig3 приведен в следующей таблице:

**Таблица 28.189. Поля регистра ctx1\_hash\_sig3**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.188 Регистр ctx1\_hash\_sig4

Смещение:0x4A0

Формат регистра ctx1\_hash\_sig4 приведен в следующей таблице:

**Таблица 28.190. Поля регистра ctx1\_hash\_sig4**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.189 Регистр ctx1\_hash\_sig5

Смещение:0x4A4

Формат регистра ctx1\_hash\_sig5 приведен в следующей таблице:

**Таблица 28.191. Поля регистра ctx1\_hash\_sig5**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.190 Регистр ctx1\_hash\_sig6

Смещение:0x4A8

Формат регистра ctx1\_hash\_sig6 приведен в следующей таблице:

**Таблица 28.192. Поля регистра ctx1\_hash\_sig6**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.191 Регистр ctx1\_hash\_sig7

Смещение:0x4AC

Формат регистра ctx1\_hash\_sig7 приведен в следующей таблице:

**Таблица 28.193. Поля регистра ctx1\_hash\_sig7**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.192 Регистр ctx1\_hash\_sig8

Смещение:0x4B0

Формат регистра ctx1\_hash\_sig8 приведен в следующей таблице:

**Таблица 28.194. Поля регистра ctx1\_hash\_sig8**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.193 Регистр ctx1\_hash\_sig9

Смещение:0x4B4

Формат регистра ctx1\_hash\_sig9 приведен в следующей таблице:

**Таблица 28.195. Поля регистра ctx1\_hash\_sig9**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.194 Регистр ctx1\_hash\_sig10

Смещение:0x4B8

Формат регистра ctx1\_hash\_sig10 приведен в следующей таблице:

**Таблица 28.196. Поля регистра ctx1\_hash\_sig10**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.195 Регистр ctx1\_hash\_sig11

Смещение:0x4BC

Формат регистра ctx1\_hash\_sig11 приведен в следующей таблице:

**Таблица 28.197. Поля регистра ctx1\_hash\_sig11**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.196 Регистр ctx1\_hash\_sig12

Смещение:0x4C0

Формат регистра ctx1\_hash\_sig12 приведен в следующей таблице:

**Таблица 28.198. Поля регистра ctx1\_hash\_sig12**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.197 Регистр ctx1\_hash\_sig13

Смещение:0x4C4

Формат регистра ctx1\_hash\_sig13 приведен в следующей таблице:

**Таблица 28.199. Поля регистра ctx1\_hash\_sig13**

Разряды	Название	Описание	Сброс
31:0	sig		0

### 28.3.198 Регистр ctx1\_hash\_sig14

Смещение:0x4C8

Формат регистра ctx1\_hash\_sig14 приведен в следующей таблице:

**Таблица 28.200. Поля регистра ctx1\_hash\_sig14**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.199 Регистр ctx1\_hash\_sig15**

Смещение:0x4CC

Формат регистра ctx1\_hash\_sig15 приведен в следующей таблице:

**Таблица 28.201. Поля регистра ctx1\_hash\_sig15**

Разряды	Название	Описание	Сброс
31:0	sig		0

**28.3.200 Регистр wcddata**

Смещение:0x500

Если session\_ctr0.enable\_crypt\_hash\_dma==0, то через этот регистр происходит запись данных в cryptCore.

Для возможности пакетной записи к этому регистру можно обратиться по следующим адресам:

0x500, 0x504, 0x508, 0x50C,  
 0x510, 0x514, 0x518, 0x51C  
 0x520, 0x524, 0x528, 0x52C  
 0x530, 0x534, 0x538, 0x53C

Формат регистра wcddata приведен в следующей таблице:

**Таблица 28.202. Поля регистра wcddata**

Разряды	Название	Описание	Сброс
31:0	wcddata		0

**28.3.201 Регистр rcddata**

Смещение:0x540

Если cryptCore\_session\_ctr0.enable\_dma\_read==0, то через этот регистр осуществляется чтение данных из cryptCore.

Для возможности пакетного чтения, к этому регистру можно обратиться по следующим адресам:

0x540, 0x544, 0x548, 0x54C,



0x550, 0x554, 0x558, 0x55C  
 0x560, 0x564, 0x568, 0x56C  
 0x570, 0x574, 0x578, 0x57C

Формат регистра rdata приведен в следующей таблице:

**Таблица 28.203. Поля регистра rdata**

Разряды	Название	Описание	Сброс
31:0	rdata		0

### 28.3.202 Регистр whdata

Смещение: 0x580

Если hashCore\_session\_ctr0.enable\_hash\_dma==0, то через этот регистр происходит запись данных в hashCore.

Для возможности пакетного чтения, к этому регистру можно обратиться по следующим адресам:

0x580, 0x584, 0x588, 0x58C,  
 0x590, 0x594, 0x598, 0x59C  
 0x5A0, 0x5A4, 0x5A8, 0x5AC  
 0x5B0, 0x5B4, 0x5B8, 0x5BC

Формат регистра whdata приведен в следующей таблице:

**Таблица 28.204. Поля регистра whdata**

Разряды	Название	Описание	Сброс
31:0	whdata		0

### 28.3.203 Регистр soft\_reset

Смещение: 0x600

Программный сброс gms.

Формат регистра soft\_reset приведен в следующей таблице:

**Таблица 28.205. Поля регистра soft\_reset**

Разряды	Название	Описание	Сброс
31:1	-	Резерв	0
0	reset	Запись 1 приведет к сбросу gms_crypto.	1

## 28.4 Функционирование

### 28.4.1 Инициализация/завершение сессии шифрования cryptoCore.

Старт сессии шифрования происходит в следующих случаях:

- При записи 1 в поле start регистра cryptCore\_session\_ctr0.
- При записи 1 в поле resume\_crypt регистра session\_ctr1.

Все доступные для записи поля регистра session\_ctr1 – w1-типа. Таким образом, обращение к отдельным полям регистра освобождает от необходимости выполнения последовательности чтение-модификация-запись, необходимой при работе с регистром cryptCore\_session\_ctr0.

Завершение сессии шифрования происходит в следующих случаях:

- Достигает нуля счетчик crypt\_block\_cnt, принимающий при старте сессии значение регистров {crypt\_block\_size\_high, crypt\_block\_size\_low} и декрементирующийся после обработки каждого блока данных.
- Происходит запись 1 в поле stop\_crypt регистра session\_ctr1.

При завершении сессии обновляется актуальный контекст. Таким образом, его можно выгрузить и сохранить для последующего возобновление сессии.

Любое запись/чтение данных должно производиться блоками данных, размер которых кратен размеру базового блока шифрования: 128 бит - при работе с шифром “Кузнечик”, 64 бита – при работе с шифром “Магма”.

### 28.4.2 Загрузка/выгрузка контекста шифрования cryptoCore.

Выбор активного контекста текущей сессии шифрования производится через поле ctx регистра cryptCore\_session\_ctr0.

Для корректного переключения между контекстами необходимо соблюдение следующих условий:

- Смена контекста может проводиться лишь при неактивной сессии.
- До переключения между контекстами, в случае принудительной остановки сессии через поле stop\_crypt регистра session\_ctr1, необходимо предварительно убедиться в отсутствии данных в wc-fifo и rc-fifo.

### 28.4.3 Инициализация/завершение сессии HashCore

Старт сессии вчисления хэш-суммы происходит в следующих случаях:

- При записи 1 в поле start регистра hashCore\_session\_ctr0.
- При записи 1 в поле resume\_hash регистра session\_ctr1.

Все доступные для записи поля регистра session\_ctr1 – w1-типа. Таким образом, обращение к отдельным полям регистра освобождает от необходимости выполнения последовательности чтение-модификация-запись, необходимой при работе с регистром hashCore\_session\_ctr0.

Если при старте сессии, значение регистров активного контекста {ctxn\_ctxn\_hash\_N\_high, ctxn\_ctxn\_hash\_N\_low}  $\neq 0$ , то произойдет загрузка пререквизитов активного контекста (параметры hash, sig, N). В противном случае, сессия будет считаться новой, и значения указанных регистров будут игнорироваться. В этом случае, выбор режима вычисления хеш (256/512 бит) будет происходить через поле hashCore\_session\_ctr0/hash\_size.

Завершение сессии происходит в следующих случаях:

- Достигает нуля счетчик hash\_block\_cnt, принимающий при старте сессии значение регистров {hash\_block\_size\_high, hash\_block\_size\_low} и декрементирующийся после обработки каждого блока данных.
- Происходит запись 1 в поле stop\_hash регистра session\_ctr1.

При завершении сессии, обновляется актуальный контекст. Таким образом, его можно выгрузить и сохранить для последующего возобновления сессии.

Запись данных должна производиться блоками данных, размер которых кратен размеру блока вычисления хэш-суммы 64 байта. При этом важно, что:

- Паддинг данных при финализации должен быть добавлен программно.
- Если размер области хэшируемых данных, записанных в счетчик hash\_block\_cnt, не кратен 64 байтам, то необходимо программно дополнить нулями последний блок до 64 байт. При этом финализация будет выполнена с учетом объема данных, указанных в регистрах hash\_block\_size\_\*.
- Если размер области хэшируемых данных, записанных в счетчик hash\_block\_cnt, кратен 64 байтам, то для паддинга необходимо дополнить данные блоком в 64 байта. В этом случае, необходимо перед стартом сессии увеличить область hash\_block\_size на 64.

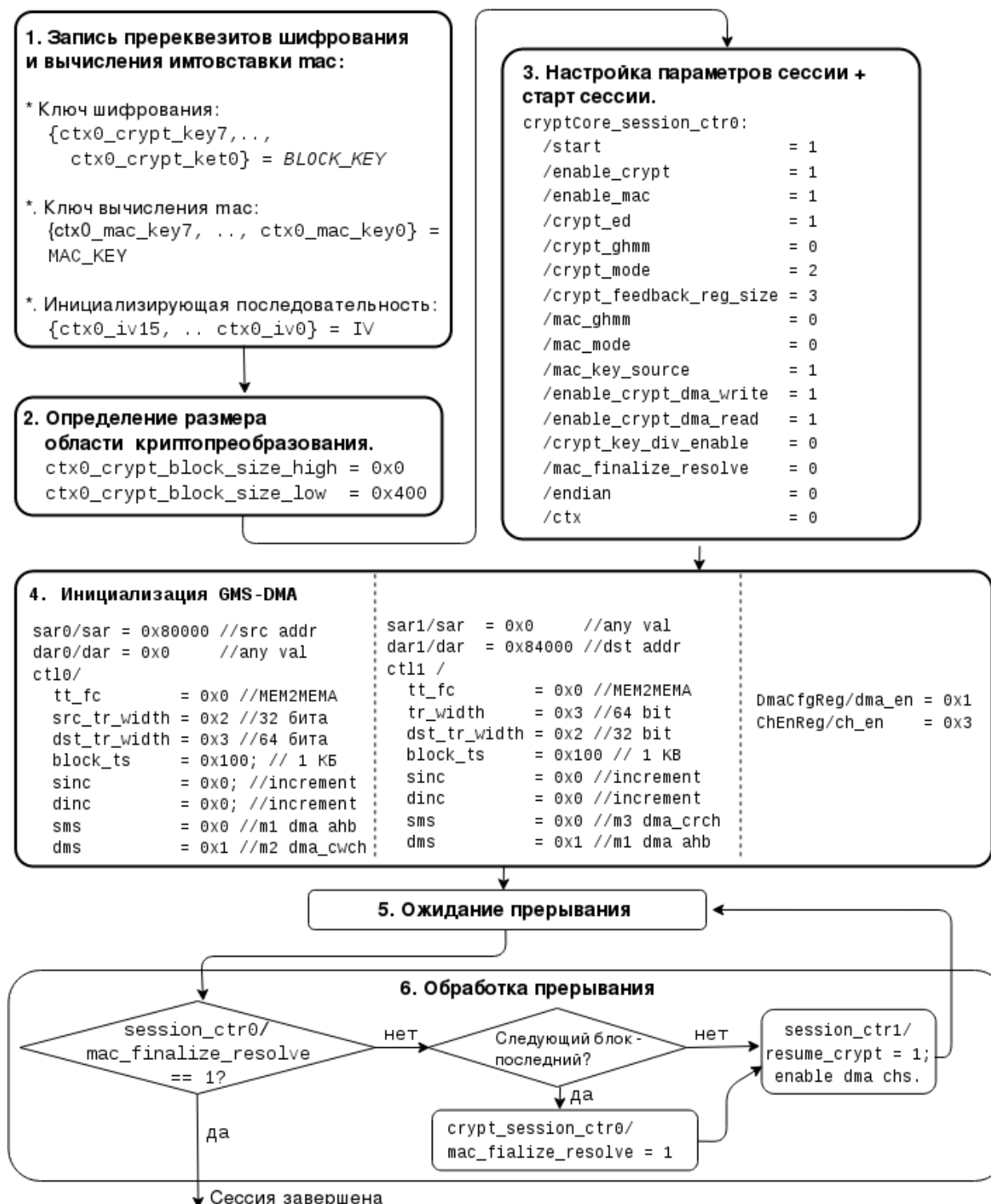
#### 28.4.4 Особенности сессии CryptCore в режиме ОМАС-АРКМ

Для корректной работы сессии CryptCore в режиме ОМАС-АСРКМ необходимо выполнение следующих условий:

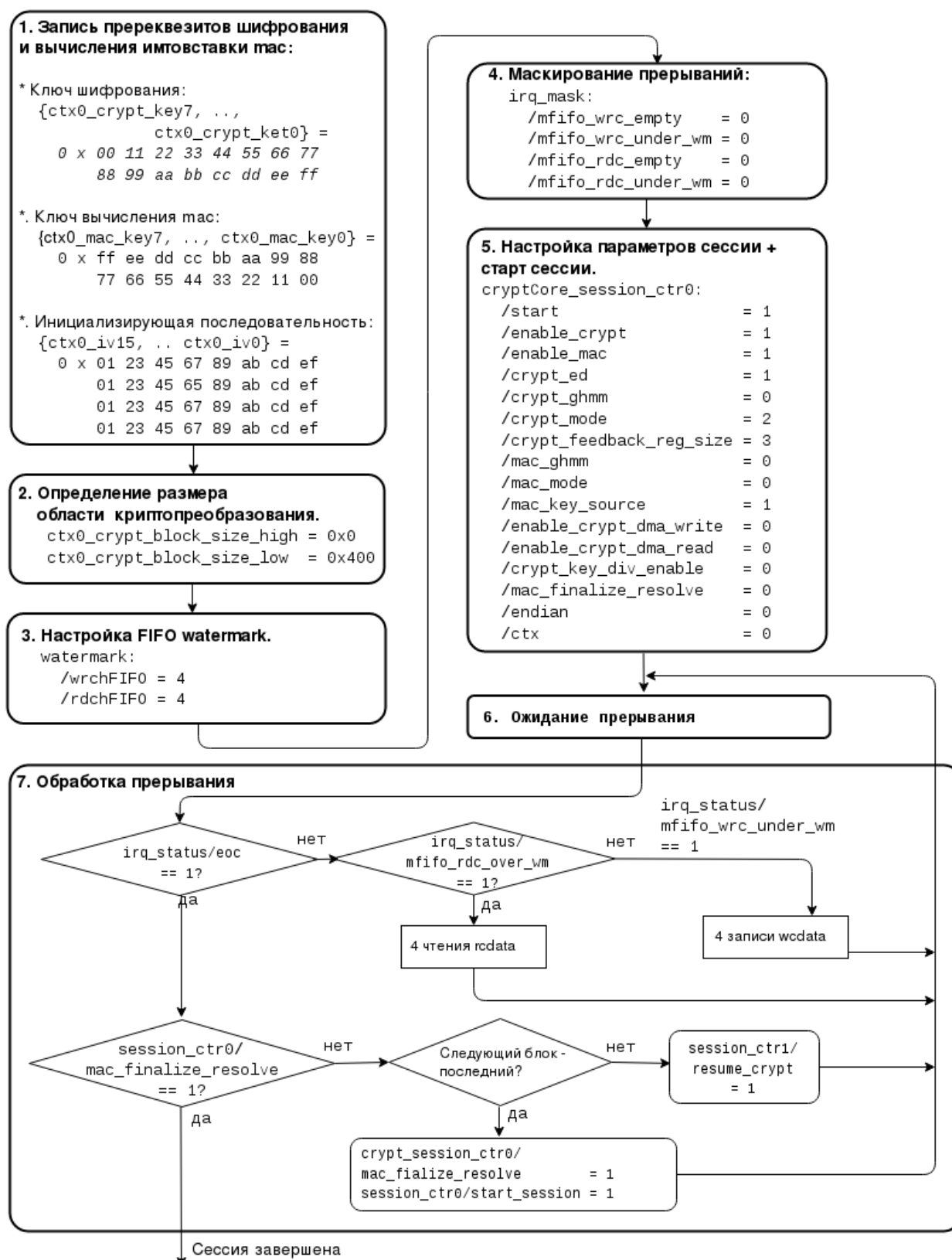
1. Вектор инициализирующей последовательности IV должен быть проинициализирован значением вида  $\{n/2'b\{1\}, n/2'b\{0\}\}$ , где  $n = 128$  для “Кузнечика”, и  $n = 64$  для “Магмы”.
2. Мастер ключ задается через регистры `crypt_key`.
3. Выбор алгоритма шифрования происходит через поле `cryptoCore_session_ctr0/mac_ghmm`.
4. Несмотря на то, что при работе `cryptoCore` в режиме ОМАС-АСРКМ задействованы оба вычислителя (МАС и блочное шифрование), необходимо выставить в 1 бит `enable_mac`, а бит `enable_crypt` надо оставить в 0.
5. Частота смены ключей (параметр  $T^*$ ) задается через регистр `ctracpkm_section_size`. Размер секции (параметр  $N$ ) задается через регистр `omac_section_size`. Необходимо учитывать, что оба параметра задаются в байтах и должны быть кратны размеру блока выбранного алгоритма блочного шифрования.
6. В процессе работы вычислителя, регистры инициализирующей последовательности IV[255:128] будут принимать в себя вспомогательные секционные ключи K1.

## 28.5 Примеры программного управления

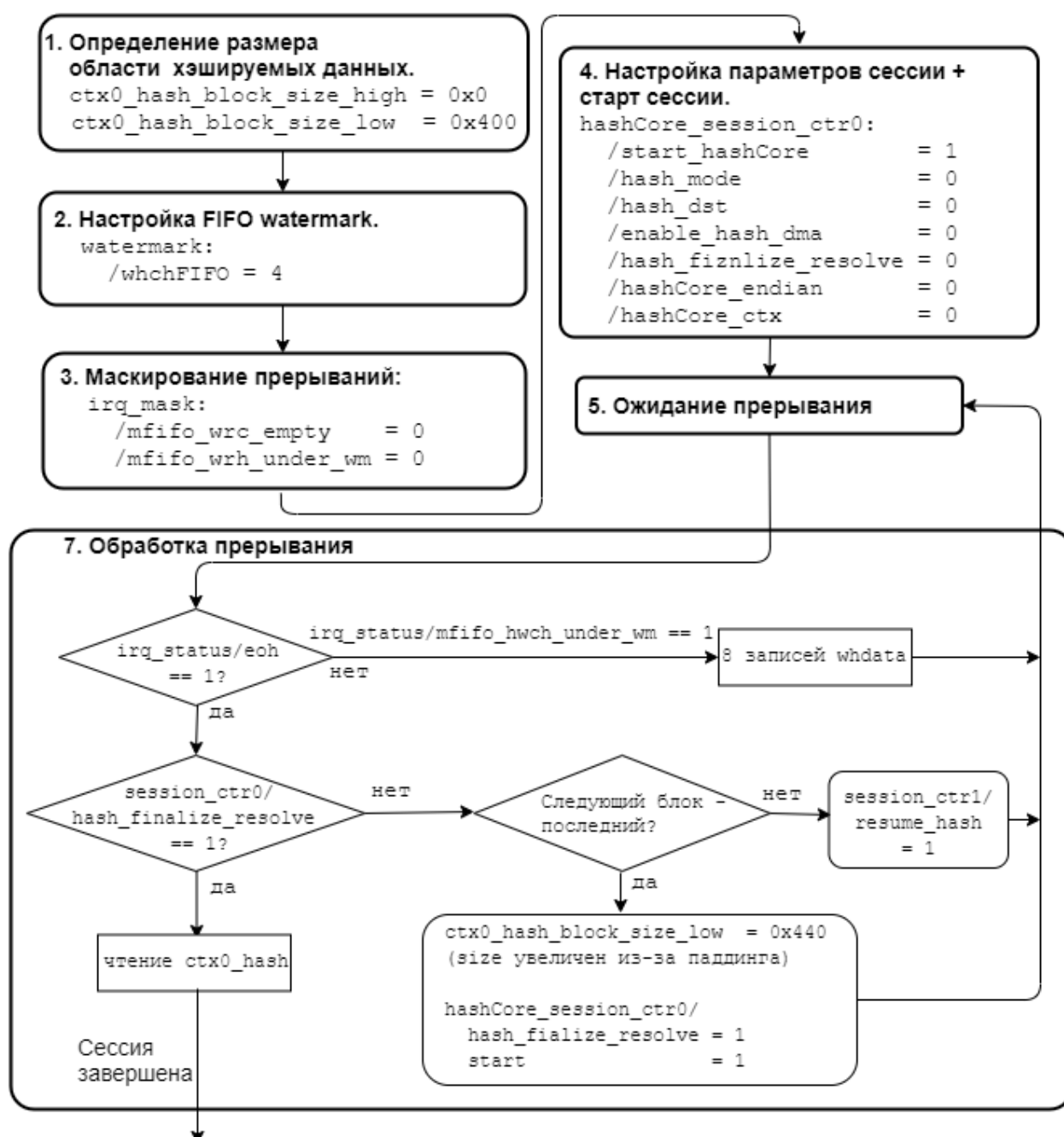
### 28.5.1 Пример сессии шифрования с выработкой имитовставки MAC без включения GMS-DMA.



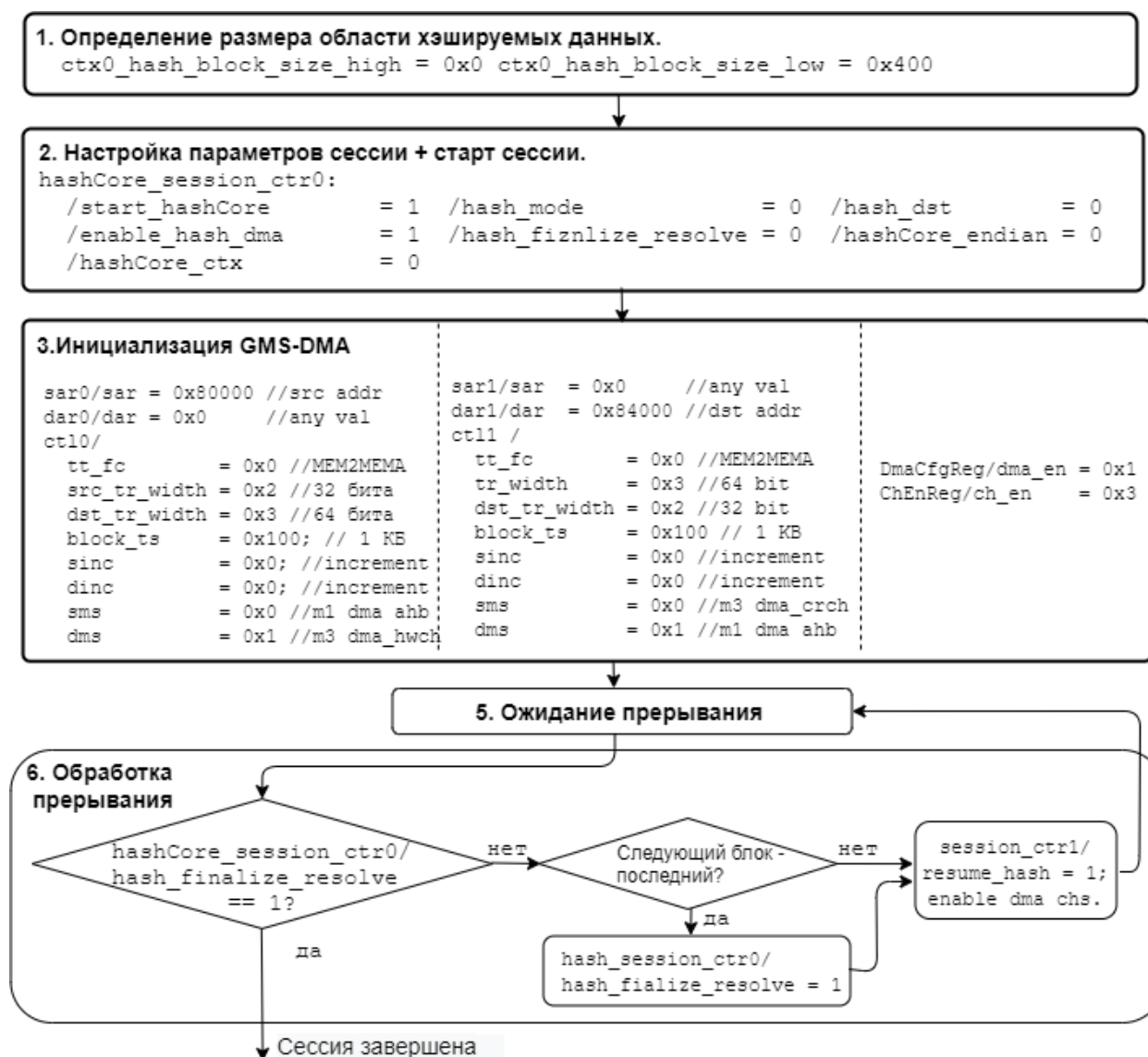
## 28.5.2 Пример сессии шифрования с выработкой имитовставки MAC с использованием GMS-DMA.



### 28.5.3 Пример сессии вычисления HASH без использования GMS-PDMA



## 28.5.4 Пример сессии вычисления HASH с использованием GMS-PDMA





## 29. МНУ: БЛОК ОБМЕНА СООБЩЕНИЯМИ

### 29.1 Введение

Два блока Message Handling Unit (МНУ) позволяют программно устанавливать прерывания для процессорных ядер.

С помощью контроллера PPC каждый блок МНУ может быть настроен как Secure, либо как Non-secure устройство:

- МНУ0: Non-secure адрес 0x4000\_3000, Secure адрес 0x5000\_3000.
- МНУ1 Non-secure адрес 0x4000\_4000, Secure адрес 0x5000\_4000.

## 29.2 Регистры

### 29.2.1 Перечень регистров MNU

Перечень регистров MNU приведен в следующей таблице.

**Таблица 29.1. Регистры MNU**

Условное обозначение	Описание	Исходное состояние	Смещение
MNU_CPU0INTR_STAT	Регистр статуса прерываний ядра CPU0	0x0	0x00
MNU_CPU0INTR_SET	Регистр установки прерываний ядра CPU0	0x0	0x04
MNU_CPU0INTR_CLR	Регистр сброса прерываний ядра CPU0	0x0	0x08
MNU_CPU1INTR_STAT	Регистр статуса прерываний ядра CPU1	0x0	0x10
MNU_CPU1INTR_SET	Регистр установки прерываний ядра CPU1	0x0	0x14
MNU_CPU1INTR_CLR	Регистр сброса прерываний ядра CPU1	0x0	0x18

Запись в регистры поддерживается только с размером 32 бита. Запись с размером 8 или 16 бит игнорируется.

MNU содержит два набора регистров: CPU0INTR\_STAT, CPU0INTR\_SET, CPU0INTR\_CLR для ядра CPU0 и CPU1INTR\_STAT, CPU1INTR\_SET, CPU1INTR\_CLR для ядра CPU1. Далее приводится обобщенное для обоих ядер описание этих регистров.

Регистры CPU<sub>n</sub>INTR\_STAT, CPU<sub>n</sub>INTR\_SET, CPU<sub>n</sub>INTR\_CLR позволяют устанавливать, сбрасывать и проверять статус прерывания MNU для ядра CPU<sub>n</sub>. Каждый регистр содержит 4 бита, которые могут быть независимо установлены и сброшены. Каждый бит может использоваться как отдельное сообщение для процессорного ядра.

### 29.2.2 Регистр MNU\_CPU<sub>n</sub>INTR\_STAT

Регистр статуса прерываний ядра CPU<sub>n</sub>.

Формат регистра MNU\_CPU<sub>n</sub>INTR\_STAT приведен в следующей таблице.

**Таблица 29.2. Поля регистра MNU\_CPU<sub>n</sub>INTR\_STAT**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:4	-	RO	0	Резерв
3:0	CPU <sub>n</sub> INTR_STAT	RO	0	Каждый бит показывает статус прерывания MNU для ядра CPU <sub>n</sub> : 0x0: прерывание не установлено; 0x1: прерывание установлено.

### 29.2.3 Регистр MNU\_CPU<sub>n</sub>INTR\_SET

Регистр установки прерываний ядра CPU<sub>n</sub>.

Формат регистра MHU\_CPUUnINTR\_SET приведен в следующей таблице.

**Таблица 29.3. Поля регистра MHU\_CPUUnINTR\_SET**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:4	-	RO	0	Резерв
3:0	CPUUnINTR_SET	WO	0	При записи 1 в бит CPUUnINTR_SET[k] устанавливается сигнал прерывания MHU для ядра CPUUn, а также устанавливается соответствующий бит статуса CPUUnINTR_STAT[k].

## 29.2.4 Регистр MHU\_CPUUnINTR\_CLR

Регистр сброса прерываний ядра CPUUn.

Формат регистра MHU\_CPUUnINTR\_CLR приведен в следующей таблице.

**Таблица 29.4. Поля регистра MHU\_CPUUnINTR\_CLR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:4	-	RO	0	Резерв
3:0	CPUUnINTR_CLR	WO	0	При записи 1 в бит CPUUnINTR_CLR[k] сбрасывается сигнал прерывания MHU для ядра CPUUn, а также сбрасывается соответствующий бит статуса CPUUnINTR_STAT[k].

## 30. RPU: КОНТРОЛЛЕР УПРАВЛЕНИЯ ПИТАНИЕМ

### 30.1 Введение

Для управления питанием отключаемых доменов в микросхеме используются контроллеры RPU (Power Policy Unit).

Каждый отключаемый домен питания ядра управляется с помощью собственного контроллера RPU и может находиться в одном из следующих состояний:

- ON – питание включено;
- OFF – питание логики и блоков памяти выключено, состояние схемы теряется;
- MEM\_RET – режим пониженного энергопотребления памяти (retention) с сохранением данных (только для доменов PD\_SRAMn).

В общем случае, RPU может управлять состоянием домена двумя способами:

- Статическое управление – домен может быть перманентно включен, выключен или переведен в режим сохранения данных (только для PD\_SRAMn).
- Динамическое управление – целевое состояние питания домена задается программно. Переход из состояния ON в заданное состояние и обратно происходит автоматически в зависимости от состояния входа DEVACTIVE.

Возможны следующие динамические переходы:

- ON → OFF
- OFF → ON
- ON → MEM\_RET
- MEM\_RET → ON

RPU разных доменов имеют разные настройки допустимых переходов. Так, например, домены PD\_CRYPT0, PD\_GMS, PD\_GNSS поддерживают только статическое управление. Параметры всех RPU приведены в Таблица 30.1.

**Таблица 30.1. Параметры RPU**

Параметр	PD_SYS	PD_CPU <sub>n</sub> , n = 0, 1	PD_SRAM <sub>n</sub> , n = 0 - 3	PD_DEBUG	PD_CRYPT0	PD_GMS, PD_GNSS
Тип интерфейса	P-channel	P-channel	Q-channel	Q-channel	Q-channel	Q-channel
Состояние после «холодного» сброса	ON	ON для CPU0; OFF для CPU1	ON	ON	ON	OFF
Динамический переход в OFF	+	+	+	+	-	-
Динамический переход в ON	+	+	+	+	-	-
Динамический переход в MEM_RET	-	-	+	-	-	-

Статический переход в OFF	-1	-1	-1	-1	+	+
Статический переход в ON	-1	+	+	+	+	+
Статический переход в MEM_RET	-1	-1	-1	-	-	-
Примечания: 1 – попытка установки данного статического состояния приводит к возникновению исключения BusFault.						

Подробнее о работе PPU см. в Arm® Power Policy Unit Version 1.1 Architecture Specification.

## 30.2 Регистры

### 30.2.1 Перечень регистров PPU

Перечень регистров PPU приведен в следующей таблице.

**Таблица 30.2 Регистры PPU**

Условное обозначение	Описание	Исходное состояние	Смещение
PPU_PWPR	Power Policy Register	0x00000000	0x000
PPU_PWSR	Power Status Register	0x00000000	0x008
PPU_DISR	Device Interface Input Current Status Register	0x00000000	0x010
PPU_MISR	Miscellaneous Input Current Status Register	0x00000000	0x014
PPU_STSR	Stored Status Register	0x00000000	0x018
PPU_PWCR	Power Configuration Register	0x00000000	0x020
PPU_IMR	Interrupt Mask Register	0x00000002	0x030
PPU_AIMR	Additional Interrupt Mask Register	0x00000006	0x034
PPU_ISR	Interrupt Status Register	0x00000000	0x038
PPU_AISR	Additional Interrupt Status Register	0x00000000	0x03C
PPU_IESR	Input Edge Sensitivity Register	0x00000000	0x040
PPU_DCDR0	Device Control Delay Configuration Register 0	0x00000000	0x170
PPU_DCDR1	Device Control Delay Configuration Register 1	0x00000000	0x174
PPU_IDR0	PPU Identification Register 0	0x00030100	0xFB0
PPU_IDR1	PPU Identification Register 1	0x00000070	0xFB4
PPU_IIDR	Implementation Identification Register	0x00000000	0xFC8
PPU_AIDR	Architecture Identification Register	0x00000011	0xFCC

### 30.2.2 Регистр PPU\_PWPR

This register enables software to program power mode policy. It also contains settings including the enable for dynamic transitions. This register does not reflect the current power mode value. The current power mode of the domain is reflected in the Power Status Register (PPU\_PWSR).

Формат регистра PPU\_PWPR приведен в следующей таблице.

**Таблица 30.3 Поля регистра PPU\_PWPR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:9	-	RO	0x0	Резерв
8	PWR_DYN_EN	RW	0x0	Power mode dynamic transition enable. When this bit is set to 0b1 dynamic transitions are enabled for power modes, allowing transitions to be initiated by changes on power mode DEVACTIVE inputs.
7:4	-	RO	0x0	Резерв
3:0	PWR_POLICY	RW	0x0	Power mode policy. When static power mode transitions are enabled, PWR_DYN_EN is set to 0b0, this is the target power mode for the PPU. When dynamic power mode transitions are enabled, PWR_DYN_EN is set to 0b1, this is the minimum power mode for the PPU.

### 30.2.3 Регистр PPU\_PWSR

This read-only register contains status information for the power mode, dynamic transitions.

Формат регистра PPU\_PWSR приведен в следующей таблице.

**Таблица 30.4 Поля регистра PPU\_PWSR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:9	-	RO	0x0	Резерв
8	PWR_DYN_STATUS	RO	0x0	Power mode dynamic transition status. When set to 0b1 power mode dynamic transitions are enabled. There might be a delay in dynamic transitions becoming active or inactive if the PPU is transitioning when PWR_DYN_EN is programmed.
7:4	-	RO	0x0	Резерв
3:0	PWR_STATUS	RO	0x0	Power mode status. These bits reflect the current power mode of the PPU.

### 30.2.4 Регистр PPU\_DISR

This read-only register contains status reflecting the values of the device interface inputs.

Формат регистра PPU\_DISR приведен в следующей таблице.

**Таблица 30.5 Поля регистра PPU\_DISR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:11	-	RO	0x0	Резерв
10:0	PWR_DEVPACTIVE_STATUS	RO	0x0	Status of the power mode DEVPACTIVE inputs. For Q-Channel: There is one bit for each device interface Q-Channel DEVQACTIVE. For example, bit 0 is for the Q-Channel 0 DEVQACTIVE, and bit 1 for the Q-Channel 1 DEVQACTIVE. For a Q-Channel PPU bits [10:8] are always reserved as only 8 device interface Q-Channels are supported. For P-Channel: There is one bit for each bit of the power mode DEVPACTIVE inputs. For example, bit 10 is for DEVPACTIVE[10], and bit 9 is for DEVPACTIVE[9]. Bit 0 relates to DEVPACTIVE[0] and is always 0.

### 30.2.5 Регистр PPU\_MISR

This read-only register contains status reflecting the values of miscellaneous inputs.

Формат регистра PPU\_MISR приведен в следующей таблице.

**Таблица 30.6 Поля регистра PPU\_MISR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:24	-	RO	0x0	Резерв
23:16	DEV_DENY_STATUS	RO	0x0	Status of the device interface DEV_DENY inputs. For Q-Channel: There is one bit for each device interface DEVQDENY. For example, bit 16 is for Q-Channel 0 DEVQDENY, and bit 17 for Q-Channel 1 DEVQDENY. For P-Channel: Bit 16 is for the device interface DEVPDENY. Other bits are reserved.
15:8	DEV_ACCEPT_STATUS	RO	0x0	Status of the device interface DEV_ACCEPT inputs. For Q-Channel: There is one bit for each device interface DEVQACCEPTn. For example, bit 8 is for Q-Channel 0 DEVQACCEPTn and bit 9 for Q-Channel 1 DEVQACCEPTn. For P-Channel: Bit 8 is for the device interface DEVPACCEPT. Other bits are reserved.
7:1	-	RO	0x0	Резерв
0	PCSMP_ACCEPT_STATUS	RO	0x0	The status of the PCSMP_ACCEPT input.

### 30.2.6 Регистр PPU\_STSR

This read-only register contains status information about the DEV\_DENY inputs that were set HIGH during the last transition. This register updates when a transition completes. Therefore, it contains the status of the last transition. This allows software to determine the components that have denied a device interface request after the transition.

Формат регистра PPU\_STSR приведен в следующей таблице.

**Таблица 30.7 Поля регистра PPU\_STSR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:8	-	RO	0x0	Резерв
7:0	STORED_DEV_DENY	RO	0x0	Status of the DEV_DENY signals from the last device interface Q-Channel transition. For Q-Channel: There is one bit for each device interface DEVQDENY. For example, bit 0 is for Q-Channel 0 DEVQDENY, and bit 1 for Q-Channel 1 DEVQDENY. For a Q-Channel PPU with a single Q-Channel this field is reserved. For P-Channel: This field is reserved

### 30.2.7 Регистр PPU\_PWCR

This register controls enabling and disabling of hardware control inputs to the PPU. The PWR\_DEVACTIVEEN and OP\_DEVACTIVEEN fields in this register control the ability of the DEVACTIVE inputs to initiate power mode transitions, but not the ability to generate input edge interrupt events. Bits for unused channels are the MSB's and are reserved.

Формат регистра PPU\_PWCR приведен в следующей таблице.



Таблица 30.8 Поля регистра PPU\_PWCR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:19	-	RO	0x0	Резерв
18:8	PWR_DEVACTIVE EN	RW	0x0	These bits enable the power mode DEVACTIVE inputs. When a bit is to 0b1 the related DEVACTIVE input is enabled, when set to 0b0 it is disabled. All available bits are reset to 0b1. For Q-Channel: There is one bit for each device interface Q-Channel DEVQACTIVE. For example, bit 8 is for the Q-Channel 0 DEVQACTIVE, and bit 9 for the Q-Channel 1 DEVQACTIVE. For a Q-Channel PPU bits [18:16] are always reserved as only 8 device interface Q-Channels are supported. For P-Channel: There is one bit for each bit of the DEVPACTIVE input. For example, bit 18 is for DEVPACTIVE[10], and bit 17 is for DEVPACTIVE[9]. Bit 8 relates to DEVPACTIVE[0] and is always 0b0.
7:0	DEVREQEN	RW	0x0	When set to 0b1 enables the device interface handshake for transitions. All available bits are reset to 0b1. For Q-Channel: There is one bit for each device interface channel. For example, bit 0 is for Q-Channel 0, and bit 1 is for Q-Channel 1. For P-Channel: Bit 0 is for the single P-Channel. Other bits are reserved

### 30.2.8 Регистр PPU\_IMR

This register controls the events that assert the interrupt output. Additional event masking controls are in the Additional Interrupt Mask Register (PPU\_AIMR), Input Edge Sensitivity Register (PPU\_IESR), and the Operating Mode Active Edge Sensitivity Register (PPU\_OPSR). When an interrupt event is masked an occurrence of the event does not set the corresponding bit in the interrupt status register. For each bit: 0b0 – Event enabled. 0b1 – Event masked.

Формат регистра PPU\_IMR приведен в следующей таблице.

Таблица 30.9 Поля регистра PPU\_IMR

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0x0	Резерв
2	STA_DENY_IRQ_MASK	RW	0x0	Static transition denial event mask.
1	STA_ACCEPT_IRQ_MASK	RW	0x1	Static transition acceptance event mask.
0	STA_POLICY_TRN_IRQ_MASK	RW	0x0	Static full policy transition completion event mask.

### 30.2.9 Регистр PPU\_AIMR

This register controls the events that assert the interrupt output. Additional event masking controls are in the Interrupt Mask Register (PPU\_IMR), Input Edge Sensitivity Register (PPU\_IESR), and the Operating Mode Active Edge Sensitivity Register (PPU\_OPSR). When an interrupt event is

masked an occurrence of the event does not set the corresponding bit in the interrupt status register. For each bit: 0b0 – Event enabled. 0b1 – Event masked.

Формат регистра PPU\_AIMR приведен в следующей таблице.

**Таблица 30.10 Поля регистра PPU\_AIMR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0x0	Резерв
2	DYN_DENY_IRQ_MASK	RW	0x1	Dynamic transition denial event mask. When no dynamic transitions are supported this field is reserved.
1	DYN_ACCEPT_IRQ_MASK	RW	0x1	Dynamic transition acceptance event mask. When no dynamic transitions are supported this field is reserved.
0	UNSPT_POLICY_IRQ_MASK	RW	0x0	Unsupported Policy event mask.

### 30.2.10 Регистр PPU\_ISR

This register contains information about events causing the assertion of the interrupt output. It is also used to clear interrupt events. A bit set to 0b1 indicates the event asserted the interrupt output. Multiple events can be active at the same time. When an interrupt event is masked an occurrence of that event does not set the status bit. A write of 0b1 to an event bit clears that event. A write of 0b0 to a bit has no effect. The interrupt output stays HIGH until all status bits in the Interrupt Status Register (PPU\_ISR) and the Additional Interrupt Status Register (PPU\_AISR) are 0b0. When the OTHER\_IRQ bit is set, this indicates an event from the Additional Interrupt Status Register (PPU\_AISR) has caused the interrupt output to be asserted. This bit cannot be cleared by writing to this register. It must be cleared by writing to the active event in the Additional Interrupt Status Register (PPU\_AISR). Bits for unused channels are the MSBs and are reserved.

Формат регистра PPU\_ISR приведен в следующей таблице.

**Таблица 30.11 Поля регистра PPU\_ISR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:19	-	RO	0x0	Резерв
18:8	PWR_ACTIVE_EDGE_IRQ	RW	0x0	Indicates which power mode DEVPACTIVE inputs caused the input edge event. For Q-Channel: There is one bit for each device Q-Channel DEVQACTIVE. For example, bit 8 is for the Q-Channel 0 DEVQACTIVE, and bit 9 for the Q-Channel 1 DEVQACTIVE. For a Q-Channel PPU bits [18:16] are always reserved as only 8 device interface Q-Channels are supported. For P-Channel: There is one bit for each bit of the power mode DEVPACTIVE group. For example, bit 18 is for DEVPACTIVE[10], and bit 17 is for DEVPACTIVE[9]. Bit 8 relates to DEVPACTIVE[0] and is always 0.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
7	OTHER_IRQ	RO	0x0	Indicates there is an interrupt event pending in the Additional Interrupt Status Register (PPU_AISR)
6:3	-	RO	0x0	Резерв
2	STA_DENY_IRQ	RW	0x0	Static transition denial event status.
1	STA_ACCEPT_IRQ	RW	0x0	Static transition acceptance event status.
0	STA_POLICY_TRN_IRQ	RW	0x0	Static full policy transition completion event status.

### 30.2.11 Регистр PPU\_AISR

This register contains information about events causing the assertion of the interrupt output. It is also used to clear interrupt events. A bit set to 0b1 indicates the event asserted the interrupt output. Multiple events can be active at the same time. When an interrupt event is masked an occurrence of that event does not set the status bit. A write of 0b1 to an event bit clears that event. A write of 0b0 has no effect. The interrupt output stays HIGH until all status bits in the Interrupt Status Register (PPU\_ISR) and the Additional Interrupt Status Register (PPU\_AISR) are set to 0b0. When an interrupt status is set to 0b1 in this register it sets the OTHER\_IRQ bit in the Interrupt Status Register (PPU\_ISR). Status bits in this register are only cleared by writing to this register.

Формат регистра PPU\_AISR приведен в следующей таблице.

**Таблица 30.12 Поля регистра PPU\_AISR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:3	-	RO	0x0	Резерв
2	DYN_DENY_IRQ	RW	0x0	Dynamic transition denial event status. When no dynamic transitions are supported this field is reserved.
1	DYN_ACCEPT_IRQ	RW	0x0	Dynamic transition acceptance event status. When no dynamic transitions are supported this field is reserved.
0	UNSPT_POLICY_IRQ	RW	0x0	Unsupported Policy event status.

### 30.2.12 Регистр PPU\_IESR

This register configures the transitions on the power mode DEVACTIVE inputs that generate an Input Edge interrupt event. For each input, there are two registers bits with the following encoding:

- 0b00 – Event masked.
- 0b01 – Event on rising edge.
- 0b10 – Event on falling edge.
- 0b11 – Event on both edges.

When an event is masked an occurrence of the event does not set the corresponding bit in the interrupt status register. Fields for power mode DEVACTIVE inputs that are not supported are reserved.

Формат регистра PPU\_IESR приведен в следующей таблице.

**Таблица 30.13 Поля регистра PPU\_IESR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:22	-	RO	0x0	Резерв
21:20	DEVACTIVE10_EDGE	RW	0x0	or Q-Channel: Reserved. For P-Channel: DEVACTIVE 10 edge sensitivity.
19:18	DEVACTIVE09_EDGE	RW	0x0	For Q-Channel: Reserved. For P-Channel: DEVACTIVE 9 edge sensitivity.
17:16	DEVACTIVE08_EDGE	RW	0x0	For Q-Channel: Reserved. For P-Channel: DEVACTIVE 8 edge sensitivity.
15:14	DEVACTIVE07_EDGE	RW	0x0	DEVACTIVE 7 edge sensitivity.
13:12	DEVACTIVE06_EDGE	RW	0x0	DEVACTIVE 6 edge sensitivity.
11:10	DEVACTIVE05_EDGE	RW	0x0	DEVACTIVE 5 edge sensitivity.
9:8	DEVACTIVE04_EDGE	RW	0x0	DEVACTIVE 4 edge sensitivity.
7:6	DEVACTIVE03_EDGE	RW	0x0	DEVACTIVE 3 edge sensitivity.
5:4	DEVACTIVE02_EDGE	RW	0x0	DEVACTIVE 2 edge sensitivity.
3:2	DEVACTIVE01_EDGE	RW	0x0	DEVACTIVE 1 edge sensitivity.
1:0	DEVACTIVE00_EDGE	RW	0x0	For Q-Channel: DEVACTIVE 0 edge sensitivity. For P-Channel: Reserved.

### 30.2.13 Регистр PPU\_DCDR0

This register is used to program device control delay parameters.

Формат регистра PPU\_DCDR0 приведен в следующей таблице.

**Таблица 30.14 Поля регистра PPU\_DCDR0**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:24	-	RO	0x0	Резерв
23:16	RST_HWSTAT_DELAY	RW	0x0	Delay from reset de-assertion to HWSTAT update.
15:8	ISO_CLKEN_DLY	RW	0x0	Delay from isolation enable de-assertion to clock enable assertion.
7:0	CLKEN_RST_DLY	RW	0x0	Delay from clock enable assertion to reset de-assertion.

### 30.2.14 Регистр PPU\_DCDR1

This register is used to program device control delay parameters.

Формат регистра PPU\_DCDR1 приведен в следующей таблице.

**Таблица 30.15 Поля регистра PPU\_DCDR1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:16	-	RO	0x0	Резерв
15:8	CLKEN_ISO_DLY	RW	0x0	Delay from clock enable de-assertion to isolation enable assertion.
7:0	ISO_RST_DLY	RW	0x0	Delay from isolation enable assertion to reset assertion.

### 30.2.15 Регистр PPU\_IDR0

This read-only register contains information on the type and number of channels on the device interface and power and operating modes supported. For each \*\_SPT bit: 0b0: Feature not supported. 0b1: Feature supported. The default values for the fields of this register are dependent on PPU configuration options.

Формат регистра PPU\_IDR0 приведен в следующей таблице.

**Таблица 30.16 Поля регистра PPU\_IDR0**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:30	-	RO	0x0	Резерв
29	DYN_WRM_RST_SPT	RO	0x0	Dynamic WARM_RST support. Always set to 0b0 for a Q- Channel PPU.
28	DYN_ON_SPT	RO	0x0	Dynamic ON support.
27	DYN_FUNC_RET_SPT	RO	0x0	Dynamic FUNC_RET support.
26	DYN_FULL_RET_SPT	RO	0x0	Dynamic FULL_RET support.
25	DYN_MEM_OFF_SPT	RO	0x0	Dynamic MEM_OFF support.
24	DYN_LGC_RET_SPT	RO	0x0	Dynamic LOGIC_RET support. Always set to 0b0 for a Q- Channel PPU.
23	DYN_MEM_RET_EMU_SPT	RO	0x0	Dynamic MEM_RET_EMU support
22	DYN_MEM_RET_SPT	RO	0x0	Dynamic MEM_RET support.
21	DYN_OFF_EMU_SPT	RO	0x0	Dynamic OFF_EMU support.
20	DYN_OFF_SPT	RO	0x0	Dynamic OFF support.
19	-	RO	0x0	Резерв
18	STA_DBG_RECOV_SPT	RO	0x0	DBG_RECOV support. Always set to 0b0 for a Q- Channel PPU.
17	STA_WRM_RST_SPT	RO	0x1	WARM_RST support.
16	STA_ON_SPT	RO	0x1	ON support.
15	STA_FUNC_RET_SPT	RO	0x0	FUNC_RET support.
14	STA_FULL_RET_SPT	RO	0x0	FULL_RET support.
13	STA_MEM_OFF_SPT	RO	0x0	MEM_OFF support.
12	STA_LGC_RET_SPT	RO	0x0	LOGIC_RET support. Always set to 0b0 for a Q- Channel PPU.

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
11	STA_MEM_RET_EMU_SPT	RO	0x0	MEM_RET_EMU support.
10	STA_MEM_RET_SPT	RO	0x0	MEM_RET support.
9	STA_OFF_EMU_SPT	RO	0x0	OFF_EMU support.
8	STA_OFF_SPT	RO	0x1	OFF support.
7:4	NUM_OPMODE	RO	0x0	No. of operating modes supported is NUM_OPMODE + 1.
3:0	DEVCHAN	RO	0x0	No. of Device Interface Channels • Q-Channel PPU: The number of Q-Channels (minimum of 1). • P-Channel PPU: Set to 0.

### 30.2.16 Регистр PPU\_IDR1

This read-only register contains information on the optional features and configurations that are supported by this PPU. For each \*\_SPT bit: 0b0: Feature not supported. 0b1: Feature supported. The default values for the fields of this register are dependent on PPU configuration options

Формат регистра PPU\_IDR1 приведен в следующей таблице.

**Таблица 30.17 Поля регистра PPU\_IDR1**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:13	-	RO	0x0	Резерв
12	OFF_MEM_RET_TRANS	RO	0x0	OFF to MEM_RET direct transition. Indicates if direct transitions from OFF to MEM_RET and from OFF_EMU to MEM_RET_EMU are supported.
11	-	RO	0x0	Резерв
10	OP_ACTIVE	RO	0x0	Operating mode use model for dynamic transitions. When operating modes are not supported this field is reserved.
9	STA_POLICY_OP_IRQ_SPT	RO	0x0	Operating policy transition completion event status.
8	STA_POLICY_PWR_IRQ_SPT	RO	0x0	Power policy transition completion event status.
7	-	RO	0x0	Резерв
6	FUNC_RET_RAM_REG	RO	0x1	Indicates if the PPU_FUNRR register is present or reserved. 0b0 – Reserved. 0b1 – Present.
5	FULL_RET_RAM_REG	RO	0x1	Indicates if the PPU_FULRR register is present or reserved. 0b0 – Reserved. 0b1 – Present.
4	MEM_RET_RAM_REG	RO	0x1	Indicates if the PPU_MEMRR register is present or reserved. 0b0 – Reserved. 0b1 – Present.
3	-	RO	0x0	Резерв
2	LOCK_SPT	RO	0x0	Lock and the lock interrupt event are supported.
1	SW_DEV_DEL_SPT	RO	0x0	Software device delay control configuration support.
0	PWR_MODE_ENTRY_DEL_SPT	RO	0x0	Power mode entry delay support.

### 30.2.17 Регистр PPU\_IIDR

This register provides information about the implementer and implementation of the PPU.

Формат регистра PPU\_IIDR приведен в следующей таблице.

**Таблица 30.18 Поля регистра PPU\_IIDR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:20	PRODUCT_ID	RO	0x0	IMPLEMENTATION DEFINED value identifying the PPU part.
19:16	VARIANT	RO	0x0	IMPLEMENTATION DEFINED value used to distinguish product variants, or major revisions of the product.
15:12	REVISION	RO	0x0	IMPLEMENTATION DEFINED value used to distinguish minor revisions of the product.
11:0	IMPLEMENTER	RO	0x0	Implementer identification. [11:8] The JEP106 continuation code of the implementer. [7] Always 0. [6:0] The JEP106 identity code of the implementer. For an Arm implementation, bits [11:0] are 0x43B.

### 30.2.18 Регистр PPU\_AIDR

This register identifies the PPU architecture revision. [7:0] – 0x11: Indicates PPU architecture revision 1.1.

Формат регистра PPU\_AIDR приведен в следующей таблице.

**Таблица 30.19 Поля регистра PPU\_AIDR**

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:8	-	RO	0x0	Резерв
7:4	ARCH_REV_MAJOR	RO	0x1	0x1 – PPU architecture major revision 1. Other Values – Reserved.
3:0	ARCH_REV_MINOR	RO	0x1	0x1 – PPU Architecture minor revision 1. Other Values – Reserved.

## 31. SWDT, NSWDT, LPWDT: СТОРОЖЕВЫЕ ТАЙМЕРЫ

В микросхеме реализованы три сторожевых таймера для различных сценариев применения:

- SWDT – доверенный сторожевой таймер (secure watchdog);
- NSWDT – сторожевой таймер общего назначения (non-secure watchdog);
- LPWDT – сторожевой таймер в домене PD\_AON, для работы в режимах энергосбережения.

NSWDT, SWDT, LPWDT могут служить источником сброса микросхемы, SWDT и LPWDT также генерируют NMI прерывания для ядер CPU. Подробнее см. в главе 2.

Программная модель и логика функционирования таймера описаны в разделе 4.5 англоязычной документации [1].

### 31.1 Отладка

Сторожевые таймеры подключены к выходным сигналам-триггерам блока CTI0 (Cross Trigger Interface) отладочной подсистемы. Это позволяет приостанавливать и возобновлять работу таймеров при возникновении определенных отладочных событий.

Подробнее см. в разделе 34 «Система отладки микросхемы».

### 31.2 Ссылки

1. Arm® Cortex® -M System Design Kit Revision: r1p1 Technical Reference Manual.



## 32. TIM0/TIM1/LPTIM: ТАЙМЕРЫ ОБЩЕГО НАЗНАЧЕНИЯ

### 32.1 Введение

Микросхема содержит три интервальных таймера TIM0, TIM1, LPTIM со следующими характеристиками:

- разрядность счетчика – 32 бита
- направление счета – декремент
- генерирование прерывания CPU при достижении таймером значения 0
- остановка/запуск таймеров с помощью блока Cross Trigger Interface (CTI) для целей отладки
- таймер LPTIM может функционировать в режиме STANDBY

Таймеры TIM0, TIM1, LPTIM имеют одинаковую структуру и далее обозначаются как TIMx.

### 32.2 Описание функционирования

#### 32.2.1 Структурная схема

На рисунке ниже показана структурная схема TIMx.

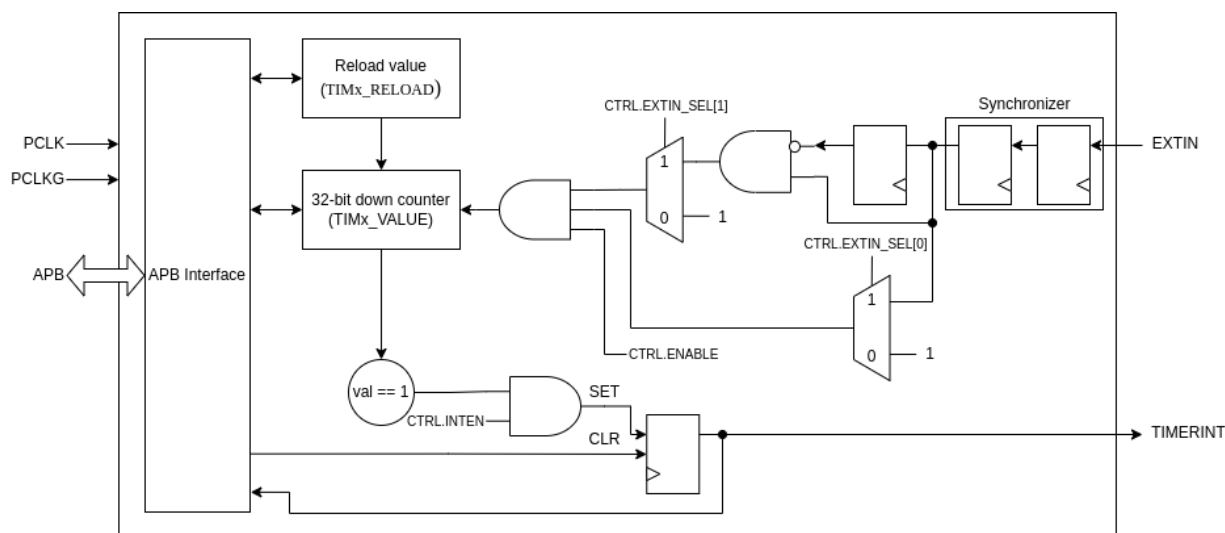


Рисунок 32.1. Структурная схема TIMx

#### 32.2.2 Внутренние сигналы

В следующей таблице показаны сигналы таймера TIMx и их подключение в системе.

**Таблица 32.1 Сигналы TIMx**

Сигнал	Подключение	Описание
PCLK	TIM0/TIM1: SYS_SYSCLK; LPTIM: LPCLK	Тактовый сигнал счетчика и регистра прерывания. Всегда включен.
PCLKG	TIM0/TIM1: SYS_SYSCLK; LPTIM: LPCLK	Тактовый сигнал интерфейса APB. Может быть отключен (кроме LPCLK).
EXTIN	TIM0/TIM1: CTIO.TIMHALTREQ/TIMRESTARTREQ; LPTIM: CTIO.LPTIMHALTREQ/LPTIMRESTARTREQ	Запрос остановки о возобновления работы от CTIO (см. раздел «Система отладки микросхемы»).
TIMERINT	CPU0, CPU1	Прерывание (см. раздел «Системная организация микросхемы»).

### 32.2.3 Режимы пониженного энергопотребления

Таймеры TIM0, TIM1 расположены в домене PD\_SYS и функционируют только в режиме RUN.

Таймер LPTIM расположен в домене PD\_AON и функционирует в режимах RUN и STANDBY.

### 32.2.4 Отладка

Входы EXTIN всех таймеров подключены к выходным сигналам-триггерам \*TIMHALTREQ и \*TIMRESTARTREQ блока CTIO (Cross Trigger Interface) отладочной подсистемы. Это позволяет приостанавливать и возобновлять работу таймеров при возникновении определенных отладочных событий.

Для возможности остановки/запуска таймеров с помощью CTI необходимо в регистре CTRL установить EXTIN\_SEL=1.

Подробнее см. в разделе 34 «Система отладки микросхемы».

## 32.3 Регистры TIMx

### 32.3.1 Перечень регистров TIMx

Таблица 32.2 Регистры TIMx

Условное обозначение	Описание	Исходное состояние	Смещение
TIMx_CTRL	Регистр управления таймером	0x0	0x000
TIMx_VALUE	Регистр текущего значения таймера	0x0	0x004
TIMx_RELOAD	Регистр значения перезагрузки таймера	0x0	0x008
TIMx_INTSTATUS	Регистр статуса/сброса прерывания	0x0	0x00C

### 32.3.2 Регистр управления таймером TIMx\_CTRL

Таблица 32.3 Описание регистра TIMx\_CTRL

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:4	-	RO	0	Резерв.
3	INTEN	RW	0	Включение сигнала прерывания таймера: 0x0: прерывание выключено; 0x1: прерывание включено
2:1	EXTIN_SEL	RW	0	Выбор функции входа EXTIN: 0x0: вход EXTIN не используется; 0x1: вход EXTIN используется как сигнал разрешения счета таймера при EXTIN=1. Применяется совместно с блоком Cross Trigger Interface (CTI); 0x2: вход EXTIN используется как сигнал тактирования таймера. Не применимо в данной системе; 0x3: резерв
0	ENABLE	RW	0	Разрешение работы таймера: 0x0: таймер выключен; 0x1: таймер включен

### 32.3.3 Регистр текущего значения таймера TIMx\_VALUE

Таблица 32.4 Описание регистра TIMx\_VALUE

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	VALUE	RW	0	Текущее значение счетчика таймера.

### 32.3.4 Регистр значения перезагрузки таймера TIMx\_RELOAD

Таблица 32.5 Описание регистра TIMx\_RELOAD

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:0	RELOAD	RW	0	Значение перезагрузки счетчика при достижении им значения 0.

### 32.3.5 Регистр статуса/сброса прерывания TIMx\_INTSTATUS

Таблица 32.6 Описание регистра TIMx\_INTSTATUS

Номер бита	Название	Тип доступа	Исходное состояние	Назначение
31:1	-	RO	0	Резерв.
0	INTSTATUS	RW	0	При чтении – значение статуса прерывания. Запись 1 сбрасывает статус прерывания. Запись 0 игнорируется.

## 33. DTIM: СДВОЕННЫЙ ТАЙМЕР ОБЩЕГО НАЗНАЧЕНИЯ

В микросхеме реализован сдвоенный интервальный таймер общего назначения со следующими возможностями:

- два интервальных таймера разрядностью 16 или 32 бит;
- режимы работы – интервальный, one-shot, свободный.

Таймеры могут служить источником прерываний для ядер CPU. Подробнее см. в главе 2.

Программная модель и логика функционирования таймера описаны в разделе 4.4 англоязычной документации [1].

### 33.1 Отладка

Вход EXTIN таймера подключен к выходным сигналам-триггерам блока CTI0 (Cross Trigger Interface) отладочной подсистемы. Это позволяет приостанавливать и возобновлять работу таймера при возникновении определенных отладочных событий.

Подробнее см. в разделе 34 «Система отладки микросхемы».

### 33.2 Ссылки

1. Arm® Cortex® -M System Design Kit Revision: r1p1 Technical Reference Manual.

## 34. СИСТЕМА ОТЛАДКИ МИКРОСХЕМЫ

### 34.1 Введение

Микросхема имеет широкие возможности по отладке, включая:

- отладку с помощью аппаратных точек останова;
- программную и аппаратную трассу;
- синхронизацию отладочных событий с помощью Cross-triggering.

Для доступа к отладочным ресурсам системы используются стандартные интерфейсы JTAG/Serial-wire. Для вывода и анализа данных в реальном времени используется порт трассы.

Схема отладки построена на следующих компонентах Arm Coresight:

- SWJ-DP: JTAG/Serial-wire debug port
- AHB-AP: AHB access port
- APB-AP: APB access port
- GPR: Granular Power Requester
- ROM table
- SCS: System control space
- Breakpoint Unit (BPU)
- Data Watchpoint and Trace Unit (DWT)
- Instrumentation Trace Macrocell (ITM)
- Cross Trigger Interface (CTI)
- Trace Port Interface Unit (TPIU)

Дополнительная информация может быть найдена в документах Arm®, указанных в разделе 34.2.13.

## 34.2 Описание функционирования

### 34.2.1 Структурная схема

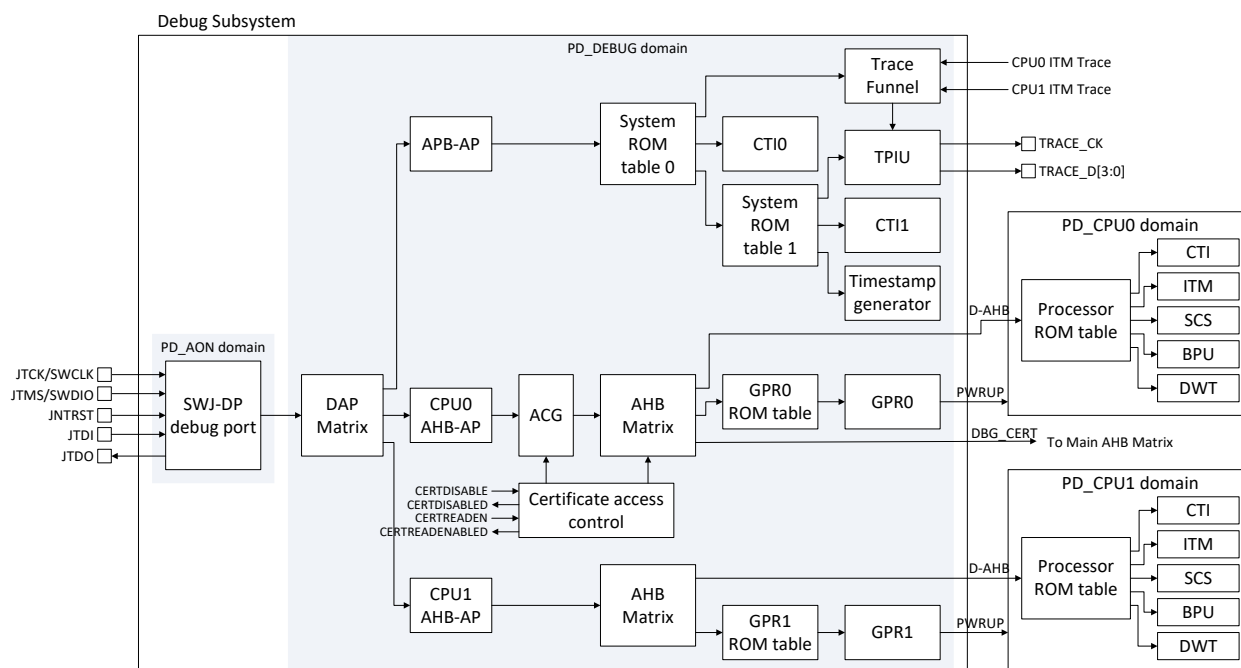


Рисунок 34.1. Структура отладки микросхемы

### 34.2.2 Выводы подсистемы отладки

В следующей таблице приведено описание выводов подсистемы отладки. Все выводы подключены к многофункциональным портам ввода-вывода (см. раздел 35.3 «Альтернативные функции выводов»).

Таблица 34.1. Описание выводов подсистемы отладки

Название вывода	Описание
JTCK/SWCLK	Тактовый сигнал JTAG / Тактовый сигнал Serial wire
JTMS/SWDIO	Выбор тестового режима JTAG / Вход-выход данных Serial wire
JNTRST	Сброс JTAG
JTDI	Вход данных JTAG
JTDO	Выход данных JTAG
TRACE_CK	Выход тактового сигнала интерфейса трассы
TRACE_D0	Выход данных интерфейса трассы, бит 0
TRACE_D1	Выход данных интерфейса трассы, бит 1
TRACE_D2	Выход данных интерфейса трассы, бит 2
TRACE_D3	Выход данных интерфейса трассы, бит 3

### 34.2.3 Домены питания и режимы работы

Все компоненты отладки микросхемы распределены по нескольким доменам питания:

- Порт SWJ-DP расположен в домене PD\_AON.
- Большая часть компонентов подсистемы отладки расположена в отключаемом домене PD\_DEBUG.
- Coresight компоненты процессоров Cortex-M33 расположены в соответствующих доменах PD\_CPU0 и PD\_CPU1.

Расположение порта SWJ-DP в неотключаемом домене питания позволяет использовать интерфейс JTAG/SWD для включения питания домена PD\_DEBUG и вывода системы из режима STANDBY с помощью внешнего отладчика.

Блоки Granular Power Requester предназначены для запроса включения питания доменов PD\_CPU0 и PD\_CPU1 процессоров.

### 34.2.4 Безопасность

Для контроля доступа к ресурсам отладки микросхемы предназначены сигналы DBGEN, NIDEN, SPIDEN, SPNIDEN. Подробнее – в разделе 2.7.8 «Контроль отладочного доступа».

### 34.2.5 Порт отладки SWJDP

Подробнее см. в [1].

### 34.2.6 Порты доступа AP

Подробнее см. в [1].

### 34.2.7 ROM-таблицы

Подробнее см. в [2, 3].

### 34.2.8 DWT

Блок DWT в составе каждого процессора содержит 4 компаратора DWT\_COMP0 – DWT\_COMP3 со следующими возможностями:

- Аппаратные точки наблюдения (watchpoints).
- Аппаратная трассировка совместно с ITM.
- Триггеры CMPMATCH для CTI.
- Сопоставление адреса инструкций и данных.
- Сопоставление значения данных (только DWT\_COMP3).

Подробнее о DWT см. в [2, 3].



### 34.2.9 ITM

Подробнее о ITM см. в [2, 3].

### 34.2.10 BPU

Блок BPU поддерживает до 8 аппаратных точек останова (breakpoints).

Подробнее о BPU см. в [2, 3].

### 34.2.11 TPIU

Блок TPIU получает два потока трассировочных данных от блоков ITM в каждом из процессоров Cortex-M33, выполняет их форматирование и вывод через порт трассы.

Порт трассы – синхронный параллельный порт, состоящий из тактового сигнала TRACE\_CLK и четырех линий данных TRACE\_D0 – TRACE\_D3. Используемое количество линий данных настраивается от 1 до 4. Тактовый сигнал TRACE\_CLK формируется делением пополам частоты FCLK. Данные передаются по обоим фронтам TRACE\_CLK.

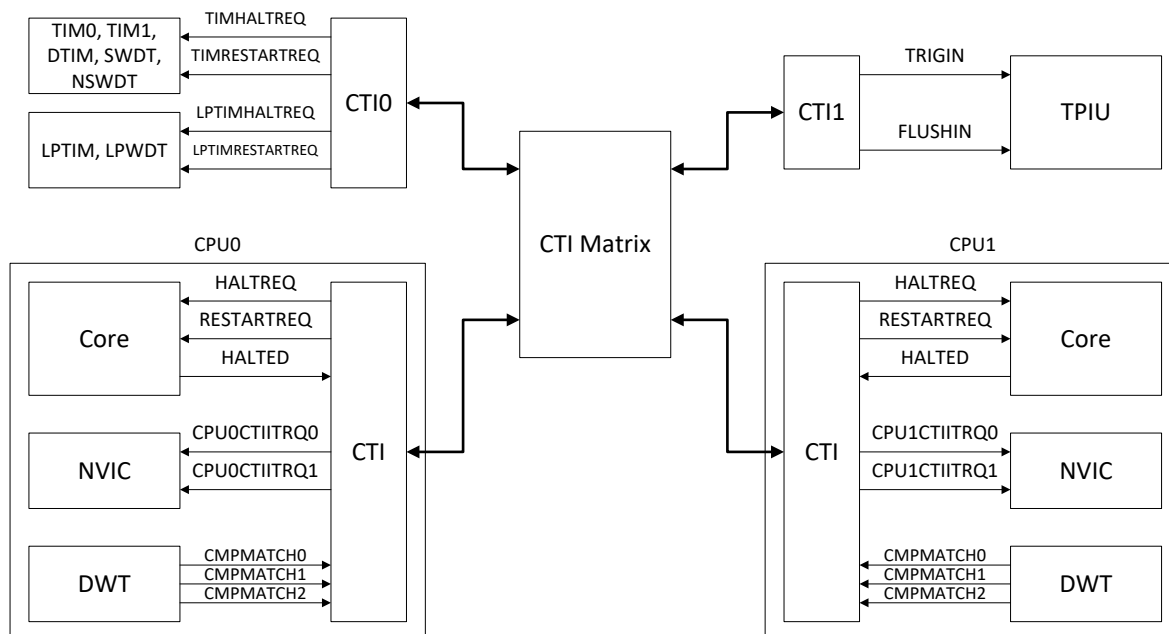
Подробнее о TPIU см. в [1].

### 34.2.12 CTI

Блоки CTI позволяют взаимодействовать отладочным компонентам между собой, а также с другими блоками системы посредством сигналов-триггеров.

Микросхема содержит несколько блоков CTI: блоки CTI0 и CTI1 в отладочной подсистеме, блоки CPU<sub>n</sub> CTI (n = 0, 1) в каждом процессоре Cortex-M33.

На следующем рисунке показана схема взаимодействия CTI.



**Рисунок 34.2. Схема взаимодействия CTI**

В следующих таблицах приведено назначение входов CTITRIGIN и выходов CTITRIGOUT для всех блоков CTI в системе.

**Таблица 34.2. Назначение входов блока CTI процессоров CPU<sub>n</sub>, n = 0, 1**

Номер	Сигнал	Источник	Описание	Подтверждение
0	HALTED	CPU <sub>n</sub>	Сигнал об остановке процессора и переходе в отладочный режим	Аппаратное
1	CMPMATCH0	DWT	Выход компаратора DWT 0	Аппаратное
2	CMPMATCH1	DWT	Выход компаратора DWT 1	Аппаратное
3	CMPMATCH2	DWT	Выход компаратора DWT 2	Аппаратное
4	-	-	Не используется	-
5	-	-	Не используется	-
6	-	-	Не используется	-
7	-	-	Не используется	-

**Таблица 34.3. Назначение выходов блока CTI процессоров CPU<sub>n</sub>, n = 0, 1**

Номер	Сигнал	Приемник	Описание	Подтверждение
0	HALTREQ	CPU <sub>n</sub>	Запрос остановки процессора и перехода в отладочный режим	Для снятия запроса после остановки CPU внешний отладчик должен записать регистр CTINTACK перед возобновлением работы
1	RESTARTREQ	CPU <sub>n</sub>	Запрос возобновления работы процессора и выхода из отладочного режима	Аппаратное
2	CPU <sub>n</sub> CTIIRQ0	CPU <sub>n</sub> NVIC	Запрос прерывания 0	Подтверждается обработчиком прерывания записью в регистр CTINTACK
3	CPU <sub>n</sub> CTIIRQ1	CPU <sub>n</sub> NVIC	Запрос прерывания 1	
4	-	-	Не используется	-
5	-	-	Не используется	-
6	-	-	Не используется	-

7	-	-	Не используется	-
---	---	---	-----------------	---

**Таблица 34.4. Назначение выходов блока СТИ0 отладочной подсистемы**

Номер	Сигнал	Приемник	Описание	Подтверждение
0	TIMHALTREQ	TIM0, TIM1, DTIM, SWDT, NSWDT	Запрос остановки таймеров. Подключается ко входу EXTIN таймеров TIM0, TIM1 и DTIM, а также ко входу WDOGCLKEN таймеров SWDT, NSWDT	Запрос снимается записью в регистр STINTACK
1	TIMRESTARTREQ	TIM0, TIM1, DTIM, SWDT, NSWDT	Запрос возобновления работы таймеров	Аппаратное
2	LPTIMHALTREQ	LPTIM, LPWDT	Запрос остановки таймеров. Подключается ко входу EXTIN таймера LPTIM и ко входу WDOGCLKEN таймера LPWDT	Запрос снимается записью в регистр STINTACK
3	LPTIMRESTARTREQ	LPTIM, LPWDT	Запрос возобновления работы таймеров	Аппаратное
4	-	-	Не используется	-
5	-	-	Не используется	-
6	-	-	Не используется	-
7	-	-	Не используется	-

**Таблица 34.5. Назначение выходов блока СТИ1 отладочной подсистемы**

Номер	Сигнал	Приемник	Описание	Подтверждение
0	TRIGIN	TPIU	Сигнал trigin блока TPIU	Аппаратное
1	FLUSHIN	TPIU	Сигнал flushin блока TPIU	Аппаратное
2	-	-	Не используется	-
3	-	-	Не используется	-
4	-	-	Не используется	-
5	-	-	Не используется	-
6	-	-	Не используется	-
7	-	-	Не используется	-

Входы STITRIGIN[7:0] блоков СТИ0 и СТИ1 отладочной подсистемы не используются.

Все выходы STITRIGOUT маскируются при DBGEN = 0. Все входы STITRIGIN маскируются при NIDEN = 0.

Подробнее о СТИ см. в [1, 2].

### 34.2.13 Ссылки

1. ARM® CoreSight™ SoC-400 Technical Reference Manual.
2. Arm® Cortex®-M33 Processor Technical Reference Manual.
3. ARMv8-M Architecture Reference Manual.

## 35. ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ

Параметры корпуса микросхемы:

- тип корпуса: BGA;
- размер корпуса (габариты): 7x7 мм;
- шаг по выводам: 0,5 мм;
- матрица выводов: 12 x 12;
- общее число выводов: 132.

### 35.1 Перечень выводов

Таблица 35.1. Перечень и параметры выводов микросхемы

Номер вывода	Название вывода	Тип вывода	Тип I/O	Назначение вывода
M7	VBAT	S	-	Резервное (батарейное) питание (1.6 – 3.3 В) для RWC и Vascup-памяти
K7	VBKP	S	-	Напряжение питания Vascup-памяти. Используется для отладки.
L6, L7	VDDA	S	-	Аналоговое питание (3.3 В) и земля
K8, K9, L9	VSSA	S	-	
M10	VDDCA	S	-	
L10	VSSCA	S	-	Аналоговое питание (0.9 – 1.1 В) и земля PLL
G4, G7, H9, J4, J8	VDDIO	S	-	
D4, D7, D9, F6, F9, J6	VDDC	S	-	Питание цифрового ядра (0.9 – 1.1 В)
E4, E9, D6, G6, G9, F4, F7, H4, J5, J7, J9, M11	VSS	S	-	Общая цифровая земля
K1, K2, K3	VDDREG	S	-	Входное напряжение DC-DC конвертера (3.3 В)
M1, M2, M3	VLXREG	S	-	Выходное напряжение DC-DC конвертера, подключение внешней индуктивности
L1, L2, L3	VSSREG	S	-	Земля DC-DC конвертера
K6	VFB	A	-	Вывод монитора питания ядра, подключить к внешнему конденсатору DC-DC (цепь VDDC)
J3	VDDIO2	S	-	Питание (1.8/3.3 В) буферов I/O выводов PD6 – PD11
J1	VPPFL	A	-	Выводы для тестирования FLASH памяти. Оставить неподключенными
L4	TMFL0	A	-	
M4	TMFL1	A	-	
K4, K5	NC	-	-	Не подключены
B6, B7	VSSUSB	S	-	Земля USB PHY
C7	VDDUSB	S	-	Питание USB PHY (3.3 В)
C8	VDDCUSB	S	-	Питание USB PHY (1.1 В)
C6	USBTXRTUNE	A	-	Вывод для подключения опорного резистора (200 Ом ± 1%) для трансивера USB
A8	USBVBUS	S	-	Сигналы интерфейса USB
A7	USBDP	A	-	
A6	USBDM	A	-	
D8	USBC1	A	-	
B8	USBC2	A	-	

Номер вывода	Название вывода	Тип вывода	Тип I/O	Назначение вывода
M6	SRSTn	I	-	Системный сброс, активный уровень – 0, имеет подтяжку к питанию
K12	PORSTn	I	-	Вход внешнего power-on сброса, активный уровень – 0, имеет подтяжку к питанию
M12	XTI	A	-	Вход осциллятора HFE для подключения кварцевого резонатора либо внешнего генератора
L12	XTO	A		Выход осциллятора HFE для подключения кварцевого резонатора
M8	XTI32	A	-	Вход осциллятора для подключения кварцевого резонатора 32.768 кГц либо внешнего генератора
M9	XTO32	A	-	Выход осциллятора для подключения кварцевого резонатора 32.768 кГц
L8	WKUP	I	-	Вход внешнего прерывания в батарейном домене
L5	ANATEST	A	-	Вывод для тестирования PMU, оставить неподключенным
M5	PMUDIS	I	-	Вход для резервирования функций PMU, имеет подтяжку к земле
L11	DFTTM	I	-	Включение режима производственного тестирования. Имеет подтяжку к земле. 0x0 – функциональный режим; 0x1 – тестовый режим DFT
B2	PA0	I/O	FT	Многофункциональные выводы порта PA. Основная функция – GPIO0. Альтернативные функции перечислены в Таблица 35.2.
A2	PA1	I/O	FT	
C4	PA2	I/O	FT	
A3	PA3	I/O	FT	
B4	PA4	I/O	FT	
C5	PA5	I/O	FT	
A4	PA6	I/O	FT, I2C	
B5	PA7	I/O	FT, I2C	
A5	PA8	I/O	FT	
B3	PA9	I/O	FT	
H12	PA10	I/O	FT	
J10	PA11	I/O	FT	
J11	PA12	I/O	FT	
K10	PA13	I/O	FT	
J12	PA14	I/O	FT	
K11	PA15	I/O	FT	
E11	PB0	I/O	FT	Многофункциональные выводы порта PB. Основная функция – GPIO1. Альтернативные функции перечислены в Таблица 35.3.
F10	PB1	I/O	FT	
F11	PB2	I/O	FT	
A12	PB3	I/O	FT	
B12	PB4	I/O	FT	
C11	PB5	I/O	FT	
C12	PB6	I/O	FT	
D10	PB7	I/O	FT	
D11	PB8	I/O	FT	
D12	PB9	I/O	FT, I2C	
E10	PB10	I/O	FT, I2C	
E12	PB11	I/O	FT	
B10	PB12	I/O	FT	
A10	PB13	I/O	FT	
A11	PB14	I/O	FT	

Номер вывода	Название вывода	Тип вывода	Тип I/O	Назначение вывода
B11	PB15	I/O	FT	Многофункциональные выводы порта PC. Основная функция – GPIO2. Альтернативные функции перечислены в Таблица 35.4.
B9	PC0	I/O	FT	
A9	PC1	I/O	FT	
C9	PC2	I/O	FT	
C10	PC3	I/O	FT	
G10	PC4	I/O	FT	
G11	PC5	I/O	FT	
F12	PC6	I/O	FT	
H10	PC7	I/O	FT	
G12	PC8	I/O	FT	
H11	PC9	I/O	FT	
D3	PC10	I/O	FT	
D5	PC11	I/O	FT	
C2	PC12	I/O	FT	
B1	PC13	I/O	FT	
C3	PC14	I/O	FT	
A1	PC15	I/O	FT	Многофункциональные выводы порта PD. Основная функция – GPIO3. Альтернативные функции перечислены в Таблица 35.5.
J2	PD0	I/O	FT	
H2	PD1	I/O	FT	
H1	PD2	I/O	FT	
H3	PD3	I/O	FT	
G1	PD4	I/O	FT	
G2	PD5	I/O	FT	
G3	PD6	I/O	FT	
F2	PD7	I/O	FT	
F1	PD8	I/O	FT	
F3	PD9	I/O	FT	
E1	PD10	I/O	FT	
E2	PD11	I/O	FT	
D1	PD12	I/O	FT	
E3	PD13	I/O	FT	
D2	PD14	I/O	FT	
C1	PD15	I/O	FT	

Обозначения:

S – вывод питания

A – аналоговый вход/выход

I – цифровой вход

I/O – цифровой вход/выход

FT – 5 В толерантность

I2C – поддержка режима I2C

## 35.2 Расположение выводов на корпусе

	1	2	3	4	5	6	7	8	9	10	11	12
A	PC15	PA1	PA3	PA6	PA8	USBDM	USBDP	USBVBUS	PC1	PB13	PB14	PB3
B	PC13	PA0	PA9	PA4	PA7	VSSUSB	VSSUSB	USBC2	PC0	PB12	PB15	PB4
C	PD15	PC12	PC14	PA2	PA5	USBTXRT UNE	VDDUSB	VDDCUSB	PC2	PC3	PB5	PB6
D	PD12	PD14	PC10	VDDC	PC11	VSS	VDDC	USBC1	VDDC	PB7	PB8	PB9
E	PD10	PD11	PD13	VSS					VSS	PB10	PB0	PB11
F	PD8	PD7	PD9	VSS		VDDC	VSS		VDDC	PB1	PB2	PC6
G	PD4	PD5	PD6	VDDIO		VSS	VDDIO		VSS	PC4	PC5	PC8
H	PD2	PD1	PD3	VSS					VDDIO	PC7	PC9	PA10
J	VPPFL	PD0	VDDIO2	VDDIO	VSS	VDDC	VSS	VDDIO	VSS	PA11	PA12	PA14
K	VDDREG	VDDREG	VDDREG	NC	NC	VFB	VBKP	VSSA	VSSA	PA13	PA15	PORSTn
L	VSSREG	VSSREG	VSSREG	TMFLO	ANATEST	VDDA	VDDA	WKUP	VSSA	VSSCA	DFTTM	XTO
M	VLXREG	VLXREG	VLXREG	TMFL1	PMUDIS	SRSTn	VBAT	XTI32	XTO32	VDDCA	VSS	XTI

**Рисунок 35.1. Расположение выводов на корпусе**

### 35.3 Альтернативные функции выводов

Таблица 35.2. Альтернативные функции выводов порта A

Port		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
		SYSTEM	PWM, VTU0, VTU1	I2C0, I2C1, I2S	SPI0, SPI1	UART0, UART1, UART2, UART3	CAN, GNSS, USB	QSPI, SPI2	SDMMC, SMC
Port A	PA0		PWM_OUTA0			UART0_TXD	GNSS_SIG1_I0		
	PA1		PWM_OUTB0			UART0_RXD	GNSS_SIG1_I1		
	PA2		PWM_OUTA1		SPI0_SCK		GNSS_SIG1_Q0		
	PA3		PWM_OUTB1		SPI0_MOSI		GNSS_SIG1_Q1		
	PA4		PWM_OUTA2		SPI0_MISO		GNSS_SIG2_I0	SPI2_SCK	
	PA5		PWM_OUTB2		SPI0_SS		GNSS_SIG2_I1	SPI2_MOSI	
	PA6		PWM_OUTA3	I2C0_SDA			GNSS_SIG2_Q0	SPI2_MISO	
	PA7		PWM_OUTB3	I2C0_SCL			GNSS_SIG2_Q1	SPI2_SS	
	PA8		VTU0_TIO1		SPI0_SS		GNSS_MCLK		
	PA9	PMU_DTB	VTU0_TIO2		SPI1_SCK		GNSS_OPPS		SMC_DA5
	PA10	JTDO	VTU0_TIO3						SMC_DA6
	PA11	JTDI	VTU0_TIO4						SMC_DA7
	PA12	JNTRST	VTU0_TIO5						SMC_DA8
	PA13	JTCK/SWCLK							SMC_DA9
	PA14	JTMS/SWDIO							SMC_DA10
	PA15	MCO		I2S_EXTCLK	SPI0_SS	UART0_CK			



Таблица 35.3. Альтернативные функции выводов порта B

Port		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
		SYSTEM	PWM, VTU0, VTU1	I2C0, I2C1, I2S	SPI0, SPI1	UART0, UART1, UART2, UART3	CAN, GNSS, USB	QSPI, SPI2	SDMMC, SMC
Port B	PB0			I2S_SDO	SPI0_SS	UART0_TXD			
	PB1			I2S_SCLK		UART0_RXD			SMC_CRE
	PB2	TRACE_CK		I2S_WS	SPI1_SCK	UART3_RXD		SPI2_SCK	
	PB3		PWM_OUTA0		SPI1_SS		ULPI_D0		SMC_A16
	PB4		PWM_OUTB0		SPI1_SS		ULPI_D1		SMC_A17
	PB5		PWM_OUTA1		SPI1_SS		ULPI_D2		SMC_A18
	PB6		PWM_OUTB1		SPI1_SS		ULPI_D3		SMC_A19
	PB7	TRACE_D3	PWM_OUTA2	I2S_SDO	SPI1_MOSI	UART0_CTS/ UART0_RE	CAN_TXD		SMC_A20
	PB8	TRACE_D2	PWM_OUTB2	I2S_WS	SPI1_SS	UART0_RTS/ UART0_DE	CAN_RXD		SMC_A21
	PB9	TRACE_D1	PWM_OUTA3	I2C1_SDA	SPI1_SS	UART1_TXD	CAN_TXD	SPI2_MOSI	SMC_A22
	PB10	TRACE_D0	PWM_OUTB3	I2C1_SCL	SPI1_SS	UART1_RXD	CAN_RXD	SPI2_MISO	SMC_A23
	PB11		VTU0_TIO6		SPI1_SCK		GNSS_OPDS		SMC_DA0
	PB12		VTU0_TIO7				ULPI_D4	SPI2_SCK	SMC_DA1
	PB13		VTU0_TIO8				ULPI_D5	SPI2_MOSI	SMC_DA2
	PB14						ULPI_D6	SPI2_MISO	SMC_DA3
	PB15						ULPI_D7	SPI2_SS	SMC_DA4

Таблица 35.4. Альтернативные функции выводов порта C

Port		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
		SYSTEM	PWM, VTU0, VTU1	I2C0, I2C1, I2S	SPI0, SPI1	UART0, UART1, UART2, UART3	CAN, GNSS, USB	QSPI, SPI2	SDMMC, SMC
Port C	PC0				SPI0_MOSI		ULPI_NXT	SPI2_SCK	SMC_DA5
	PC1				SPI0_MISO		ULPI_CK	SPI2_MOSI	SMC_DA6
	PC2				SPI0_SCK		ULPI_DIR	SPI2_MISO	SMC_DA7
	PC3				SPI0_SS		ULPI_STP	SPI2_SS	SMC_DA8
	PC4			I2C0_SDA	SPI1_MOSI	UART2_TXD	CAN_TXD		SMC_DA9
	PC5			I2C0_SCL	SPI1_MISO	UART2_RXD	CAN_RXD		SMC_DA10
	PC6		VTU1_TIO1			UART0_CK			SMC_DA11
	PC7		VTU1_TIO2			UART0_CTS/ UART0_RE			SMC_DA12
	PC8		VTU1_TIO3			UART0_RTS/ UART0_DE			SMC_DA13
	PC9		VTU1_TIO4			UART2_TXD			SMC_DA14
	PC10		VTU1_TIO5			UART2_RXD			SMC_DA15
	PC11		VTU1_TIO6						SMC_NWE
	PC12		VTU1_TIO7						SMC_CLK
	PC13		VTU1_TIO8						SMC_NOE
	PC14				SPI1_SS				SMC_NCS0
	PC15			I2S_SCLK	SPI1_SS	UART3_TXD	GNSS_OPDS	SPI2_SS	SMC_NCS1

Таблица 35.5. Альтернативные функции выводов порта D

Port		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
		SYSTEM	PWM, VTU0, VTU1	I2C0, I2C1, I2S	SPI0, SPI1	UART0, UART1, UART2, UART3	CAN, GNSS, USB	QSPI, SPI2	SDMMC, SMC
Port D	PD0				SPI1_SS	UART1_TXD			SMC_NWAIT
	PD1				SPI1_SS	UART1_RXD			SMC_NADV
	PD2		VTU0_TIO1	I2C0_SDA				QSPI_IO0	SMC_NBLS0
	PD3		VTU0_TIO2	I2C0_SCL				QSPI_IO1	SMC_NBLS1
	PD4		VTU0_TIO3	I2C1_SDA	SPI1_MOSI	UART3_TXD	CAN_TXD	QSPI_SCK	SDMMC_CD
	PD5		VTU0_TIO4	I2C1_SCL	SPI1_MISO	UART3_RXD	CAN_RXD	QSPI_SS	SDMMC_WP
	PD6	FBIST_TCK							SDMMC_CK
	PD7								SDMMC_CMD
	PD8								SDMMC_D0
	PD9	FBIST_TMS							SDMMC_D1
	PD10	FBIST_TDI			SPI0_SS				SDMMC_D2
	PD11	FBIST_TDO			SPI0_SS				SDMMC_D3
	PD12	TRACE_D0	VTU0_TIO5	I2S_SDO	SPI1_SS	UART1_TXD	CAN_TXD	QSPI_IO0	SDMMC_D4
	PD13	TRACE_D1	VTU0_TIO6	I2S_SCLK	SPI1_SS	UART1_RXD	CAN_RXD	QSPI_IO1	SDMMC_D5
	PD14	TRACE_D2	VTU0_TIO7	I2S_WS	SPI1_SS	UART3_TXD	CAN_TXD	QSPI_IO2	SDMMC_D6
	PD15	TRACE_D3	VTU0_TIO8		SPI1_SS	UART3_RXD	CAN_RXD	QSPI_IO3	SDMMC_D7

## 36. СПРАВОЧНАЯ ИНФОРМАЦИЯ

### 36.1 Предельно-допустимые характеристики

Таблица 36.1. Предельно-допустимое напряжение питания

Обозначение параметра	Наименование параметра, единица измерения	Мин.	Ном.	Макс.	Примечание
U <sub>VDDC</sub>	Напряжение питания ядра, В	0.81	0.9	0.99	VLEVEL0
		0.99	1.1	1.21	VLEVEL2
U <sub>VDDCA</sub>	Напряжение питания PLL, В	0.81	0.9/1.1	1.21	Должно совпадать с U <sub>VDDC</sub>
U <sub>VDDCUSB</sub>	Напряжение питания ядра USB PHY, В	0.81	0.9/1.1	1.21	Должно совпадать с U <sub>VDDC</sub>
U <sub>VDDIO</sub>	Напряжение питания буферов ввода-вывода (кроме PD6 – PD11) и Flash-памяти, В	1.62	1.8/3.3	3.63	
U <sub>VDDIO2</sub>	Напряжение питания буферов ввода-вывода PD6 – PD11, В	1.62	1.8/3.3	3.63	
U <sub>VDDA</sub>	Напряжение питания аналоговых блоков, В	2.97	3.3	3.63	
U <sub>VBAT</sub>	Напряжение питания батарейного домена, В	1.62	3.3	3.63	
U <sub>VDDREG</sub>	Входное напряжение питания внутреннего DC-DC регулятора, В	1.6	3.3	3.63	Должно совпадать с U <sub>VDDA</sub>
U <sub>VDDUSB</sub>	Напряжение питания USB PHY, В	2.97	3.3	3.63	Допускает неподачу питания

Таблица 36.2. Предельно-допустимые частоты внутренних тактовых сигналов

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Макс.
F <sub>FCLK</sub>	Частота тактового сигнала FCLK, МГц	U <sub>VDDC</sub> = 0.9 В ± 10 %	0	50
		U <sub>VDDC</sub> = 1.1 В ± 10 %	0	150
F <sub>SYCLK</sub>	Частота тактового сигнала SYCLK, МГц	U <sub>VDDC</sub> = 0.9 В ± 10 %	0	25
		U <sub>VDDC</sub> = 1.1 В ± 10 %	0	50
F <sub>QSPICLK</sub>	Частота тактового сигнала QSPICLK, МГц	U <sub>VDDC</sub> = 0.9 В ± 10 %	0	30
		U <sub>VDDC</sub> = 1.1 В ± 10 %		80
F <sub>GNSSCLK</sub>	Частота тактового сигнала GNSSCLK, МГц	U <sub>VDDC</sub> = 1.1 В ± 10 %	0	80
F <sub>I2SCLK</sub>	Частота тактового сигнала I2SCLK, МГц	U <sub>VDDC</sub> = 0.9 В ± 10 %	0	16
		U <sub>VDDC</sub> = 1.1 В ± 10 %		25

### 36.2 Энергопотребление

#### 36.2.1 Пиковые токи

Приведенные в таблице данные являются справочными.

Таблица 36.3. Пиковые токи потребления

Цепь питания	Пиковый ток	Единица измерения
VDDC	200	мА
VDDCA	н.д. <sup>3)</sup>	мА
VDDIO	н.д. <sup>3)</sup>	мА
VDDIO2	н.д. <sup>3)</sup>	мА
VDDA	н.д. <sup>3)</sup>	мА
VBAT	н.д. <sup>3)</sup>	мА
VDDREG	н.д. <sup>3)</sup>	мА
VDDUSB	35	мА
VDDCUSB	15	мА

## 36.3 Характеристики буферов ввода-вывода

### 36.3.1 Статические характеристики

В следующей таблице приведены статические характеристики буферов ввода-вывода портов PA[15:0], PB[15:0], PC[15:0], PD[15:0], а также выводов XTI, PORSTn, SRSTn, PMUDIS, DFTTM. Выводы PA6, PA7, PB9, PB10 работают в режиме общего назначения (I2CMODE = 0).

Таблица 36.4. Статические характеристики буферов ввода-вывода

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Макс.
V <sub>IH</sub>	Входное напряжение высокого уровня, В	$U_{VDDIO}^3 = 3.3 \text{ В} \pm 10 \%$	2.0	$U_{VDDIO}^3$
		$U_{VDDIO}^3 = 1.8 \text{ В} \pm 10 \%$	$0.7 \times U_{VDDIO}^3$	$U_{VDDIO}^3$
V <sub>IL</sub>	Входное напряжение низкого уровня, В	$U_{VDDIO}^3 = 3.3 \text{ В} \pm 10 \%$	$U_{VSS}$	0.8
		$U_{VDDIO}^3 = 1.8 \text{ В} \pm 10 \%$	$U_{VSS}$	$0.3 \times U_{VDDIO}^3$
V <sub>OH</sub>	Выходное напряжение высокого уровня, В	$U_{VDDC} = 0.81, U_{VDDIO}^3 = 3.0 \text{ В}, I_{OH} = 2 \text{ мА}, 4 \text{ мА}, 8 \text{ мА}, 12 \text{ мА}$	2.4	$U_{VDDIO}^3$
		$U_{VDDC} = 0.81, U_{VDDIO}^3 = 1.62 \text{ В}, I_{OH} = 2 \text{ мА}, 4 \text{ мА}, 8 \text{ мА}, 12 \text{ мА}$	$0.8 \times U_{VDDIO}^3$	$U_{VDDIO}^3$
V <sub>OL</sub>	Выходное напряжение низкого уровня, В	$U_{VDDC} = 0.81, U_{VDDIO}^3 = 3.0 \text{ В}, I_{OL} = 2 \text{ мА}, 4 \text{ мА}, 8 \text{ мА}, 12 \text{ мА}$	$U_{VSS}$	0.4
		$U_{VDDC} = 0.81, U_{VDDIO}^3 = 1.62 \text{ В}, I_{OL} = 2 \text{ мА}, 4 \text{ мА}, 8 \text{ мА}, 12 \text{ мА}$	$U_{VSS}$	$0.2 \times U_{VDDIO}^3$
I <sub>PAD</sub>	Входной ток утечки, мкА	$U_{VDDIO}^3 = 3.63 \text{ В}, V_I = 0 \text{ В}$ или 3.63 В	-1	1
I <sub>PAD</sub>	Входной ток утечки Fail-safe <sup>1</sup> , мкА	$U_{VDDIO}^3 = 0 \text{ В}, V_I = 3.63 \text{ В}$	-	1
I <sub>OZ</sub>	Ток утечки при выключенном драйвере, мкА	$U_{VDDIO}^3 = 3.63 \text{ В}, V_I = 0 \text{ В}$ или 3.63 В	-	1.2
R <sub>PU</sub>	Сопротивление резистора подтяжки к питанию, кОм	$V_I = V_{IH}, U_{VDDIO}^3 = 3.0$	9	19.4
		$V_I = V_{IH}, U_{VDDIO}^3 = 1.62$	11.2	32.4
R <sub>PD</sub>	Сопротивление резистора подтяжки к земле, кОм	$V_I = V_{IL}, U_{VDDIO}^3 = 3.0$	6,7	16
		$V_I = V_{IL}, U_{VDDIO}^3 = 1.62$	9.4	32.4

$V_H$	Входной гистерезис <sup>2</sup> , В	-	0.16	-
Примечания: 1) Входной ток утечки Fail-safe – это ток утечки через PAD к VDDIO или VSS, когда на VDDIO отсутствует напряжение. 2) Применимо для вывода SRSTn, а также для выводов PA[15:0], PB[15:0], PC[15:0], PD[15:0], если для них включен триггер Шмитта 3) VDDIO2 для выводов PD6 – PD11				

**Таблица 36.5 Разрешенная токовая нагрузка для различных режимов Drive strength**

Drive strength	$I_{OH}$ , $U_{VDDIO} = 3.3V$	$I_{OL}$ , $U_{VDDIO} = 3.3V$	$I_{OH}$ , $U_{VDDIO} = 3.3V$	$I_{OL}$ , $U_{VDDIO} = 3.3V$
2	2.45 mA	2.43 mA	1.35 mA	1.28 mA
4	4.88 mA	4.96 mA	2.7 mA	2.64 mA
8	9.77 mA	9.84 mA	5.4 mA	5.23 mA
12	14.6 mA	14.7 mA	8.1 mA	7.85 mA
Значения токов получены в следующих условиях: <ul style="list-style-type: none"> <li>3.3V SS/2.97V/125°C.</li> <li>1.8V SS/1.62V/125°C.</li> </ul>				

### 36.3.2 Динамические характеристики

В следующей таблице приведена максимальная частота работы выходных буферов портов PA, PB, PC, PD за исключением выводов PA6, PA7, PB9, PB10.

Максимальная частота определяется при следующих параметрах:

- коэффициент заполнения в диапазоне 47 % - 53 %
- $U_{VDDIO}/U_{VDDIO2} = 3.3 V \pm 10 \%$
- $T_j = 125^\circ C$

**Таблица 36.6. Максимальная частота работы выходных буферов**

DS	SR	Тип драйвера	Нагрузка, пФ	Максимальная частота, МГц
0x0	1	2 mA, slow	5	50
0x0	0	2 mA, fast	5	75
0x1	1	4 mA, slow	10	50
0x1	0	4 mA, fast	10	90
0x2	1	8 mA, slow	20	50
0x2	0	8 mA, fast	20	100
0x3	1	12 mA, slow	30	50
0x3	0	12 mA, fast	30	120

В следующей таблице приведена максимальная частота работы выходных буферов выводов PA6, PA7, PB9, PB10 в режиме общего назначения (I2CMODE = 0).

**Таблица 36.7. Максимальная частота работы выходных буферов (выводы PA6, PA7, PB9, PB10)**

DS	Тип драйвера	Нагрузка, пФ	Максимальная частота, МГц
0x0	4 мА	10	50
0x1	20 мА	50	80

### 36.3.3 Характеристики буферов ввода-вывода в режиме I2C

В следующей таблице приведены характеристики буферов ввода-вывода для выводов микросхемы PA6, PA7, PB9, PB10 в режиме I2C (I2CMODE = 1).

Характеристики приведены для напряжения питания  $U_{VDDIO} = 3.3 \text{ В} \pm 10 \%$ .

**Таблица 36.8. Характеристики буферов ввода-вывода в режиме I2C**

Обозначение параметра	Параметр, единица измерения	Условия	Standard-mode		Fast-mode		Fast-mode plus	
			Мин.	Макс.	Мин.	Макс.	Мин.	Макс.
$V_{IH}$	Входное напряжение высокого уровня, В	-	$0.7 \times U_{VDDIO}$	5.5	$0.7 \times U_{VDDIO}$	5.5	$0.7 \times U_{VDDIO}$	5.5
$V_{IL}$	Входное напряжение низкого уровня, В	-	-0.5	$0.3 \times U_{VDDIO}$	-0.5	$0.3 \times U_{VDDIO}$	-0.5	$0.3 \times U_{VDDIO}$
$V_{hys}$	Входной гистерезис триггера Шмитта, В	-	-	-	$0.05 \times U_{VDDIO}$	-	$0.05 \times U_{VDDIO}$	-
$V_{OL1}$	Выходное напряжение низкого уровня, В	Ток потребления 3 мА; $U_{VDDIO} > 2 \text{ В}$	0	0.4	0	0.4	0	0.4
$V_{OL2}$	Выходное напряжение низкого уровня, В	Ток потребления 2 мА; $U_{VDDIO} > 2 \text{ В}$	-	-	0	$0.2 \times U_{VDDIO}$	0	$0.2 \times U_{VDDIO}$
$I_{OL}$	Выходной ток низкого уровня, мА	$V_{OL} = 0.4 \text{ В}$	3	-	3	-	20	-
		$V_{OL} = 0.6 \text{ В}$	-	-	6	-	-	-
$t_{of}$	Время спада выходного напряжения от $V_{IHmin}$ до $V_{ILmax}$ , нс	-	-	250	$20 \times (U_{VDDIO} / 5.5 \text{ В})$	250	$20 \times (U_{VDDIO} / 5.5 \text{ В})$	120
$I_i$	Входной ток, мА	$0.1 \times U_{VDDIO} < V_i < 0.9 \times U_{VDDIOmax}$	10	10	-10	+10	-10	+10
$C_i$	Емкость каждого вывода, пФ	-	-	10	-	10	-	10

$F_{SCL}$	Тактовая частота SCL, кГц	-	0	100	0	400	0	1000
$t_r$	Время нарастания сигналов SDA, SCL, нс	-	-	1000	20	300	-	120
$t_f$	Время спада сигналов SDA, SCL, нс	-	-	300	$20 \times (U_{VDDI} / 5.5 \text{ В})$	300	$20 \times (U_{VDDI} / 5.5 \text{ В})$	120
$C_b$	Емкость нагрузки для каждой линии, пФ	-	10	400	10	400	10	550

В следующих таблицах приведены значения сопротивления подтяжки ( $R_p$ ) для различных режимов. Указанные значения емкости нагрузки и сопротивления  $R_p$  включают емкость и сопротивление паразитных составляющих печатной платы.

**Таблица 36.9. Значения  $R_p$  для режима Fast-mode plus**

Частота, кГц	Емкость нагрузки, пФ	$U_{VDDIO}$ , В	Напряжение на шине, В	$t_r$ , нс	$t_f$ , нс	$R_p(\min) - R_p(\max)$ , Ом
1000	10	2.97 – 3.63	2.97 – 3.63	10.8 – 120	< 120	161.5 – 9100
1000	10	2.97 – 3.63	4.5 – 5.5	16.36 – 120	< 120	254 – 8600
1000	400	2.97 – 3.63	2.97 – 3.63	10.8 – 120	< 120	161.5 – 345
1000	400	2.97 – 3.63	4.5 – 5.5	16.36 – 120	< 120	254 – 345
1000	550	2.97 – 3.63	2.97 – 3.63	10.8 – 120	< 120	161.5 – 254
1000	550	2.97 – 3.63	4.5 – 5.5	16.36 – 120	< 120	254 – 254

**Таблица 36.10. Значения  $R_p$  для режима Fast-mode**

Частота, кГц	Емкость нагрузки, пФ	$U_{VDDIO}$ , В	Напряжение на шине, В	$t_r$ , нс	$t_f$ , нс	$R_p(\min) - R_p(\max)$ , Ом
400	10	2.97 – 3.63	2.97 – 3.63	10.8 – 250	20 – 300	2000 – 22800
400	10	2.97 – 3.63	4.5 – 5.5	16.36 – 250	20 – 300	2000 – 23000
400	400	2.97 – 3.63	2.97 – 3.63	10.8 – 250	20 – 300	505 – 870
400	400	2.97 – 3.63	4.5 – 5.5	16.36 – 250	20 – 300	816.7 – 870

**Таблица 36.11. Значения  $R_p$  для режима Standard-mode**

Частота, кГц	Емкость нагрузки, пФ	$U_{VDDIO}$ , В	Напряжение на шине, В	$t_r$ , нс	$t_f$ , нс	$R_p(\min) - R_p(\max)$ , кОм
100	10	2.97 – 3.63	2.97 – 3.63	< 250	< 1000	1.077 – 76
100	10	2.97 – 3.63	4.5 – 5.5	< 250	< 1000	1.7 – 77
100	400	2.97 – 3.63	2.97 – 3.63	< 250	< 1000	1.077 – 2.9
100	400	2.97 – 3.63	4.5 – 5.5	< 250	< 1000	1.7 – 2.9

Входные буферы с поддержкой режима I2C содержат фильтр подавления импульсов длительностью менее 50 нс.



## 36.4 Характеристики PLL

Таблица 36.12. Характеристики PLL

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
$F_{PLLREF}$	Частота опорного тактового сигнала PLL, МГц	-	0.03	-	50
$F_{PLLCLK}$	Частота выходного тактового сигнала PLL, МГц	Полный диапазон	1.88	-	375
		OD = 1	30	-	375
DutyCycle	Коэффициент заполнения выходного тактового сигнала, %	OD = 1	45	50	55
		OD $\neq$ 1	48	50	52
Jitter <sub>OUT</sub>	Периодический джиттер (пик-пик) выходного тактового сигнала, % от периода	-	-	-	$\pm 2.5$
I <sub>DD</sub>	Ток потребления, мкА	OD = 1, $f_{PLLCLK}$ = 30 МГц	-	20	-
t <sub>LOCK</sub>	Время захвата, тактов опорного сигнала	-	-	-	500
V <sub>SUPnoise</sub>	Низкочастотный шум цепи питания (пик-пик), % от U <sub>VDDCA</sub>	-	-	-	10
V <sub>SUBnoise</sub>	Низкочастотный шум подложки (пик-пик), % от U <sub>VDDCA</sub>	-	-	-	10
Jitter <sub>REF</sub>	Джиттер (пик-пик) опорного тактового сигнала, % от периода	-	-	-	2
t <sub>REFpw</sub>	Длительность положительного/отрицательного импульса опорного сигнала, пс	-	670	-	-

## 36.5 Характеристики Flash памяти

Таблица 36.13. Временные и электрические характеристики Flash памяти

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
t <sub>PROG</sub>	Время программирования одного слова (32 бита), мкс	-	33	-	-
t <sub>PROG_page</sub>	Время программирования одной страницы (8 Кбайт), мс	-	67.584	-	-
t <sub>ERASE</sub>	Время стирания одной страницы (8 Кбайт), мс	-	10	-	20
t <sub>MASS_ERASE</sub>	Время стирания раздела (основного или системного), мс	-	10.1	-	20
I <sub>VDDFL</sub>	Ток потребления по выводу VDDFL, мА	Чтение при U <sub>VDDC</sub> = 1.21 В; U <sub>VDDFL</sub> = 3.63 В; T <sub>J</sub> = 125 °C; LVE = 0	-	0.4	-
		Чтение при U <sub>VDDC</sub> = 0.99 В; U <sub>VDDFL</sub> = 3.63 В; T <sub>J</sub> = 125 °C; LVE = 1	-	0.35	-
		Программирование	-	6	-
		Стирание страницы (ERASE)	-	4	-

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
		Стирание раздела (MASS ERASE)	-	4	-
I <sub>VDDC</sub>	Ток потребления по выводу VDDC, мА	Чтение при U <sub>VDDC</sub> = 1.21 В; U <sub>VDDFL</sub> = 3.63 В; T <sub>J</sub> = 125 °C; LVE = 0	-	4.663	-
		Чтение при U <sub>VDDC</sub> = 0.99 В; U <sub>VDDFL</sub> = 3.63 В; T <sub>J</sub> = 125 °C; LVE = 1	-	5.325	-
		Программирование	-	0.5	-
		Стирание страницы (ERASE)	-	0.5	-
		Стирание раздела (MASS ERASE)	-	0.5	-
I <sub>VDDFLidle</sub>	Ток потребления по выводу VDDFL в неактивном режиме, мкА	Режим ожидания	-	259	-
		Спящий режим (SLM)	-	4.63	-
		Режим пониженного потребления (PDM)	-	4.63	-
I <sub>VDDCidle</sub>	Ток потребления по выводу VDDC в неактивном режиме, мкА	Режим ожидания	-	64	-
		Спящий режим (SLM)	-	64	-
		Режим пониженного потребления (PDM)	-	0	-

Таблица 36.14. Характеристики надежности Flash памяти

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
N <sub>END</sub>	Количество циклов программирования	-	10000	-	-
t <sub>RET</sub>	Время хранения данных, лет	T = 125 °C	10	-	-

## 36.6 Характеристики OTP памяти

Таблица 36.15. Временные и электрические характеристики OTP памяти

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
T <sub>READ</sub>	Температура чтения, °C	-	-40	-	125
T <sub>PROG</sub>	Температура программирования, °C	-	-40	-	70
I <sub>VPP_PROG</sub>	Ток потребления по линии VPP при программировании одного бита, мкА	-	-	250	-
I <sub>VQQ_PROG</sub>	Ток потребления по линии VQQ при программировании одного бита, мкА	-	-	5	-
I <sub>VRR</sub>	Ток потребления по линии VRR при чтении одного бита, мкА/МГц	-	-	1.7	-
I <sub>VDDC</sub>	Ток потребления по выводу VDDC при чтении одного бита, мкА/МГц	-	-	0.25	-

Таблица 36.16. Характеристики надежности ОТР памяти

Обозначение параметра	Наименование параметра, единица измерения	Условия	Мин.	Ном.	Макс.
$t_{RET}$	Время хранения данных, лет	$T = 125\text{ }^{\circ}\text{C}$	10	-	-

## 36.7 Подключение резонатора HFE

На Рисунок 36.1 представлена схема подключения внешних компонент резонатора HFE

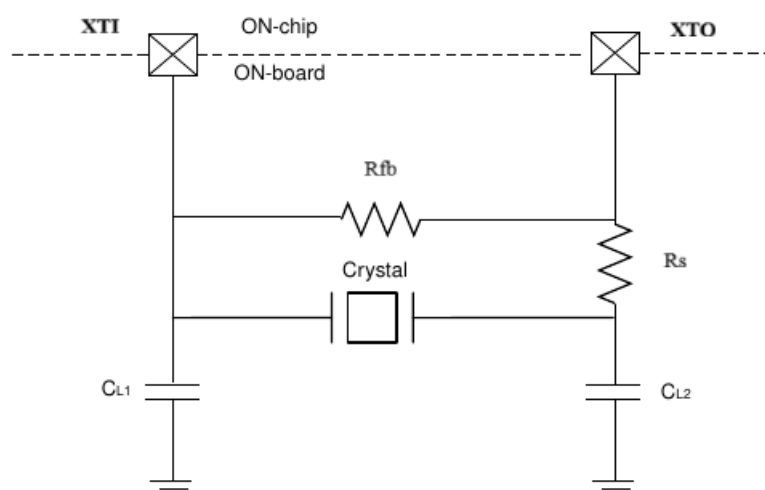


Рисунок 36.1 Схема подключения внешних компонент резонатора

Выбор диапазона частоты осциллятора и режим работы HFE осуществляется с помощью регистра **CLKCTR\_CFG** (см. раздел 2.12 Тактирование)

В Таблица 36.17 представлены номиналы внешних компонент и соответствующих им целевых частот HFE

Таблица 36.17. Номиналы внешних компонент схемы резонатора HFE

	1 МГц – 4 МГц, 1 МГц	1 МГц – 4 МГц, 4 МГц	4 МГц – 12 МГц, 4 МГц	4 МГц – 12 МГц, 12 МГц	12 МГц – 24 МГц, 12 МГц	12 МГц – 24 МГц, 24 МГц	24 МГц – 48 МГц, 24 МГц	24 МГц – 48 МГц, 48 МГц
$R_{fb}$ , Ом	500K	500K	500K	500K	500K	500K	100K	100K
$R_s$ , Ом	0	0	0	0	0	0	0	0
$CL1/CL2$ , Фарад	32п	26п	32п	20п	20п	20п	46п	16п

## **37. ИСТОРИЯ ИЗМЕНЕНИЙ**

### **37.1 Версия от 02.12.2025**

Уточнена схема формирования тактовых сигналов K1892ВГ1Я

### **37.2 Версия от 04.12.2025**

Добавлен раздел «36.1 Подключение резонатора HFE»

Обновлена глава «1. ВВЕДЕНИЕ»

### **37.3 Версия от 09.12.2025**

Обновлена глава «18 QSPI: ИНТЕРФЕЙС ПАМЯТИ QUAD SPI»

### **37.4 Версия от 10.12.2025**

Удалены ссылки на документацию отсутствующую в общем доступе. Обновлено главы:

- «16. VTU: многофункциональные блоки таймеров»
- «21. SDMMC: интерфейс карт Secure Digital Input/Output и Multi Media»
- «27. CRYPTO: Крипто-акселератор CryptoCell»

### **37.5 Версия от 11.12.2025**

Обновлено описание главы «22. USB: Universal Serial Bus 2.0 OTG»

### **37.6 Версия от 03.02.2026**

Обновлено описание главы «12. SPI: контроллер SPI»

- Скорректировано описание регистра CTRLR0

### **37.7 Версия от 18.02.2026**

Обновлено описание главы «2. Системная Организация Микросхемы»

- Исправлено описание регистра PWRCTR\_TRIM
- Скорректировано описание каналов блока JTM

Обновлено описание главы «12. SPI: контроллер SPI»

- Скорректировано описание поля DCOL регистра SR
- Скорректировано описание раздела «12.1 Основные особенности»
- Скорректировано описание поля SSTE регистра CTRLR0
- Скорректировано описание полей SPI\_FRF, SCPOL, SLV\_OE регистра CTRLR0
- Скорректированы рисунки 12.1-12.6

Обновлено описание главы «13. I2C: контроллер I2C»

- Исправлены изображения 13.2 и 13.3 и их названия
- Исправлены ошибки в нумерации изображений
- Скорректировано название регистра IC\_HS\_MADDR
- Дополнено описание поля RX\_DONE регистра IC\_RAW\_INTR\_STAT

Обновлено описание главы «14. I2S: контроллер интерфейса I2S»

- Скорректирована информация в разделе «14.3. Основные особенности»
- Скорректировано описание регистра TCRx