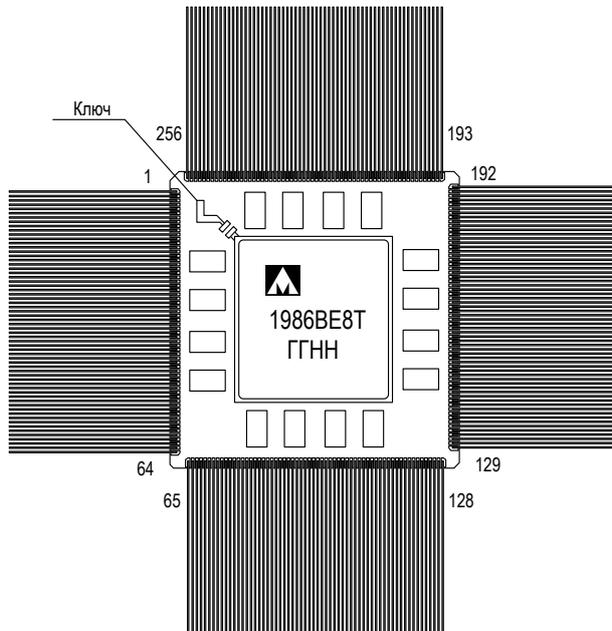




**Микросхема 32-разрядного микроконтроллера
для аппаратуры специального назначения
1986BE8T, K1986BE8T, K1986BE8TK, K1986BE8H4
1986BE81T, K1986BE81T, K1986BE81TK, K1986BE81H4**



ГГ – год выпуска
НН – неделя выпуска

Основные параметры микросхемы

- Напряжение источника питания 3,0 – 5,5 В;
- 32 разрядная RISC архитектура ARM Cortex-M4F;
- Однократно программируемая встроенная память программ 128 Кбайт (1986BE8);
- Встроенная память программ типа CO3U 128 Кбайт (1986BE81);
- Встроенная память данных 32 Кбайт;
- Температурный диапазон:

Обозначение	Диапазон
1986BE8T	минус 60 – 125 °С
K1986BE8T	минус 60 – 125 °С
K1986BE8TK	0 – 70 °С
1986BE81T	минус 60 – 125 °С
K1986BE81T	минус 60 – 125 °С
K1986BE81TK	0 – 70 °С

Тип корпуса:

- 256-ти выводной металлокерамический корпус 4244.256-3;
- микросхемы K1986BE8H4, K1986BE81H4 поставляются в бескорпусном исполнении.

Общее описание и области применения микросхемы

Микросхемы интегральные 1986BE8T, 1986BE81T (далее – микросхемы) представляют собой специализированный микроконтроллер с измерительно-информационными каналами для чувствительных элементов (датчиков). Микросхемы предназначены для использования в системах специального назначения и двойного применения для измерения, контроля, управления и диагностики.

Основные характеристики

Ядро:

- ядро ARM Cortex-M4F;
- максимальная тактовая частота до 100 МГц;
- производительность до 2,1 CoreMark/МГц;
- поддержка на уровне ядра операций ЦОС;
- аппаратное выполнение операций с плавающей точкой;
- блок аппаратной защиты регионов памяти;
- блок отладки с поддержкой трассы инструкций ETM и интерфейсами JTAG и SWD.

Питание:

- основное питание от 3,0 до 5,5 В;
- встроенные регуляторы для питания ядра;
- батарейный домен, с автоматическим переключением на питание от батарейки;
- аппаратный детектор снижения и превышения допустимого уровня питания;
- изолированные от помех питания для АЦП и ЦАП.

Тактовые частоты:

- встроенные высоконадежные RC-генераторы HSI (~8 МГц) и LSI (~40 кГц);
- внешние генераторы HSE0 и HSE1 от 1 до 30 МГц в режиме резонатора;
- внешний часовой генератор LSE в батарейном домене;
- 3 блока PLL умножения тактовых частот до 150 МГц;
- блоки аппаратной защиты от снижения/увеличения частоты тактирования.

Память:

- 32 Кбайт СОЗУ памяти данных с ECC (SEC-DED);
- 128 Кбайт однократно программируемой памяти программ с ECC (SEC-DED) для микросхем 1986BE8;
- 128 Кбайт памяти программ типа СОЗУ с ECC (SEC-DED) для микросхем 1986BE81;
- контроллер внешней системной шины с последовательной/параллельной организацией с ECC (SEC-DED).

Периферия:

- контроллер EthernetMAC 10/100 Мбит/с;
- встроенный приемопередатчик EthernetPHY 10 Мбит/с;
- контроллер SpaceWire;
- встроенный приемопередатчик SpaceWirePHY до 100 Мбит/с;
- контроллер ARINC 429 (8 приемников, 4 передатчика);
- два контроллера МКПД в режимах КШ, ОУ, Монитор;
- стандартная периферия: 1xCAN, 1xSSP, 2xUART, 1xDMA, 4xTimer;
- часы реального времени RTC;
- сторожевой таймер WDG;
- два АЦП с 16 внешними каналами с разрешающей способностью 12 бит;
- два ЦАП с разрешающей способностью 12 бит;
- до 160 выводов портов общего назначения.

Стойкость:

- технология 0,18 мкм кремний на изоляторе (КНИ);
- методы аппаратной защиты от одиночных сбоев;
- различные механизмы самодиагностики и защиты от сбоев и отказов.

Оглавление

1	Описание выводов	8
2	Условно-графическое обозначение	21
3	Структурные блок-схемы.....	22
4	Расположение выводов микроконтроллера	24
5	Указания по применению и эксплуатации.....	29
6	Архитектура микроконтроллера	30
6.1	Питание.....	30
6.1.1	Домены питания.....	30
6.1.2	Сброс при включении питания.....	31
6.1.3	Схема защиты от перенапряжения POVR.....	31
6.1.4	Перекрытие рабочих диапазонов питания схемами POR и POVR	31
6.1.5	Схема переключения основного и батарейного питаний (Ucc и BUcc).....	32
6.1.6	Блок монитора основного и батарейного питаний (Ucc и BUcc)	32
6.1.7	Управление питанием доменов DUcc.....	32
6.1.8	Блок монитора тока потребления доменов DUcc	33
6.2	Тактовые частоты.....	33
6.2.1	Встроенный низкоскоростной генератор LSI.....	33
6.2.2	Встроенный высокоскоростной генератор HSI.....	33
6.2.3	Внешний высокоскоростной генератор HSE0 и HSE1	33
6.2.4	Внешний низкоскоростной часовой генератор LSE	33
6.2.5	Блок умножения тактовой частоты PLL0, PLL1 и PLL2.....	34
6.2.6	Монитор частоты внешних генераторов и PLL (в блоке CLKNTR)	34
6.3	Сигналы сброса	35
6.3.1	Сигналы внешнего сброса RSTn0 и RSTn1	35
6.3.2	Сигнал сброса по питанию UccRESET	35
6.3.3	Сигнал сброса по питанию DUccRESET.....	35
6.3.4	Сигнал программного запроса сброса SYSRESETnREQ	36
6.3.5	Сигналы аварийного сброса FT_RESET.....	36
6.3.6	Сигналы сброса сторожевого таймера IWDG_RESET	36
6.3.7	Сигнал сброса по перенапряжению POVR.....	36
6.3.8	Сигнал сброса отладочного интерфейса TRSTn	36
6.4	Режимы работы микроконтроллера	36
6.4.1	Режим OTP+JA.....	38
6.4.2	Режим OTP+JB.....	38
6.4.3	Режим EXTBUS_8_ECC+JA.....	38
6.4.4	Режим EXTBUS_8_ECC+JB.....	39
6.4.5	Режим EXTBUS_CFG+JA	39
6.4.6	Режим EXTBUS_CFG+JB	40
6.4.7	Режим SPI0+JB	40
6.4.8	Режим SPI1+JA	41
6.4.9	Режим SPI2+JB	42
6.4.10	Режим SPI3+JA	43
6.4.11	Режим UART0+JA	43
6.4.12	Режим UART0+JB	47
6.4.13	Режим TEST_MODE+JB	47
6.4.14	Режим WAIT_BOOT_JA	47
6.4.15	Режим WAIT_BOOT_JB	48
6.4.16	Статус загрузчика	48
6.5	Организация памяти.....	49
6.5.1	Распределение областей памяти	52
6.5.2	Распределение областей памяти для регистров периферии	53
6.5.3	Распределение областей памяти для регистров периферии ядра.....	55
6.6	Помехозащищенное кодирование.....	55
6.6.1	ЕСС кодирование для ПЗУ.....	55

6.6.2	ЕСС кодирование для OTP памяти программ (1986VE8)	56
6.6.1	ЕСС кодирование для ОЗУ памяти программ (1986VE81)	57
6.6.2	ЕСС кодирование для периферии	58
6.6.3	Н-матрица ЕСС по коду Хемминга (72, 64)	58
6.6.4	Н-матрица ЕСС по коду Хемминга (8, 4)	58
6.6.5	Функция вычисления ЕСС для записи в память	58
6.7	Система команд процессорного ядра	59
6.8	Интерфейс отладки	59
6.9	Тестовые режимы работы	60
6.9.1	Режим тестирования ОЗУ	61
6.9.2	Режим тестирования Масочного ПЗУ (BOOTROM)	62
6.9.3	Режим тестирования OTP (для микросхем 1986VE8Т)	63
6.9.4	Режим тестирования RAM (для микросхем 1986VE81Т)	64
6.9.5	Режим тестирования RAM_CAN	65
6.9.6	Режим тестирования ETH_MEM	66
6.9.7	Режим тестирования RAM_MIL	67
6.9.8	Режим тестирования RAM_ARC	68
6.9.9	Режим тестирования генераторов и PLL	69
6.9.10	Режим тестирования АЦП	70
6.9.11	Режим тестирования DAC	70
6.9.12	Режим тестирования приемопередатчиков SPW	71
6.9.13	Режим тестирования ЕТНРНУ	71
6.9.14	Режим тестирования РМУ (VoltageDetect)	72
6.9.15	Режим тестирования ВКР	73
7	Программная модель микроконтроллера	74
7.1	Идентификация микросхемы	74
7.1.1	Описание полей идентификации	74
7.2	Контроллер тактовых частот (CLKCNTR)	75
7.2.1	Формирование и управление тактовыми сигналами	75
7.2.2	Схема контроля тактовых частот	77
7.2.3	Описание регистров	81
7.3	Контроллер батарейного домена (BKPCNTR)	116
7.3.1	Часы реального времени	117
7.3.2	Описание регистров	117
7.4	PWRCNTR - Контроллер детектора напряжения питания	132
7.4.1	Описание регистров	132
7.5	Контроллер обработки событий отказов, сбоев и ошибок (FTCNTR)	139
7.5.1	Описание регистров	139
7.6	Контроллер сторожевых таймеров (WDCNTR)	174
7.6.1	Описание регистров	175
7.7	Контроллер внешней шины (EXTBUSCNTR)	178
7.7.1	Организация доступа к внешней шине EXTBUS	178
7.7.2	Описание регистров	192
7.8	Контроллер кэш-памяти (CACHCNTR)	200
7.8.1	Описание регистров	201
7.9	Контроллер OTP памяти программ для микросхем 1986VE8Т (OTPCNTR)	207
7.9.1	Методика программирования OTP	207
7.9.2	Параметры контроллера OTP	214
7.9.3	Описание регистров	215
7.10	Контроллер ОЗУ памяти программ для микросхем 1986VE81Т (OTPCNTR)	223
7.10.1	Описание регистров	225
7.11	Контроллер ПЗУ (ROMCNTR)	233
7.11.1	Описание регистров	233
7.12	Контроллер ОЗУ (RAMCNTR)	237
7.12.1	Описание регистров	237
7.13	Контроллер DMA (DMACNTR)	241

7.13.1	Основные характеристики контроллера DMA	241
7.13.2	Термины и определения.....	241
7.13.3	Функциональное описание	242
7.13.4	Управление DMA.....	244
7.13.5	Структура управляющих данных канала	263
7.13.6	Описание регистров контроллера DMA.....	272
7.14	Контроллер портов ввода-вывода (PORTx).....	293
7.14.1	Описание регистров.....	294
7.15	Контроллер Ethernet (EthernetMAC)	322
7.15.1	Передача пакета	323
7.15.2	Принцип работы передатчика	324
7.15.3	Прием пакета	324
7.15.4	Принцип работы приемника	324
7.15.5	Линейный режим работы буферов	325
7.15.6	Автоматический режим работы буферов	325
7.15.7	Режим FIFO работы буферов.....	325
7.15.8	События приемника и передатчика	325
7.15.9	Прерывания.....	326
7.15.10	Режим детерминированного времени доставки.....	326
7.15.11	Режим КЗ.....	326
7.15.12	Режимы отладки	326
7.15.13	Блок PHY	327
7.15.14	Описание регистров.....	327
7.16	Контроллер SpaceWire (SpaceWire)	342
7.16.1	Кодирование данных	342
7.16.2	Скорость интерфейса	342
7.16.3	Представление символов.....	342
7.16.4	Диаграмма состояний	343
7.16.5	Инициализация	346
7.16.6	Run (Рабочий режим).....	346
7.16.7	Передача сообщений.....	347
7.16.8	Прием сообщений	347
7.16.9	Передача и прием маркера времени	347
7.16.10	Описание регистров.....	348
7.17	Контроллер таймеров общего назначения (TIMER32x)	361
7.17.1	Функционирование.....	361
7.17.2	Инициализация таймера	362
7.17.3	Режим таймера	363
7.17.4	Режимы счета	363
7.17.5	Источник событий для счета	366
7.17.6	Режим захвата	374
7.17.7	Режим ШИМ	375
7.17.8	Примеры.....	378
7.17.9	Описание регистров.....	382
7.18	Контроллер CAN.....	398
7.18.1	Режимы работы.....	398
7.18.2	Типы пакетов сообщений	400
7.18.3	Структура пакета данных (DataFrame)	400
7.18.4	Структура пакета удаленного запроса данных (Remoteframe).....	403
7.18.5	Арбитраж на шине	403
7.18.6	Инициализация	405
7.18.7	Передача сообщений.....	405
7.18.8	Передача сообщений по RemoteTransmitRequest (RTR).....	405
7.18.9	Прием сообщений.....	405
7.18.10	Автоматическая фильтрация принимаемых сообщений	406
7.18.11	Задание скорости передачи и момента сэмплирования	406

7.18.12	Перезапись принятых сообщений.....	407
7.18.13	Синхронизация.....	407
7.18.14	Обработка ошибок.....	408
7.18.15	Прерывания.....	411
7.18.16	Описание регистров.....	413
7.19	Контроллер SSP.....	428
7.19.1	Программируемые параметры.....	428
7.19.2	Общий обзор модуля SSP.....	430
7.19.3	Интерфейс прямого доступа к памяти.....	444
7.19.4	Прерывания.....	445
7.19.5	Программное управление модулем.....	446
7.19.6	Описание регистров.....	446
7.20	Контроллер UART (UARTx).....	454
7.20.1	Основные сведения.....	454
7.20.2	Основные характеристики модуля UART.....	454
7.20.3	Программируемые параметры.....	454
7.20.4	Описание функционирования блока UART.....	456
7.20.5	Интерфейс прямого доступа к памяти.....	458
7.20.6	Блок и регистры синхронизации.....	458
7.20.7	Описание работы UART.....	458
7.20.8	Интерфейс прямого доступа к памяти.....	461
7.20.9	Прерывания.....	462
7.20.10	Программное управление модулем.....	464
7.20.11	Описание регистров.....	465
7.21	Контроллер интерфейса ARINC (ARINCx).....	478
7.21.1	Формат слова.....	479
7.21.2	Структурная схема канала приёма.....	480
7.21.3	Структурная схема канала передачи.....	481
7.21.4	Описание регистров приемника.....	483
7.21.5	Описание регистров передатчика.....	494
7.22	Контроллер интерфейса МКПД (MILx).....	501
7.22.1	Режимы работы.....	502
7.22.2	Форматы сообщений.....	502
7.22.3	Формат слов.....	504
7.22.4	Структурная схема в режиме КШ.....	506
7.22.5	Структурная схема в режиме ОУ.....	506
7.22.6	Структурная схема в режиме М.....	507
7.22.7	Инициализация.....	507
7.22.8	Приём и передача в режиме ОУ.....	508
7.22.9	Прием и передача в режиме КШ.....	509
7.22.10	Прерывания.....	510
7.22.11	Описание регистров.....	511
7.23	Контроллер АЦП (ADCx).....	525
7.23.1	Преобразуемые сигналы.....	526
7.23.2	Источники опорного напряжения.....	527
7.23.3	Синхронизация АЦП.....	527
7.23.4	Запуск преобразования.....	527
7.23.5	Последовательный опрос каналов АЦП.....	528
7.23.6	Получение результатов преобразования.....	528
7.23.7	Обработка результатов с помощью DMA.....	529
7.23.8	Преобразование с контролем границ.....	529
7.23.9	Автоматическая калибровка АЦП.....	529
7.23.10	Диаграмма работы АЦП.....	529
7.23.11	Схема входного каскада АЦП.....	530
7.23.12	Описание регистров.....	531
7.24	Контроллер ЦАП (DACx).....	545

7.24.1	Источники опорного напряжения	545
7.24.2	Синхронизация ЦАП	545
7.24.3	Очередь данных.....	546
7.24.4	Описание регистров.....	547
7.25	Векторы прерываний и регистры управления ядра.....	554
7.25.1	Распределение векторов прерываний.....	554
7.25.2	Описание регистров управления ядра (SCS).....	557
7.26	Контроллер блока защиты памяти (MPU)	559
7.27	Контроллер блока вычисления с плавающей запятой (FPU).....	559
7.28	Контроллер порта трассировки (TPIU)	559
7.29	Контроллер блока расширенной трассировки (ETM)	559
8	Типовая схема включения.....	560
9	Основные зависимости.....	561
10	Предельно-допустимые характеристики микросхемы	564
11	Электрические параметры микросхемы	566
12	Справочные данные	568
13	Габаритный чертеж микросхемы	571
14	Информация для заказа	576

1 Описание выводов

Таблица 1 – Описание выводов

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
1	1	RSTn0	I 5V	Вход нулевого сигнала сброса 0 – запрос сброса 1 – рабочий режим
2	3	FT_RSTn	O 5V	Выход с открытым стоком сигнала запроса сброса от блока 0 – сигнал запроса сброса
3	2	RSTn1	I 5V	Вход первого сигнала сброса 0 – запрос сброса 1 – рабочий режим Сброс микросхемы выполняется при условии запроса сброса по обоим выводам RSTn0 и RSTn1
4	5	FT_WRNn	O 5V	Выход с открытым стоком сигнала предупреждения о сбое от блока контроллера сбоев 0 – предупреждение о сбое
5	4	PE[31]	IO 5V	Вход-выход 31 порта E
6	7	PE[30]	IO 5V	Вход-выход 30 порта E
7	6	PE[29]	IO 5V	Вход-выход 29 порта E
8	9	PE[28]	IO 5V	Вход-выход 28 порта E
9	8	PE[27]	IO 5V	Вход-выход 27 порта E
10	11	PE[26]	IO 5V	Вход-выход 26 порта E
11	10	PE[25]	IO 5V	Вход-выход 25 порта E
12	13	PE[24]	IO 5V	Вход-выход 24 порта E
13	12	PE[23]	IO 5V	Вход-выход 23 порта E
14	15	PE[22]	IO 5V	Вход-выход 22 порта E
15	14	PE[21]	IO 5V	Вход-выход 21 порта E
16	17	PE[20]	IO 5V	Вход-выход 20 порта E
17	16	PE[19]	IO 5V	Вход-выход 19 порта E
18	19	PE[18]	IO 5V	Вход-выход 18 порта E
19	18	PE[17]	IO 5V	Вход-выход 17 порта E
20	21	U _{cc}	PWR	Питание площадок ввода-вывода
21	20			
22	22	GND	PWR	Общий
23	23			
24	24	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
25	25			
26	26	PE[16]	IO 5V	Вход-выход 16 порта E
27	27	PE[15]	IO 5V	Вход-выход 15 порта E
28	28	PE[14]	IO 5V	Вход-выход 14 порта E
29	29	PE[13]	IO 5V	Вход-выход 13 порта E
30	30	PE[12]	IO 5V	Вход-выход 12 порта E
31	31	PE[11]	IO 5V	Вход-выход 11 порта E
32	–	NC	–	Не подключен
33	32	PE[10]	IO 5V	Вход-выход 10 порта E
34	33	PE[9]	IO 5V	Вход-выход 9 порта E
35	34	PE[8]	IO 5V	Вход-выход 8 порта E
36	–	NC	–	Не подключен
37	35	PE[7]	IO 5V	Вход-выход 7 порта E
38	36	PE[6]	IO 5V	Вход-выход 6 порта E
39	37	PE[5]	IO 5V	Вход-выход 5 порта E
40	38	PE[4]	IO 5V	Вход-выход 4 порта E

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
41	39	PE[3]	IO 5V	Вход-выход 3 порта E
42	40	PE[2]	IO 5V	Вход-выход 2 порта E
43	41	PE[1]	IO 5V	Вход-выход 1 порта E
44	42	PE[0]	IO 5V	Вход-выход 0 порта E
45	43	PD[31]	IO 5V	Вход-выход 31 порта D
46	44	PD[30]	IO 5V	Вход-выход 30 порта D
47	46	PD[29]	IO 5V	Вход-выход 29 порта D
48	45	PD[28]	IO 5V	Вход-выход 28 порта D
49	48	PD[27]	IO 5V	Вход-выход 27 порта D
50	47	PD[26]	IO 5V	Вход-выход 26 порта D
51	50	PD[25]	IO 5V	Вход-выход 25 порта D
52	49	PD[24]	IO 5V	Вход-выход 24 порта D
53	52	PD[23]	IO 5V	Вход-выход 23 порта D
54	51	PD[22]	IO 5V	Вход-выход 22 порта D
55	54	PD[21]	IO 5V	Вход-выход 21 порта D
56	53	PD[20]	IO 5V	Вход-выход 20 порта D
57	56	PD[19]	IO 5V	Вход-выход 19 порта D
58	55	PD[18]	IO 5V	Вход-выход 18 порта D
59	58	PD[17]	IO 5V	Вход-выход 17 порта D
60	57	PD[16]	IO 5V	Вход-выход 16 порта D
61	60	SHDN	I	Вход выключения встроенных регуляторов напряжения 0 – регуляторы включены 1 – регуляторы выключены
62	59	PD[15]	IO 5V	Вход-выход 15 порта D
63	62	DU _{cc} _B	PWR	Питание 1,8 В цифровой части батарейного домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
64	61	BDU _{cc}	PWR	Вывод подключения конденсатора к питанию LDO регулятора батарейного домена
65	63	BU _{cc}	PWR	Батарейное питание
66	64	LSE_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору LSE
67	65	LSE_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору LSE
68	66	STANDBY	O	Выход состояния встроенного регулятора 0 – регулятор включен 1 – регулятор программно выключен
69	67	WAKEUP	I 5V	Вход сигнала запроса включения регулятора 1 – нет запроса включения 0 – есть запрос включения встроенного регулятора
70	68, 69	U _{cc}	PWR	Питание площадок ввода-вывода
71	70	PD[14]	IO 5V	Вход-выход 14 порта D
72	71	PD[13]	IO 5V	Вход-выход 13 порта D
73	72	PD[12]	IO 5V	Вход-выход 12 порта D
74	73	PD[11]	IO 5V	Вход-выход 11 порта D
75	74, 76	GND	PWR	Общий
76	77	PD[10]	IO 5V	Вход-выход 10 порта D
77	78	PD[9]	IO 5V	Вход-выход 9 порта D
78	79	PD[8]	IO 5V	Вход-выход 8 порта D
79	80	PD[7]	IO 5V	Вход-выход 7 порта D
80	81	PD[6]	IO 5V	Вход-выход 6 порта D

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
81	82, 84	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
82	85	PD[5]	IO 5V	Вход-выход 5 порта D
83	86	PD[4]	IO 5V	Вход-выход 4 порта D
84	87	PD[3]	IO 5V	Вход-выход 3 порта D
85	88	PD[2]	IO 5V	Вход-выход 2 порта D
86	89	PD[1]	IO 5V	Вход-выход 1 порта D
87	90, 91	DU _{cc2}	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
88	92	PD[0]	IO 5V	Вход-выход 0 порта D
89	93	PC[31]	IO 5V	Вход-выход 31 порта C
90	94	PC[30]	IO 5V	Вход-выход 30 порта C
91	95	DAC_IREF	A	Контрольный вывод проверки источника опорного тока
92	96, 97	DU _{cc0}	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
93	98	DAC_EXTR0	A	Выводы для подключения внешних токозадающих резисторов для формирования опорных токов R = 120 кОм ± 5 %
94	99	DAC_EXTR1	A	
95	100	PC[29]	AIO	Вход-выход 29 порта C. Совмещен с сигналом DAC1_REF+
96	101	PC[28]	AIO	Вход-выход 28 порта C. Совмещен с сигналом DAC1_OUT
97	102, 103	GND	PWR	Общий
98	104	PC[27]	AIO	Вход-выход 27 порта C. Совмещен с сигналом DAC1_REF-
99	105	PC[26]	AIO	Вход-выход 26 порта C. Совмещен с сигналом DAC0_REF-
100	106	PC[25]	AIO	Вход-выход 25 порта C. Совмещен с сигналом DAC0_OUT
101	107	PC[24]	AIO	Вход-выход 24 порта C. Совмещен с сигналом DAC0_REF+
102	108	PC[23]	AIO	Вход-выход 23 порта C. Совмещен с сигналом ADC17_REF1-
103	109, 110	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
104	111	PC[22]	AIO	Вход-выход 22 порта C. Совмещен с сигналом ADC16_REF1+
105	112	PC[21]	AIO	Вход-выход 21 порта C. Совмещен с сигналом ADC15
106	113	PC[20]	AIO	Вход-выход 20 порта C. Совмещен с сигналом ADC14
107	114	PC[19]	AIO	Вход-выход 19 порта C. Совмещен с сигналом ADC13
108	115, 116	DAC_U _{cc}	PWR	Питание аналоговых блоков ЦАП
109	117	PC[18]	AIO	Вход-выход 18 порта C. Совмещен с сигналом ADC12
110	118	PC[17]	AIO	Вход-выход 17 порта C. Совмещен с сигналом ADC11

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
111	119	PC[16]	AIO	Вход-выход 16 порта С. Совмещен с сигналом ADC10
112	120	PC[15]	AIO	Вход-выход 15 порта С. Совмещен с сигналом ADC9
113	121, 122	DAC_GND	PWR	Общий аналоговых блоков ЦАП
114	123	PC[14]	AIO	Вход-выход 14 порта С. Совмещен с сигналом ADC8
115	124	PC[13]	AIO	Вход-выход 13 порта С. Совмещен с сигналом ADC7
116	125	PC[12]	AIO	Вход-выход 12 порта С. Совмещен с сигналом ADC6
117	126	PC[11]	AIO	Вход-выход 11 порта С. Совмещен с сигналом ADC5
118	127	ADC_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока $R = 120 \text{ кОм} \pm 5 \%$
119	128, 129	ADC_GND	PWR	Общий аналоговых блоков АЦП
120	130	PC[10]	AIO	Вход-выход 10 порта С. Совмещен с сигналом ADC4
121	131	PC[9]	AIO	Вход-выход 9 порта С. Совмещен с сигналом ADC3
122	132	PC[8]	AIO	Вход-выход 8 порта С. Совмещен с сигналом ADC2
123	133	PC[7]	AIO	Вход-выход 7 порта С. Совмещен с сигналом ADC1_REF0-
124	134, 135	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
125	136			
126	137, 139	ADC_U _{cc}	PWR	Питание аналоговых блоков АЦП
127	138	GND	PWR	Общий
128	140	PC[6]	AIO	Вход-выход 6 порта С. Совмещен с сигналом ADC0_REF0+
129	141	DU _{cc1}	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
130	143			
131	142	HV	PWR	Высоковольтный вход напряжения программирования однократно программируемой памяти программ. В рабочем режиме подключить к U _{cc}
132	145			
133	144	PC[5]	IO 5V	Вход-выход 5 порта С
134	147	PC[4]	IO 5V	Вход-выход 4 порта С
135	146	PC[3]	IO 5V	Вход-выход 3 порта С
136	–	NC	–	Не подключен
137	148	PC[2]	IO 5V	Вход-выход 2 порта С
138	150	PC[1]	IO 5V	Вход-выход 1 порта С
139	149	PC[0]	IO 5V	Вход-выход 0 порта С
140	152	PB[31]	IO 5V	Вход-выход 31 порта В
141	151	PB[30]	IO 5V	Вход-выход 30 порта В
142	–	NC	–	Не подключен
143	153	PB[29]	IO 5V	Вход-выход 29 порта В
144	155	PB[28]	IO 5V	Вход-выход 28 порта В
145	154	PB[27]	IO 5V	Вход-выход 27 порта В
146	157	PB[26]	IO 5V	Вход-выход 26 порта В
147	156	PB[25]	IO 5V	Вход-выход 25 порта В
148	159	PB[24]	IO 5V	Вход-выход 24 порта В
149	158	PB[23]	IO 5V	Вход-выход 23 порта В

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
150	160	PВ[22]	IO 5V	Вход-выход 22 порта В
151	161	PВ[21]	IO 5V	Вход-выход 21 порта В
152	162	PВ[20]	IO 5V	Вход-выход 20 порта В
153	163	U _{cc}	PWR	Питание площадок ввода-вывода
154	164	PВ[19]	IO 5V	Вход-выход 19 порта В
155	165	U _{cc}	PWR	Питание площадок ввода-вывода
156	–	NC	–	Не подключен
157	166	GND	PWR	Общий
158	–	NC	–	Не подключен
159	167	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
160	168	GND	PWR	Общий
161	169	PВ[18]	IO 5V	Вход-выход 18 порта В
162	170	U _{cc1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{cc0} , DU _{cc1} , DU _{cc2} , DU _{cc_PLL}
163	171	PВ[17]	IO 5V	Вход-выход 17 порта В
164	172	PВ[16]	IO 5V	Вход-выход 16 порта В
165	173	PВ[15]	IO 5V	Вход-выход 15 порта В
166	174	PВ[14]	IO 5V	Вход-выход 14 порта В
167	175	PВ[13]	IO 5V	Вход-выход 13 порта В
168	–	NC	–	Не подключен
169	176	PВ[12]	IO 5V	Вход-выход 12 порта В
170	177	PВ[11]	IO 5V	Вход-выход 11 порта В
171	178	PВ[10]	IO 5V	Вход-выход 10 порта В
172	179	PВ[9]	IO 5V	Вход-выход 9 порта В
173	180	PВ[8]	IO 5V	Вход-выход 8 порта В. При выборе JTAG_V совмещен с JTAGB_SWVTDO
174	181	PВ[7]	IO 5V	Вход-выход 7 порта В. При выборе JTAG_V совмещен с JTAGB_TDI
175	–	NC	–	Не подключен
176	182	PВ[6]	IO 5V	Вход-выход 6 порта В. При выборе JTAG_V совмещен с JTAGB_SWDIOTMS
177	184	PВ[5]	IO 5V	Вход-выход 5 порта В. При выборе JTAG_V совмещен с JTAGB_SWCLKTCK
178	183	PВ[4]	IO 5V	Вход-выход 4 порта В. При выборе JTAG_V совмещен с JTAGB_nTRST
179	186	PВ[3]	IO 5V	Вход-выход 3 порта В
180	185	PВ[2]	IO 5V	Вход-выход 2 порта В
181	188	PВ[1]	IO 5V	Вход-выход 1 порта В
182	187	PВ[0]	IO 5V	Вход-выход 0 порта В
183	190	РА[31]	IO 5V	Вход-выход 31 порта А
184	189	РА[30]	IO 5V	Вход-выход 30 порта А
185	192	РА[29]	IO 5V	Вход-выход 29 порта А
186	191	РА[28]	IO 5V	Вход-выход 28 порта А
187	194	РА[27]	IO 5V	Вход-выход 27 порта А
188	193	РА[26]	IO 5V	Вход-выход 26 порта А
189	196	GND	PWR	Общий
190	195	РА[25]	IO 5V	Вход-выход 25 порта А
191	197	GND	PWR	Общий
192	–	NC	–	Не подключен

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
193	198	Ucc1	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DUcc0, DUcc1, DUcc2, DUcc_PLL
194	199, 200			
195	201	SPW_IREF	A	Контрольный вывод проверки источника опорного тока
196	202	SPW_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока R = 120 кОм ± 5 %
197	203	SPW_Ucc	PWR	Питание приемопередатчика SpaceWire PHY
198	204, 205	GND	PWR	Общий
199	206	SPW_GND	PWR	Общий приемопередатчика SpaceWirePHY
200	207	SPW_RXD+	AI	Сигналы приемопередатчика интерфейса SpaceWire
201	208	SPW_RXD-	AI	
202	209	SPW_RXS+	AI	
203	210	SPW_RXS-	AI	
204	211	SPW_TXS+	AO	
205	212	SPW_TXS-	AO	
206	213	SPW_TXD+	AO	
207	214	SPW_TXD-	AO	
208	215	GND	PWR	Общий
209	216	PA[24]	IO 5V	Вход-выход 24 порта A
210	217	PA[23]	IO 5V	Вход-выход 23 порта A
211	218	PA[22]	IO 5V	Вход-выход 22 порта A
212	219	PA[21]	IO 5V	Вход-выход 21 порта A
213	220, 221	Ucc1	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DUcc0, DUcc1, DUcc2, DUcc_PLL
214	222	PA[20]	IO 5V	Вход-выход 20 порта A
215	223	PA[19]	IO 5V	Вход-выход 19 порта A
216	224	PA[18]	IO 5V	Вход-выход 18 порта A
217	225	PA[17]	IO 5V	Вход-выход 17 порта A
218	226, 227	GND	PWR	Общий
219	228	HSE1_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору HSE1
220	229	HSE1_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору HSE1
221	230	ETH_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока R = 120 кОм ± 5 %
222	231	ETH_IREF	A	Контрольный вывод проверки источника опорного тока
223	232, 233	ETH_GND	PWR	Общий приемопередатчика EthernetPHY
224	234	ETH_TX+	AO	Сигналы приемопередатчика интерфейса Ethernet
225	235	ETH_TX-	AO	
226	236	ETH_RX+	AI	
227	237	ETH_RX-	AI	
228	238, 239	ETH_Ucc	PWR	Питание приемопередатчика Ethernet PHY
229	240	PA[16]	IO 5V	Вход-выход 16 порта A
230	241	PA[15]	IO 5V	Вход-выход 15 порта A
231	242	PA[14]	IO 5V	Вход-выход 14 порта A
232	243	Ucc1	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DUcc0, DUcc1, DUcc2, DUcc_PLL
233	244, 245			
234	246	PA[13]	IO 5V	Вход-выход 13 порта A
235	247	PA[12]	IO 5V	Вход-выход 12 порта A
236	248	PA[11]	IO 5V	Вход-выход 11 порта A
237	249	GND	PWR	Общий

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Назначение и функции вывода
238	250, 251			
239	252	PA[10]	IO 5V	Вход-выход 10 порта А
240	253	PA[9]	IO 5V	Вход-выход 9 порта А
241	254	PA[8]	IO 5V	Вход-выход 8 порта А
242	255	PA[7]	IO 5V	Вход-выход 7 порта А
243	256, 257	DUcc0	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
244	258	PA[6]	IO 5V	Вход-выход 6 порта А
245	259	PA[5]	IO 5V	Вход-выход 5 порта А
246	260	PA[4]	IO 5V	Вход-выход 4 порта А. Совмещен с JTAGA_SWVTD0
247	261	PA[3]	IO 5V	Вход-выход 3 порта А. Совмещен с JTAGA_TDI
248	262	PA[2]	IO 5V	Вход-выход 2 порта А. Совмещен с JTAGA_SWDIOTMS
249	263	DUcc_PLL	PWR	Питание 1,8 В домена PLL при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
250	264	PA[1]	IO 5V	Вход-выход 1 порта А. Совмещен с JTAGA_SWCLKTCK
251	265	PA[0]	IO 5V	Вход-выход 0 порта А. Совмещен с JTAGA_nTRST
252	266	HSE0_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору HSE0
253	267	HSE0_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору HSE0
254	268, 270	Ucc	PWR	Питание площадок ввода-вывода
255	269	JTAG_EN	I	Вход выбора тестового режима 0 – рабочий режим 1 – тестовый режим
256	–	GND(NC)	PWR	Общий подложки кристалла

Примечания:

1. Пользовательский порт А ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и Ethernet MAC, Timer, TRACE.
2. Пользовательский порт В ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и SpaceWire MAC, Ethernet MAC, Timer.
3. Пользовательский порт С ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.
4. Пользовательский порт D ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.
5. Пользовательский порт Е ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.
6. Типы выводов:
 - PWR – вывод питания
 - I – цифровой вход
 - O – цифровой выход
 - IO – цифровой вход-выход
 - A – аналоговый вывод
 - AI – аналоговый вход
 - AO – аналоговый выход
 - AIO – цифровой вход-выход совмещенный с аналоговым выводом
 - 5V – толерантный к 5 В цифровой вход

Таблица 2 – Описание выводов по блокам

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода
Порт А				
251	265	PA[0]	IO 5V	Пользовательский порт А ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и Ethernet MAC, Timer, TRACE При выборе JTAG_A Вывод PA[0] совмещен с JTAGA_nTRST Вывод PA[1] совмещен с JTAGA_SWCLKTCK Вывод PA[2] совмещен с JTAGA_SWDIOTMS Вывод PA[3] совмещен с JTAGA_TDI Вывод PA[4] совмещен с JTAGA_SWVTDO
250	264	PA[1]	IO 5V	
248	262	PA[2]	IO 5V	
247	261	PA[3]	IO 5V	
246	260	PA[4]	IO 5V	
245	259	PA[5]	IO 5V	
244	258	PA[6]	IO 5V	
242	255	PA[7]	IO 5V	
241	254	PA[8]	IO 5V	
240	253	PA[9]	IO 5V	
239	252	PA[10]	IO 5V	
236	248	PA[11]	IO 5V	
235	247	PA[12]	IO 5V	
234	246	PA[13]	IO 5V	
231	242	PA[14]	IO 5V	
230	241	PA[15]	IO 5V	
229	240	PA[16]	IO 5V	
217	225	PA[17]	IO 5V	
216	224	PA[18]	IO 5V	
215	223	PA[19]	IO 5V	
214	222	PA[20]	IO 5V	
212	219	PA[21]	IO 5V	
211	218	PA[22]	IO 5V	
210	217	PA[23]	IO 5V	
209	216	PA[24]	IO 5V	
190	195	PA[25]	IO 5V	
188	193	PA[26]	IO 5V	
187	194	PA[27]	IO 5V	
186	191	PA[28]	IO 5V	
185	192	PA[29]	IO 5V	
184	189	PA[30]	IO 5V	
183	190	PA[31]	IO 5V	
Порт В				
182	187	PB[0]	IO 5V	Пользовательский порт В ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и SpaceWire MAC, Ethernet MAC, Timer При выборе JTAG_B Вывод PB[4] совмещен с JTAGB_nTRST Вывод PB[5] совмещен с JTAGB_SWCLKTCK Вывод PB[6] совмещен с JTAGB_SWDIOTMS Вывод PB[7] совмещен с JTAGB_TDI Вывод PB[8] совмещен с JTAGB_SWVTDO
181	188	PB[1]	IO 5V	
180	185	PB[2]	IO 5V	
179	186	PB[3]	IO 5V	
178	183	PB[4]	IO 5V	
177	184	PB[5]	IO 5V	
176	182	PB[6]	IO 5V	
174	181	PB[7]	IO 5V	
173	180	PB[8]	IO 5V	
172	179	PB[9]	IO 5V	
171	178	PB[10]	IO 5V	
170	177	PB[11]	IO 5V	
169	176	PB[12]	IO 5V	
167	175	PB[13]	IO 5V	
166	174	PB[14]	IO 5V	
165	173	PB[15]	IO 5V	
164	172	PB[16]	IO 5V	
163	171	PB[17]	IO 5V	
161	169	PB[18]	IO 5V	
154	164	PB[19]	IO 5V	

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода	
152	162	PВ[20]	IO 5V		
151	161	PВ[21]	IO 5V		
150	160	PВ[22]	IO 5V		
149	158	PВ[23]	IO 5V		
148	159	PВ[24]	IO 5V		
147	156	PВ[25]	IO 5V		
146	157	PВ[26]	IO 5V		
145	154	PВ[27]	IO 5V		
144	155	PВ[28]	IO 5V		
143	153	PВ[29]	IO 5V		
141	151	PВ[30]	IO 5V		
140	152	PВ[31]	IO 5V		
Порт С					
139	149	РС[0]	IO 5V	Пользовательский порт С ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer	
138	150	РС[1]	IO 5V		
137	148	РС[2]	IO 5V		
135	146	РС[3]	IO 5V		
134	147	РС[4]	IO 5V		
133	144	РС[5]	IO 5V		
128	140	РС[6]	AIO		Вывод РС[6] совмещен с сигналом ADC0_REF0+
123	133	РС[7]	AIO		Вывод РС[7] совмещен с сигналом ADC1_REF0-
122	132	РС[8]	AIO		Вывод РС[8] совмещен с сигналом ADC2
121	131	РС[9]	AIO		Вывод РС[9] совмещен с сигналом ADC3
120	130	РС[10]	AIO		Вывод РС[10] совмещен с сигналом ADC4
117	126	РС[11]	AIO		Вывод РС[11] совмещен с сигналом ADC5
116	125	РС[12]	AIO		Вывод РС[12] совмещен с сигналом ADC6
115	124	РС[13]	AIO		Вывод РС[13] совмещен с сигналом ADC7
114	123	РС[14]	AIO		Вывод РС[14] совмещен с сигналом ADC8
112	120	РС[15]	AIO		Вывод РС[15] совмещен с сигналом ADC9
111	119	РС[16]	AIO		Вывод РС[16] совмещен с сигналом ADC10
110	118	РС[17]	AIO		Вывод РС[17] совмещен с сигналом ADC11
109	117	РС[18]	AIO		Вывод РС[18] совмещен с сигналом ADC12
107	114	РС[19]	AIO		Вывод РС[19] совмещен с сигналом ADC13
106	113	РС[20]	AIO		Вывод РС[20] совмещен с сигналом ADC14
105	112	РС[21]	AIO		Вывод РС[21] совмещен с сигналом ADC15
104	111	РС[22]	AIO		Вывод РС[22] совмещен с сигналом ADC16_REF1+
102	108	РС[23]	AIO		Вывод РС[23] совмещен с сигналом ADC17_REF1-
101	107	РС[24]	AIO		Вывод РС[24] совмещен с сигналом DAC0_REF+
100	106	РС[25]	AIO		Вывод РС[25] совмещен с сигналом DAC0_OUT
99	105	РС[26]	AIO		Вывод РС[26] совмещен с сигналом DAC0_REF-
98	104	РС[27]	AIO		Вывод РС[27] совмещен с сигналом DAC1_REF-
96	101	РС[28]	AIO		Вывод РС[28] совмещен с сигналом DAC1_OUT
95	100	РС[29]	AIO		Вывод РС[29] совмещен с сигналом DAC1_REF+
90	94	РС[30]	IO 5V		
89	93	РС[31]	IO 5V		
Порт D					
88	92	PD[0]	IO 5V	Пользовательский порт D ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer	
86	89	PD[1]	IO 5V		
85	88	PD[2]	IO 5V		
84	87	PD[3]	IO 5V		
83	86	PD[4]	IO 5V		
82	85	PD[5]	IO 5V		
80	81	PD[6]	IO 5V		
79	80	PD[7]	IO 5V		
78	79	PD[8]	IO 5V		
77	78	PD[9]	IO 5V		

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода
76	77	PD[10]	IO 5V	
74	73	PD[11]	IO 5V	
73	72	PD[12]	IO 5V	
72	71	PD[13]	IO 5V	
71	70	PD[14]	IO 5V	
62	59	PD[15]	IO 5V	
60	57	PD[16]	IO 5V	
59	58	PD[17]	IO 5V	
58	55	PD[18]	IO 5V	
57	56	PD[19]	IO 5V	
56	53	PD[20]	IO 5V	
55	54	PD[21]	IO 5V	
54	51	PD[22]	IO 5V	
53	52	PD[23]	IO 5V	
52	49	PD[24]	IO 5V	
51	50	PD[25]	IO 5V	
50	47	PD[26]	IO 5V	
49	48	PD[27]	IO 5V	
48	45	PD[28]	IO 5V	
47	46	PD[29]	IO 5V	
46	44	PD[30]	IO 5V	
45	43	PD[31]	IO 5V	
Порт Е				
44	42	PE[0]	IO 5V	Пользовательский порт Е ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer
43	41	PE[1]	IO 5V	
42	40	PE[2]	IO 5V	
41	39	PE[3]	IO 5V	
40	38	PE[4]	IO 5V	
39	37	PE[5]	IO 5V	
38	36	PE[6]	IO 5V	
37	35	PE[7]	IO 5V	
35	34	PE[8]	IO 5V	
34	33	PE[9]	IO 5V	
33	32	PE[10]	IO 5V	
31	31	PE[11]	IO 5V	
30	30	PE[12]	IO 5V	
29	29	PE[13]	IO 5V	
28	28	PE[14]	IO 5V	
27	27	PE[15]	IO 5V	
26	26	PE[16]	IO 5V	
19	18	PE[17]	IO 5V	
18	19	PE[18]	IO 5V	
17	16	PE[19]	IO 5V	
16	17	PE[20]	IO 5V	
15	14	PE[21]	IO 5V	
14	15	PE[22]	IO 5V	
13	12	PE[23]	IO 5V	
12	13	PE[24]	IO 5V	
11	10	PE[25]	IO 5V	
10	11	PE[26]	IO 5V	
9	8	PE[27]	IO 5V	
8	9	PE[28]	IO 5V	
7	6	PE[29]	IO 5V	
6	7	PE[30]	IO 5V	
5	4	PE[31]	IO 5V	

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода
Интерфейс SpaceWire				
207	214	SPW_TXD-	АО	Сигналы приемопередатчика интерфейса SpaceWire
206	213	SPW_TXD+	АО	
205	212	SPW_TXS-	АО	
204	211	SPW_TXS+	АО	
203	210	SPW_RXS-	АИ	
202	209	SPW_RXS+	АИ	
201	208	SPW_RXD-	АИ	
200	207	SPW_RXD+	АИ	
Интерфейс Ethernet				
224	234	ETH_TX+	АО	Сигналы приемопередатчика интерфейса Ethernet
225	235	ETH_TX-	АО	
226	236	ETH_RX+	АИ	
227	237	ETH_RX-	АИ	
Сигналы управления				
1	1	RSTn0	I 5V	Вход нулевого сигнала сброса 0 – запрос сброса 1 – рабочий режим
3	2	RSTn1	I 5V	Вход первого сигнала сброса 0 – запрос сброса 1 – рабочий режим Сброс микросхемы выполняется при условии запроса сброса по обоим выводам RSTn0 и RSTn1
4	5	FT_WRNn	O 5V	Выход с открытым стоком сигнала предупреждения об сбое от блока контроллера сбоев 0 – предупреждение о сбое
2	3	FT_RSTn	O 5V	Выход с открытым стоком сигнала запроса сброса от блока 0 – сигнал запроса сброса
255	269	JTAG_EN	I	Вход выбора тестового режима 0 – рабочий режим 1 – тестовый режим
68	66	STANDBY	O	Выход состояния встроенного регулятора 0 – регулятор включен 1 – регулятор программно выключен
69	67	WAKEUP	I 5V	Вход сигнала запроса включения регулятора 1 – нет запроса включения 0 – есть запрос включения встроенного регулятора
61	60	SHDN	I	Вход выключения встроенных регуляторов напряжения 0 – регуляторы включены 1 – регуляторы выключены
Сигналы тактирования				
253	267	HSE0_OSC_OUT	A	Вывод для подключения внешнего кварцевого резонатора к генератору HSE0
252	266	HSE0_OSC_IN	A	Вывод для подключения кварцевого резонатора или генератора к генератору HSE0
220	229	HSE1_OSC_OUT	A	Вывод для подключения внешнего кварцевого резонатора к генератору HSE1
219	228	HSE1_OSC_IN	A	Вывод для подключения кварцевого резонатора или генератора к генератору HSE1
66	64	LSE_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору LSE
67	65	LSE_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору LSE

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода
Вспомогательные сигналы				
222	231	ETH_IREF		Контрольные выводы проверки источников опорных токов
91	95	DAC_IREF		
195	201	SPW_IREF		
196	202	SPW_EXTR		Выводы для подключения внешних токозадающих резисторов для формирования опорных токов R = 120 кОм ± 5 %
118	127	ADC_EXTR		
93	98	DAC_EXTR0		
94	99	DAC_EXTR1		
221	230	ETH_EXTR		
Питание				
131, 132	142, 145	HV	PWR	Высоковольтный вход напряжения программирования однократно программируемой памяти программ. В рабочем режиме подключить к U _{CC}
92, 243	96, 97, 256, 257	DU _{CC0}	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
129, 130	141, 143	DU _{CC1}	PWR	
87	90, 91	DU _{CC2}	PWR	Питание 1,8 В домена PLL при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
249	263	DU _{CC_PLL}	PWR	
64	61	BDU _{CC}	PWR	Вывод подключения конденсатора к питанию LDO регулятора батарейного домена
65	63	BU _{CC}	PWR	Батарейное питание
63	62	DU _{CC_B}	PWR	Питание 1,8 В цифровой части батарейного домена при SHDN = 1. Выход LDO регулятора при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
20, 21, 70, 153, 155, 254	21, 20, 68, 69, 163, 165, 268, 270	U _{CC}	PWR	Питание площадок ввода-вывода
24, 25, 81, 103, 124, 125, 159, 162, 193, 194, 213, 232, 233	24, 25, 82, 84, 109, 110, 134 – 136, 167, 170, 198 – 200, 220, 221, 243 – 245	U _{CC1}	PWR	Питание встроенного LDO регулятора для формирования напряжений питания DU _{CC0} , DU _{CC1} , DU _{CC2} , DU _{CC_PLL}
108	115, 116	DAC_U _{CC}	PWR	Питание аналоговых блоков ЦАП
126	137, 139	ADC_U _{CC}	PWR	Питание аналоговых блоков АЦП
197	203	SPW_U _{CC}	PWR	Питание приемопередатчика SpaceWire PHY
228	238, 239	ETH_U _{CC}	PWR	Питание приемопередатчика Ethernet PHY
22, 23, 75, 97, 127, 157, 160, 189, 191, 198, 208, 218, 237, 238	22, 23, 74, 76, 102, 103, 138, 166, 168, 196, 197, 204, 205, 215, 226, 227, 249 – 251	GND	PWR	Общий
113	121, 122	DAC_GND	PWR	Общий аналоговых блоков ЦАП
119	128, 129	ADC_GND	PWR	Общий аналоговых блоков АЦП
199	206	SPW_GND	PWR	Общий приемопередатчика SpaceWirePHY
223	232, 233	ETH_GND	PWR	Общий приемопередатчика EthernetPHY
256	–	GND(NC)	PWR	Общий подложки кристалла

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода*	Назначение и функции вывода
32, 36, 136, 142, 156, 158, 168, 175, 192	75, 83	NC	–	Не подключены
<p>Обозначения: PWR – вывод питания I – цифровой вход O – цифровой выход IO – цифровой вход-выход A – аналоговых вывод AI – аналоговый вход AO – аналоговый выход AIO – цифровой вход-выход совмещенный с аналоговых выводом 5V – толерантный к 5 В цифровой вход</p>				

2 Условно-графическое обозначение

183-188,190, 209-212,214-217, 229-231,234-236, 239-242,244-248, 250, 251	Порт А ↔ РА[31-0]	1986BE8Т 1986BE81Т	Порт С ↔ РС[31-0]	89, 90, 95, 96, 98-102, 104-107, 109-112,114-117, 120-123,128, 133-135,137-139
140,141,143-152, 154,161,163-167, 169-174,176-182	Порт В ↔ РВ[31-0]		Порт D ↔ РD[31-0]	45-60, 62, 71-74, 76-80, 82-86, 88
			Порт Е ↔ РЕ[31-0]	5-19, 26-31, 33-35, 37-44
200	SPW_RXD+		SPW_TXD-	207
201	SPW_RXD-		SPW_TXD+	206
202	SPW_RXS+		SPW_TXS-	205
203	SPW_RXS-		SPW_TXS+	204
			SPW_IREF ✗	195
			SPW_EXTR ✗	196
226	ETH_RX+		ETH_TX+	224
227	ETH_RX-		ETH_TX-	225
			ETH_IREF ✗	222
			ETH_EXTR ✗	221
252	✗ HSE0_OSC_IN		HSE0_OSC_OUT ✗	253
219	✗ HSE1_OSC_IN		HSE1_OSC_OUT ✗	220
66	✗ LSE_OSC_IN		LSE_OSC_OUT ✗	67
1	RSTn0		◇ FT_RSTn	2
3	RSTn1		◇ FT_WRNn	4
255	JTAG_EN		STANDBY	68
69	WAKEUP		DAC_IREF ✗	91
131, 132	✗ HV		DAC_EXTR0 ✗	93
61	SHDN		DAC_EXTR1 ✗	94
			ADC_EXTR ✗	118
24,25,81,103,124, 125,159,162,193, 194,213,232,233	✗ Ucc1		DUcc_PLL ✗	249
			DUcc0 ✗	243,92
			DUcc1 ✗	129,130
			DUcc2 ✗	87
65	BUcc		BDUcc ✗	64
			DUcc_B ✗	63
20,21,70,153, 155,254	Ucc			22,23,75,97,127, 157,160,189,191, 198,208,218,237, 238
126	ADC_Ucc		GND ✗	119
108	DAC_Ucc		ADC_GND ✗	113
228	ETH_Ucc		DAC_GND ✗	223
197	SPW_Ucc		ETH_GND ✗	199
			SPW_GND ✗	256
			GND(NC) ✗	32,36,136,142, 156,158,168,175, 192
			NC ✗	

Рисунок 1 – Условно-графическое обозначение

3 Структурные блок-схемы

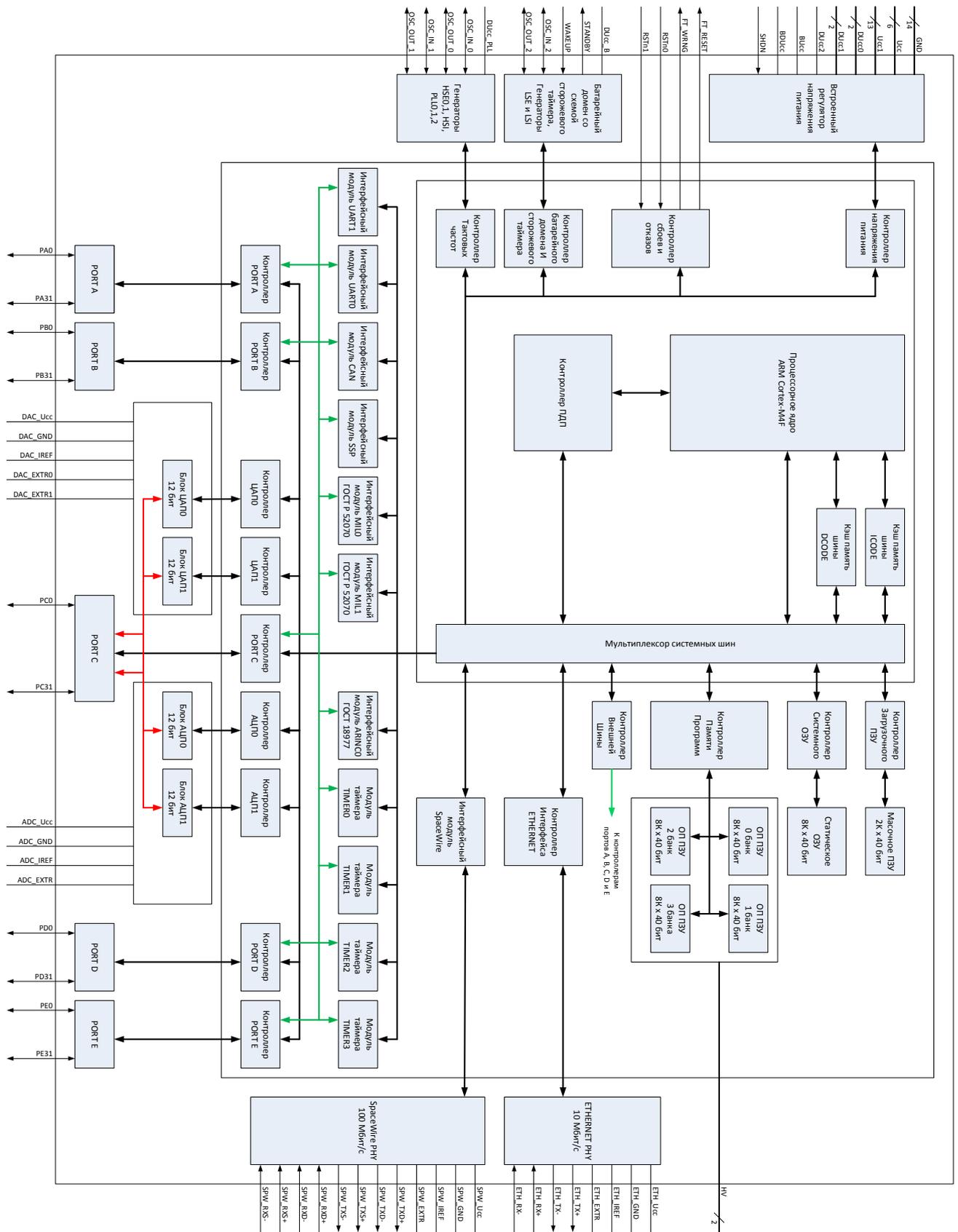


Рисунок 2 – Структурная блок-схема для микросхем 1986BE8Т

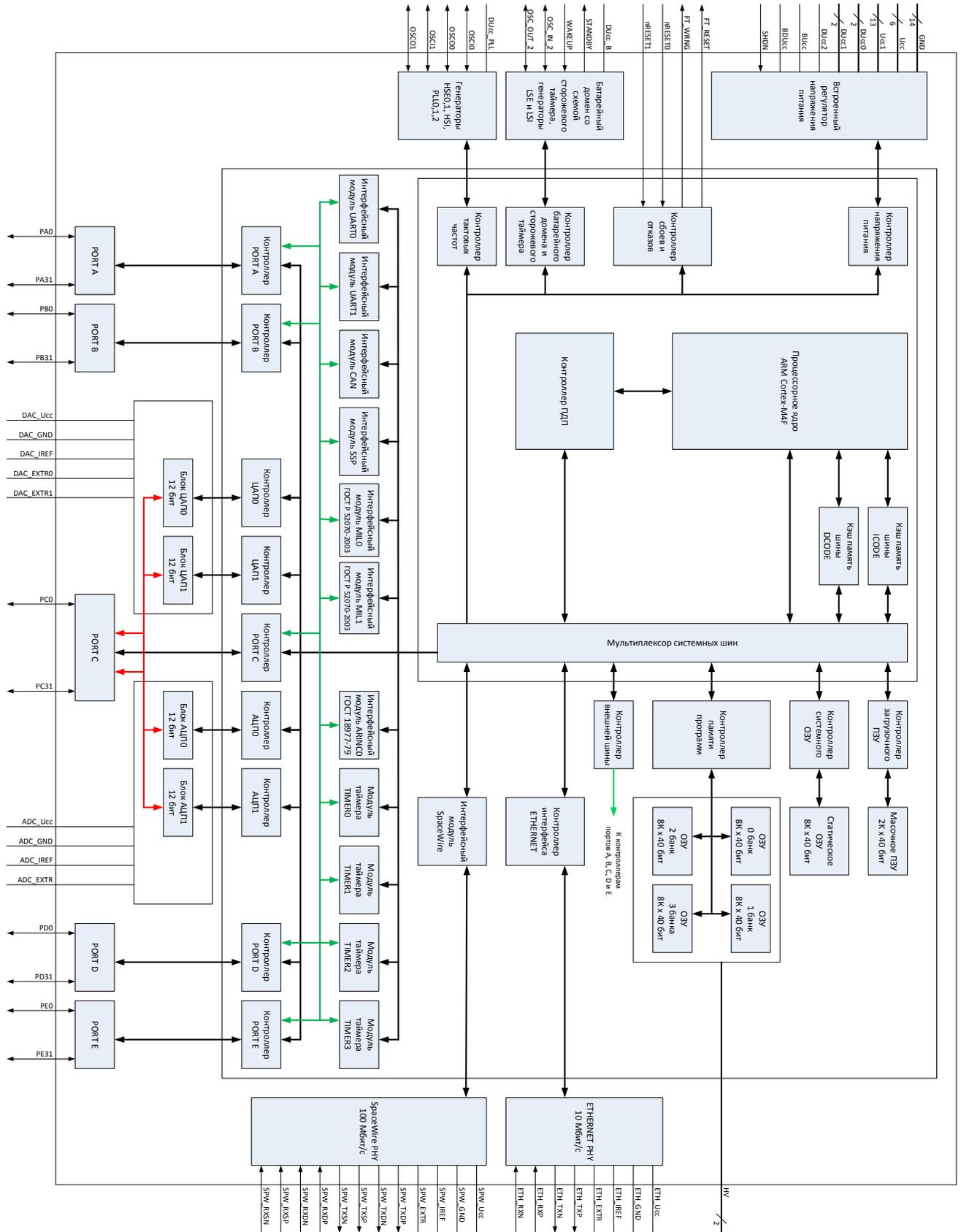


Рисунок 3 – Структурная блок-схема для микросхем 1986BE81Т

4 Расположение выводов микроконтроллера

256	GND (NC)	1	RS7n0	192	Ucc
255	JTAG_EN	2	FT_RS7n	191	GND
254	Ucc	3	RS7n1	190	PA25
253	HS10_OSC_OUT	4	FT_WRNn	189	GND
252	HS10_OSC_IN	5	PE31	188	PA26
251	PA0	6	PE30	187	PA27
250	PA1	7	PE29	186	PA28
249	DUccB-BL	8	PE28	185	PA29
248	PA2	9	PE27	184	PA30
247	PA3	10	PE26	183	PA31
246	PA4	11	PE25	182	PB0
245	PA5	12	PE24	181	PB1
244	PA6	13	PE23	180	PB2
243	DUcc0	14	PE22	179	PB3
242	PA7	15	PE21	178	PB4
241	PA8	16	PE20	177	PB5
240	PA9	17	PE19	176	PB6
239	PA10	18	PE18	175	
238	GND	19	PE17	174	PB7
237	PA11	20	Ucc	173	PB8
236	PA12	21	Ucc	172	PB9
235	PA13	22	GND	171	PB10
234	PA13	23	GND	170	PB11
233	Ucc	24	Ucc1	169	PB12
232	Ucc	25	Ucc1	168	
231	PA14	26	PE16	167	PB13
230	PA15	27	PE15	166	PB14
229	PA16	28	PE14	165	PB15
228	ETH_Ucc	29	PE13	164	PB16
227	ETH_RX	30	PE12	163	PB17
226	ETH_RX	31	PE11	162	Ucc1
225	ETH_TX	32		161	PB18
224	ETH_TX	33	PE10	160	GND
223	ETH_GND	34	PE9	159	Ucc1
222	ETH_IRQ	35	PE8	158	
221	ETH_EXTB	36		157	GND
220	HS10_OSC_OUT	37	PE7	156	
219	HS10_OSC_IN	38	PE6	155	Ucc
218	GND	39	PE5	154	PB19
217	PA17	40	PE4	153	Ucc
216	PA18	41	PE3	152	PB20
215	PA19	42	PE2	151	PB21
214	PA20	43	PE1	150	PB22
213	Ucc	44	PE0	149	PB23
212	PA21	45	PD31	148	PB24
211	PA22	46	PD30	147	PB25
210	PA23	47	PD29	146	PB26
209	PA24	48	PD28	145	PB27
208	GND	49	PD27	144	PB28
207	SPW_TX0	50	PD26	143	PB29
206	SPW_TX0	51	PD25	142	
205	SPW_TX5	52	PD24	141	PB30
204	SPW_TX5	53	PD23	140	PB31
203	SPW_RX5	54	PD22	139	PC0
202	SPW_RX5	55	PD21	138	PC1
201	SPW_RX0	56	PD20	137	PC2
200	SPW_RX0	57	PD19	136	
199	SPW_GND	58	PD18	135	PC3
198	GND	59	PD17	134	PC4
197	SPW_Ucc	60	PD16	133	PC5
196	SPW_EXT0	61	SHDN	132	HV
195	SPW_IRQ	62	PD15	131	HV
194	Ucc	63	DUccB	130	DUcc1
193	Ucc	64	BDUcc	129	DUcc1
65	BUcc				
66	LSF_OSC_IN				
67	LSF_OSC_OUT				
68	SATNBUR				
69	WAKEUP				
70	Ucc				
71	PD14				
72	PD13				
73	PD12				
74	PD11				
75	GND				
76	PD10				
77	PD9				
78	PD8				
79	PD7				
80	PD6				
81	Ucc1				
82	PD5				
83	PD4				
84	PD3				
85	PD2				
86	PD1				
87	DUcc2				
88	PD0				
89	PC31				
90	PC30				
91	DAC_IREF				
92	DUcc0				
93	DAC_EXTR0				
94	DAC_EXTR1				
95	PC29				
96	PC28				
97	GND				
98	PC27				
99	PC26				
100	PC25				
101	PC24				
102	PC23				
103	Ucc1				
104	PC22				
105	PC21				
106	PC20				
107	PC19				
108	DAC_Ucc				
109	PC18				
110	PC17				
111	PC16				
112	PC15				
113	DAC_GND				
114	PC14				
115	PC13				
116	PC12				
117	PC11				
118	DAC_EXTR				
119	DAC_GND				
120	PC10				
121	PC9				
122	PC8				
123	PC7				
124	Ucc1				
125	HS10				
126	DAC_Ucc				
127	GND				
128	PC6				

Рисунок 4 – Расположение выводов микроконтроллера

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

FUNC 15	FUNC 14	FUNC 13	FUNC 12	FUNC 11	FUNC 10	FUNC 9	FUNC 8	FUNC 7	FUNC 6	FUNC 5	FUNC 4	FUNC 3	FUNC 2	FUNC 1	FUNC 0	ANALOG	PIN	
																	RSTn0	1
																	FT_RSTn	2
																	RSTn1	3
																	FT_WRNn	4
			MILO_RXBN	MIL1_RXBN		ARCO_IN5P	TMR0_CH2N		TMR3_CH0N		UART0_TX		CS[4]	DATA[41]		PE31	MODE[7]	5
			MILO_RXBP	MIL1_RXBP		ARCO_IN5N	TMR0_CH2P		TMR3_CH0P		CAN0_RX	CS[5]	DATA[40]		PE30		MODE[6]	6
			MILO_ENB	MIL1_ENB		ARCO_IN4P	TMR0_CH1N		TMR3_ETR		CAN0_TX	CS[6]	DATA[39]		PE29		MODE[5]	7
			MILO_TXBP	MIL1_TXBP		ARCO_IN4N	TMR0_CH1P		TMR2_BRK			CS[7]	DATA[38]		PE28		MODE[4]	8
			MILO_TXBN	MIL1_TXBN		ARCO_OUT1P	TMR0_CH0N		TMR2_CH3N			ADDR[31]	DATA[37]		PE27		MODE[3]	9
			MILO_RXAN	MIL1_RXAN		ARCO_OUT1N	TMR0_CH0P		TMR2_CH3P	SSPO_RX	UART1_RX	ADDR[30]	DATA[36]		PE26		MODE[2]	10
			MILO_RXAP	MIL1_RXAP		ARCO_IN3P	TMR0_ETR		TMR2_CH2N	SSPO_FS	UART1_TX	ADDR[29]	DATA[35]		PE25		MODE[1]	11
			MILO_ENA	MIL1_ENA		ARCO_IN3N			TMR2_CH2P	SSPO_CLK	UART0_RX	ADDR[28]	DATA[34]		PE24		MODE[0]	12
			MILO_TXAP	MIL1_TXAP		ARCO_IN2P			TMR2_CH1N	SSPO_TX	UART0_TX	ADDR[27]	DATA[33]		PE23			13
			MILO_TXAN	MIL1_TXAN		ARCO_IN2N			TMR2_CH1P			ADDR[26]	DATA[32]		PE22			14
			MIL1_RXBN	MILO_RXBN		ARCO_OUT0P		TMR3_ETR	TMR2_CH0N			ADDR[25]	DATA[31]		PE21			15
			MIL1_RXBP	MILO_RXBP		ARCO_OUT0N		TMR3_CH0P	TMR2_CH0P		CAN0_RX	ADDR[24]	DATA[30]		PE20			16
			MIL1_ENB	MILO_ENB		ARCO_IN1P		TMR3_CH0N	TMR2_ETR		CAN0_TX	ADDR[23]	DATA[29]		PE19			17
			MIL1_TXBP	MILO_TXBP		ARCO_IN1N		TMR3_CH1P	TMR1_BRK		UART1_RX	ADDR[22]	DATA[28]		PE18			18
			MIL1_TXBN	MILO_TXBN		ARCO_IN0P		TMR3_CH1N	TMR1_CH3N		UART1_TX	ADDR[21]	DATA[27]		PE17			19
																	Ucc	20
																	Ucc	21
																	GND	22
																	GND	23
																	Ucc1	24
																	Ucc1	25
			MIL1_RXAN	MILO_RXAN		ARCO_IN0N		TMR3_CH2P	TMR1_CH3P		UART0_RX	CLOCK	DATA[66]		PE16			26
			MIL1_RXAP	MILO_RXAP		ARCO_OUT3P		TMR3_CH2N	TMR1_CH2N		UART0_TX	OCLK	DATA[65]		PE15			27
			MIL1_ENA	MILO_ENA		ARCO_OUT3N		TMR3_CH3P	TMR1_CH2P				DATA[64]		PE14			28
			MIL1_TXAP	MILO_TXAP		ARCO_IN7P		TMR3_CH3N	TMR1_CH1N				DATA[15]		PE13			29
			MIL1_TXAN	MILO_TXAN		ARCO_IN7N		TMR3_BRK	TMR1_CH1P				DATA[14]		PE12			30
			MILO_RXBN	MIL1_RXBN		ARCO_IN6P		TMR2_ETR	TMR1_CH0N				DATA[13]		PE11			31
																		32
			MILO_RXBP	MIL1_RXBP		ARCO_IN6N		TMR2_CH0P	TMR1_CH0P	SSPO_RX	UART1_RX	CAN0_RX	DATA[12]		PE10			33
			MILO_ENB	MIL1_ENB		ARCO_OUT2P		TMR2_CH0N	TMR1_ETR	SSPO_FS	UART1_TX	CAN0_TX	DATA[11]		PE9			34
			MILO_TXBP	MIL1_TXBP		ARCO_OUT2N		TMR2_CH1P	TMR0_BRK	SSPO_CLK	UART0_RX		DATA[10]		PE8			35
																		36
			MILO_TXBN	MIL1_TXBN		ARCO_IN5P		TMR2_CH1N	TMR0_CH3N	SSPO_TX	UART0_TX		DATA[9]		PE7			37
			MILO_RXAN	MIL1_RXAN		ARCO_IN5N		TMR2_CH2P	TMR0_CH3P				DATA[8]		PE6			38
			MILO_RXAP	MIL1_RXAP		ARCO_IN4P		TMR2_CH2N	TMR0_CH2N				DATA[7]		PE5			39
			MILO_ENA	MIL1_ENA		ARCO_IN4N		TMR3_BRK	TMR2_CH3P	TMR0_CH2P			DATA[6]		PE4			40
			MILO_TXAP	MIL1_TXAP		ARCO_OUT1P		TMR3_CH3N	TMR2_CH3N	TMR0_CH1N			DATA[5]		PE3			41
			MILO_TXAN	MIL1_TXAN		ARCO_OUT1N		TMR3_CH3P	TMR2_BRK	TMR0_CH1P		UART1_RX	DATA[4]		PE2			42
			MIL1_RXBN	MILO_RXBN		ARCO_IN3P		TMR3_CH2N	TMR1_ETR	TMR0_CH0N		UART1_TX	DATA[3]		PE1			43
			MIL1_RXBP	MILO_RXBP		ARCO_IN3N		TMR3_CH2P	TMR1_CH0P	TMR0_CH0P		UART0_RX	CAN0_RX	DATA[2]		PE0		44
			MIL1_ENB	MILO_ENB		ARCO_IN2P		TMR3_CH1N	TMR1_CH0N	TMR0_ETR		UART0_TX	CAN0_TX	CLOCK	DATA[1]		PD31	45
			MIL1_TXBP	MILO_TXBP		ARCO_IN2N		TMR3_CH1P	TMR1_CH1P			OCLK	DATA[0]		PD30			46
			MIL1_TXBN	MILO_TXBN		ARCO_OUT0P		TMR3_CH0N	TMR1_CH1N			BE[8]	BWE[8]		PD29			47
			MIL1_RXAN	MILO_RXAN		ARCO_OUT0N		TMR3_CH0P	TMR1_CH2P			BE[5]	BWE[5]		PD28			48
			MIL1_RXAP	MILO_RXAP		ARCO_IN1P		TMR3_ETR	TMR1_CH2N			BE[4]	BWE[4]		PD27			49
			MILO_ENA	MILO_ENA		ARCO_IN1N		TMR2_BRK	TMR1_CH3P		SSPO_RX	UART1_RX	BE[1]	BWE[1]		PD26		50
			MIL1_TXAP	MILO_TXAP		ARCO_IN0P		TMR2_CH3N	TMR1_CH3N		SSPO_FS	UART1_TX	BE[0]	BWE[0]		PD25		51
			MIL1_TXAN	MILO_TXAN		ARCO_IN0N		TMR2_CH3P	TMR1_BRK		SSPO_CLK	UART0_RX	WE[0]		PD24			52
			MILO_RXBN	MIL1_RXBN		ARCO_OUT3P		TMR2_CH2N	TMR0_ETR		SSPO_TX	UART0_TX	OEN[0]		PD23			53
			MILO_RXBP	MIL1_RXBP		ARCO_OUT3N		TMR2_CH2P	TMR0_CH0P		CAN0_RX	CLOCK	Csn[3]		PD22			54
			MILO_ENB	MIL1_ENB		ARCO_IN7P		TMR2_CH1N	TMR0_CH0N		CAN0_TX	OCLK	Csn[2]		PD21			55
			MILO_TXBP	MIL1_TXBP		ARCO_IN7N		TMR2_CH1P	TMR0_CH1P				Csn[1]		PD20			56
			MILO_TXBN	MIL1_TXBN		ARCO_IN6P		TMR2_CH0N	TMR0_CH1N				Csn[0]		PD19			57
			MILO_RXAN	MIL1_RXAN		ARCO_IN6N		TMR2_CH0P	TMR0_CH2P		UART1_RX	ADDR[20]		PD18				58
			MILO_RXAP	MIL1_RXAP		ARCO_OUT2P		TMR2_ETR	TMR0_CH2N		UART1_TX	ADDR[19]		PD17				59
			MILO_ENA	MIL1_ENA		ARCO_OUT2N		TMR1_BRK	TMR0_CH3P		UART0_RX	ADDR[18]		PD16				60
																	SHDN	61
			MILO_TXAP	MIL1_TXAP		ARCO_IN5P		TMR1_CH3N	TMR0_CH3N		UART0_TX	ADDR[17]		PD15				62
																	DUccB	63
																	BDUcc	64
15 FUNC	14 FUNC	13 FUNC	12 FUNC	11 FUNC	10 FUNC	9 FUNC	8 FUNC	7 FUNC	6 FUNC	5 FUNC	4 FUNC	3 FUNC	2 FUNC	1 FUNC	0 FUNC	ANALOG	PIN	

Рисунок 5 – Функции выводов 1 ÷ 64 в различных режимах

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

FUNC 15	FUNC 14	FUNC 13	FUNC 12	FUNC 11	FUNC 10	FUNC 9	FUNC 8	FUNC 7	FUNC 6	FUNC 5	FUNC 4	FUNC 3	FUNC 2	FUNC 1	FUNC 0	ANALOG	PIN			
																	Bucc	65		
																	LSE_OSC_IN	66		
																	LSE_OSC_OUT	67		
																	SATNDBY	68		
																	WAKEUP	69		
																	Ucc	70		
		MILO_TXAN	MIL1_TXAN		ARCO_IN5N	TMR1_CH3P	TMR0_BRK						ADDR[16]				PD14	71		
		MIL1_RXBN	MILO_RXBN		ARCO_IN4P	TMR1_CH2N							ADDR[15]				PD13	72		
		MIL1_RXBP	MILO_RXBP		ARCO_IN4N	TMR1_CH2P					CAN0_RX		ADDR[14]				PD12	73		
		MIL1_ENB	MILO_ENB		ARCO_OUT1P	TMR1_CH1N					CAN0_TX		ADDR[13]				PD11	74		
																		GND	75	
		MIL1_TXBP	MILO_TXBP		ARCO_OUT1N	TMR1_CH1P		TMR3_BRK	SSPO_RX	UART1_RX			ADDR[12]				PD10	76		
		MIL1_TXBN	MILO_TXBN		ARCO_IN3P	TMR1_CH0N		TMR3_CH3N	SSPO_FS	UART1_TX			ADDR[11]				PD9	77		
		MIL1_RXAN	MILO_RXAN		ARCO_IN3N	TMR1_CH0P		TMR3_CH3P	SSPO_CLK	UART0_RX			ADDR[10]				PD8	78		
		MIL1_RXAP	MILO_RXAP		ARCO_IN2P	TMR1_ETR		TMR3_CH2N	SSPO_TX	UART0_TX			ADDR[9]				PD7	79		
		MIL1_ENA	MILO_ENA		ARCO_IN2N	TMR0_BRK		TMR3_CH2P					ADDR[8]				PD6	80		
																		Ucc1	81	
		MIL1_TXAP	MILO_TXAP		ARCO_OUT0P	TMR0_CH3N		TMR3_CH1N					ADDR[7]				PD5	82		
		MIL1_TXAN	MILO_TXAN		ARCO_OUT0N	TMR0_CH3P		TMR3_CH1P					ADDR[6]				PD4	83		
		MILO_RXBN	MIL1_RXBN		ARCO_IN1P	TMR0_CH2N		TMR3_CH0N					ADDR[5]				PD3	84		
		MILO_RXBP	MIL1_RXBP		ARCO_IN1N	TMR0_CH2P		TMR3_CH0P		UART1_RX	CAN0_RX		ADDR[4]				PD2	85		
		MILO_ENB	MIL1_ENB		ARCO_IN0P	TMR0_CH1N		TMR3_ETR		UART1_TX	CAN0_TX		ADDR[3]				PD1	86		
																		DUcc2	87	
		MILO_TXBP	MIL1_TXBP		ARCO_IN0N	TMR0_CH1P		TMR2_BRK		UART0_RX			ADDR[2]				PD0	88		
		MILO_TXBN	MIL1_TXBN	ARCO_OUT3P		TMR0_CH0N		TMR2_CH3N		UART0_TX			ADDR[1]				PC31	89		
		MILO_RXAN	MIL1_RXAN	ARCO_OUT3N		TMR0_CH0P		TMR2_CH3P					ADDR[0]				PC30	90		
																		DAC_REF	91	
																		DUcc0	92	
																		DAC_EXTR0	93	
																		DAC_EXTR1	94	
		MILO_RXAP	MIL1_RXAP	ARCO_IN7P		TMR0_ETR		TMR2_CH2N			CLOCK	ADDR[31]				PC29		DAC1_REF+	95	
		MILO_ENA	MIL1_ENA	ARCO_IN7N				TMR2_CH2P			OCLK					PC28		DAC1_OUT	96	
																		GND	97	
		MILO_TXAP	MIL1_TXAP	ARCO_IN6P				TMR2_CH1N					ADDR[30]				PC27		DAC1_REF-	98
		MILO_TXAN	MIL1_TXAN	ARCO_IN6N				TMR2_CH1P	SSPO_RX	UART1_RX			CS[5]				PC26		DAC0_REF-	99
		MIL1_RXBN	MILO_RXBN	ARCO_OUT2P				TMR3_ETR	TMR2_CH0N	SSPO_FS	UART1_TX						PC25		DAC0_OUT	100
		MIL1_RXBP	MILO_RXBP	ARCO_OUT2N				TMR3_CH0P	TMR2_CH0P	SSPO_CLK	UART0_RX	CAN0_RX		CS[6]			PC24		DAC0_REF+	101
		MIL1_ENB	MILO_ENB	ARCO_IN5P				TMR3_CH0N	TMR2_ETR	SSPO_TX	UART0_TX	CAN0_TX	CLOCK				PC23		ADC17_REF1-	102
																			Ucc1	103
		MIL1_TXBP	MILO_TXBP	ARCO_IN5N				TMR3_CH1P	TMR1_BRK				OCLK				PC22		ADC16_REF1+	104
		MIL1_TXBN	MILO_TXBN	ARCO_IN4P				TMR3_CH1N	TMR1_CH3N				ADDR[29]				PC21		ADC15	105
		MIL1_RXAN	MILO_RXAN	ARCO_IN4N				TMR3_CH2P	TMR1_CH3P				ADDR[28]				PC20		ADC14	106
		MIL1_RXAP	MILO_RXAP	ARCO_OUT1P				TMR3_CH2N	TMR1_CH2N				ADDR[27]				PC19		ADC13	107
																			DAC_Ucc	108
		MIL1_ENA	MILO_ENA	ARCO_OUT1N				TMR3_CH3P	TMR1_CH2P		UART1_RX		ADDR[26]				PC18		ADC12	109
		MIL1_TXAP	MILO_TXAP	ARCO_IN3P				TMR3_CH3N	TMR1_CH1N		UART1_TX		ADDR[25]				PC17		ADC11	110
		MIL1_TXAN	MILO_TXAN	ARCO_IN3N				TMR3_BRK	TMR1_CH1P		UART0_RX		ADDR[24]				PC16		ADC10	111
		MILO_RXBN	MIL1_RXBN	ARCO_IN2P				TMR2_ETR	TMR1_CH0N		UART0_TX		ADDR[23]				PC15		ADC9	112
																			DAC_GND	113
		MILO_RXBP	MIL1_RXBP	ARCO_IN2N				TMR2_CH0P	TMR1_CH0P		CAN0_RX		ADDR[22]				PC14		ADC8	114
		MILO_ENB	MIL1_ENB	ARCO_OUT0P				TMR2_CH0N	TMR1_ETR		CAN0_TX		ADDR[21]				PC13		ADC7	115
		MILO_TXBP	MIL1_TXBP	ARCO_OUT0N				TMR2_CH1P	TMR0_BRK				CS[7]				PC12		ADC6	116
		MILO_TXBN	MIL1_TXBN	ARCO_IN1P				TMR2_CH1N	TMR0_CH3N								PC11		ADC5	117
																			ADC_EXTR	118
																			ADC_GND	119
		MILO_RXAN	MIL1_RXAN	ARCO_IN1N				TMR2_CH2P	TMR0_CH3P	SSPO_RX	UART1_RX						PC10		ADC4	120
		MILO_RXAP	MIL1_RXAP	ARCO_IN0P				TMR2_CH2N	TMR0_CH2N	SSPO_FS	UART1_TX						PC9		ADC3	121
		MILO_ENA	MIL1_ENA	ARCO_IN0N				TMR3_BRK	TMR2_CH3P	TMR0_CH2P	SSPO_CLK	UART0_RX					PC8		ADC2	122
		MILO_TXAP	MIL1_TXAP	ARCO_OUT3P	TMR3_CH3N	TMR2_CH3N	TMR0_CH1N	SSPO_TX	UART0_TX								PC7		ADC1_REF0-	123
																			Ucc1	124
																			Ucc1	125
																			ADC_Ucc	126
																			GND	127
		MILO_TXAN	MIL1_TXAN		ARCO_OUT3N	TMR3_CH3P	TMR2_BRK	TMR0_CH1P									PC6		ADC0_REF0+	128
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ANALOG	PIN			

Рисунок 6 – Функции выводов 65 ÷ 128 в различных режимах

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

	PIN	ANALOG	FUNC 0	FUNC 1	FUNC 2	FUNC 3	FUNC 4	FUNC 5	FUNC 6	FUNC 7	FUNC 8	FUNC 9	FUNC 10	FUNC 11	FUNC 12	FUNC 13	FUNC 14	FUNC 15
192																		
191	GND																	
190			PA25		DATA[57]	READY[2]		UART1_TX	SSP0_FS	TMR1_CH3N	TMR3_CH1N			ARCO_IN0P	MIL0_TXBN	MIL1_TXBN	MAC0_TXD2	
189	GND																	
188			PA26		DATA[56]	READY[3]		UART1_RX	SSP0_RX	TMR1_BRK	TMR3_CH1P			ARCO_IN1N	MIL0_TXBP	MIL1_TXBP	MAC0_TXD3	
187			PA27		DATA[55]	READY[4]	CAN0_TX			TMR2_ETR	TMR3_CH0N			ARCO_IN1P	MIL0_ENB	MIL1_ENB	MAC0_TXEN	
186			PA28		DATA[54]	READY[5]	CAN0_RX			TMR2_CH0P	TMR3_CH0P			ARCO_OUT0N	MIL0_RXBP	MIL1_RXBP	MAC0_TXERR	
185			PA29		DATA[53]	READY[6]				TMR2_CH0N	TMR3_ETR			ARCO_OUT0P	MIL0_RXBN	MIL1_RXBN	MAC0_TXCLK	
184			PA30		DATA[52]	READY[7]				TMR2_CH1P				ARCO_IN2N	MIL1_TXAN	MIL0_TXAN	MAC0_COL	
183			PA31		DATA[51]	CS[0]		UART0_TX		TMR2_CH1N				ARCO_IN2P	MIL1_TXAP	MIL0_TXAP	MAC0_CRS	
182			PB0		DATA[50]	CS[1]		UART0_RX		TMR2_CH2P				ARCO_IN3N	MIL1_ENA	MIL0_ENA	SPW0_TXD	
181			PB1		DATA[49]	CS[2]		UART1_TX		TMR2_CH2N		TMR0_ETR		ARCO_IN3P	MIL1_RXAP	MIL0_RXAP	SPW0_TXS	
180			PB2		DATA[48]	CS[3]		UART1_RX		TMR2_CH3P		TMR0_CH0P		ARCO_OUT1N	MIL1_RXAN	MIL0_RXAN	SPW0_RXD	
179			PB3		DATA[79]					TMR2_CH0N		TMR0_CH0N		ARCO_OUT1P	MIL1_TXBN	MIL0_TXBN	SPW0_RXS	
178	JB_NTRST		PB4		DATA[78]					TMR2_BRK		TMR0_CH1P		ARCO_IN4N	MIL1_TXBP	MIL0_TXBP		
177	JB_SWCLKTCK		PB5		DATA[77]		CAN0_TX			TMR3_ETR		TMR0_CH1N		ARCO_IN4P	MIL1_ENB	MIL0_ENB		
176	JB_SWDIOTMS		PB6		DATA[76]		CAN0_RX			TMR3_CH0P		TMR0_CH2P		ARCO_IN5N	MIL1_RXBP	MIL0_RXBP		
175																		
174	JB_TDI		PB7		DATA[75]			UART0_TX	SSP0_TX	TMR3_CH0N		TMR0_CH2N		ARCO_IN5P	MIL1_RXBN	MIL0_RXBN		
173	JB_SWVTD0		PB8		DATA[74]			UART0_RX	SSP0_CLK	TMR3_CH1P		TMR0_CH3P		ARCO_OUT2N	MIL0_TXAN	MIL1_TXAN		
172			PB9		DATA[73]	OCLK		UART1_TX	SSP0_FS	TMR3_CH1N		TMR0_CH3N		ARCO_OUT2P	MIL0_TXAP	MIL1_TXAP		
171			PB10		DATA[72]	CLOCK		UART1_RX	SSP0_RX	TMR3_CH2P		TMR0_BRK		ARCO_IN6N	MIL0_ENA	MIL1_ENA		
170			PB11		DATA[31]					TMR3_CH2N		TMR1_ETR		ARCO_IN6P	MIL0_RXAP	MIL1_RXAP		
169			PB12		DATA[30]					TMR3_CH3P		TMR1_CH0P		ARCO_IN7N	MIL0_RXAN	MIL1_RXAN		
168																		
167			PB13		DATA[29]					TMR3_CH3N		TMR1_CH0N		ARCO_IN7P	MIL0_TXBN	MIL1_TXBN		
166			PB14		DATA[28]					TMR3_BRK		TMR1_CH1P		ARCO_OUT3N	MIL0_TXBP	MIL1_TXBP		
165			PB15		DATA[27]		CAN0_TX	UART0_TX		TMR1_CH1N				ARCO_OUT3P	MIL0_ENB	MIL1_ENB		
164			PB16		DATA[26]		CAN0_RX	UART0_RX			TMR1_CH2P		ARCO_IN0N		MIL0_RXBP	MIL1_RXBP		
163			PB17		DATA[25]			UART1_TX			TMR1_CH2N		ARCO_IN0P		MIL0_RXBN	MIL1_RXBN		
162	Ucc1																	
161			PB18		DATA[24]			UART1_RX		TMR0_BRK	TMR1_CH3P		ARCO_IN1N		MIL1_TXAN	MIL0_TXAN		
160	GND																	
159	Ucc1																	
158																		
157	GND																	
156																		
155	Ucc																	
154			PB19		DATA[23]					TMR0_CH3N	TMR1_CH3N		ARCO_IN1P		MIL1_TXAP	MIL0_TXAP		
153	Ucc																	
152			PB20		DATA[22]					TMR0_CH3P	TMR1_BRK		ARCO_OUT0N		MIL1_ENA	MIL0_ENA		
151			PB21		DATA[21]					TMR0_CH2N	TMR2_ETR		ARCO_OUT0P		MIL1_RXAP	MIL0_RXAP		
150			PB22		DATA[20]					TMR0_CH2P	TMR2_CH0P		ARCO_IN2N		MIL1_RXAN	MIL0_RXAN		
149			PB23		DATA[19]			UART0_TX	SSP0_TX	TMR0_CH1N	TMR2_CH0N		ARCO_IN2P		MIL1_TXBN	MIL0_TXBN		
148			PB24		DATA[18]			UART0_RX	SSP0_CLK	TMR0_CH1P	TMR2_CH1P		ARCO_IN3N		MIL1_TXBP	MIL0_TXBP		
147			PB25		DATA[17]		CAN0_TX	UART1_TX	SSP0_FS	TMR0_CH0N	TMR2_CH1N		ARCO_IN3P		MIL1_ENB	MIL0_ENB		
146			PB26		DATA[16]		CAN0_RX	UART1_RX	SSP0_RX	TMR0_CH0P	TMR2_CH2P		ARCO_OUT1N		MIL1_RXBP	MIL0_RXBP		
145			PB27		CSn[7]	OCLK				TMR0_ETR	TMR2_CH2N		ARCO_OUT1P		MIL1_RXBN	MIL0_RXBN		
144			PB28		CSn[6]	CLOCK				TMR1_BRK	TMR2_CH3P		ARCO_IN4N		MIL0_TXAN	MIL1_TXAN		
143			PB29		CSn[5]					TMR1_CH3N	TMR2_CH3N		ARCO_IN4P		MIL0_TXAP	MIL1_TXAP		
142																		
141			PB30		CSn[4]					TMR1_CH3P	TMR2_BRK		ARCO_IN5N		MIL0_ENA	MIL1_ENA		
140			PB31		BWE[9]	BE[9]		UART0_TX		TMR1_CH2N	TMR3_ETR		ARCO_IN5P		MIL0_RXAP	MIL1_RXAP		
139			PC0		BWE[7]	BE[7]		UART0_RX		TMR1_CH2P	TMR3_CH0P		ARCO_OUT2N		MIL0_RXAN	MIL1_RXAN		
138			PC1		BWE[6]	BE[6]		UART1_TX		TMR1_CH1N	TMR3_CH0N		ARCO_OUT2P		MIL0_TXBN	MIL1_TXBN		
137			PC2		BWE[3]	BE[3]		UART1_RX		TMR1_CH1P	TMR3_CH1P		ARCO_IN6N		MIL0_TXBP	MIL1_TXBP		
136																		
135			PC3		BWE[2]	BE[2]	CAN0_TX			TMR0_ETR	TMR1_CH0N	TMR3_CH1N		ARCO_IN6P		MIL0_ENB	MIL1_ENB	
134			PC4		WE[1]		CAN0_RX			TMR0_CH0P	TMR1_CH0P	TMR3_CH2P		ARCO_IN7N		MIL0_RXBP	MIL1_RXBP	
133			PC5		OE[1]					TMR0_CH0N	TMR1_ETR	TMR3_CH2N		ARCO_IN7P		MIL0_RXBN	MIL1_RXBN	
132	HV																	
131	HV																	
130	DUcc1																	
129	DUcc1																	
PIN	ANALOG	FUNC 0	FUNC 1	FUNC 2	FUNC 3	FUNC 4	FUNC 5	FUNC 6	FUNC 7	FUNC 8	FUNC 9	FUNC 10	FUNC 11	FUNC 12	FUNC 13	FUNC 14	FUNC 15	

Рисунок 7 – Функции выводов 129 ÷ 192 в различных режимах

	PIN	ANALOG	FUNC 0	FUNC 1	FUNC 2	FUNC 3	FUNC 4	FUNC 5	FUNC 6	FUNC 7	FUNC 8	FUNC 9	FUNC 10	FUNC 11	FUNC 12	FUNC 13	FUNC 14	FUNC 15
256																		
255	JTAG_EN																	
254	Ucc																	
253	HSE1_OSC_OUT																	
252	HSE1_OSC_IN																	
251	JA_NTRST		PA0		DATA[42]			UART0_RX		TMR3_CH1P		TMR0_CH3P	ARCO_IN0N		MIL0_TXAN	MIL1_TXAN		
250	JA_SWCLKTK		PA1		DATA[43]			UART1_TX		TMR3_CH1N		TMR0_CH3N	ARCO_IN0P		MIL0_TXAP	MIL1_TXAP		
249	DUcc_PLL																	
248	JA_SVIDIOTMS		PA2		DATA[44]			UART1_RX		TMR3_CH2P		TMR0_BRK	ARCO_IN1N		MIL0_ENA	MIL1_ENA		
247	JA_TDI		PA3		DATA[45]					TMR3_CH2N		TMR1_ETR	ARCO_IN1P		MIL0_RXAP	MIL1_RXAP		
246	JA_SWWTDO		PA4		DATA[46]					TMR3_CH3P		TMR1_CH0P	ARCO_OUT0N		MIL0_RXAN	MIL1_RXAN		
245			PA5		DATA[47]					TMR3_CH3N		TMR1_CH0N	ARCO_OUT0P		MIL0_TXBN	MIL1_TXBN		
244			PA6		CS[4]					TMR3_BRK		TMR1_CH1P	ARCO_IN2N		MIL0_TXBP	MIL1_TXBP		
243	DUcc0																	
242			PA7		CS[3]		CAN0_TX	UART0_TX	SSP0_TX	TMR0_ETR		TMR1_CH1N	ARCO_IN2P		MIL0_ENB	MIL1_ENB	MAC0_MDC	
241			PA8		CS[2]		CAN0_RX	UART0_RX	SSP0_CLK	TMR0_CH0P		TMR1_CH2P	ARCO_IN3N		MIL0_RXBP	MIL1_RXBP	MAC0_MDCTR	
240			PA9		CS[1]			UART1_TX	SSP0_FS	TMR0_CH0N		TMR1_CH2N	ARCO_IN3P		MIL0_RXBN	MIL1_RXBN	MAC0_MDCOE	
239			PA10		CS[0]			UART1_RX	SSP0_RX	TMR0_CH1P	TMR2_BRK	TMR1_CH3P	ARCO_OUT1N		MIL1_TXAN	MIL0_TXAN	MAC0_MDI	
238	GND																	
237	GND																	
236			PA11		READY[7]	OCLK				TMR0_CH1N	TMR2_CH3N	TMR1_CH3N	ARCO_OUT1P		MIL1_TXAP	MIL0_TXAP	MAC0_MDO	
235			PA12		READY[6]	CLOCK				TMR0_CH2P	TMR2_CH3P	TMR1_BRK	ARCO_IN4N		MIL1_ENA	MIL0_ENA	MAC0_MDOTR	TRACE_DATA3
234			PA13		READY[5]					TMR0_CH2N	TMR2_CH2N		ARCO_IN4P		MIL1_RXAP	MIL0_RXAP	MAC0_MDOOE	TRACE_DATA2
233	Ucc1																	
232	Ucc1																	
231			PA14		READY[4]					TMR0_CH3P	TMR2_CH2P		ARCO_IN5N		MIL1_RXAN	MIL0_RXAN	MAC0_MDINT	TRACE_DATA1
230			PA15		READY[3]			UART0_TX		TMR0_CH3N	TMR2_CH1N		ARCO_IN5P		MIL1_TXBN	MIL0_TXBN		TRACE_DATA0
229			PA16		READY[2]			UART0_RX		TMR0_BRK	TMR2_CH1P		ARCO_OUT2N		MIL1_TXBP	MIL0_TXBP	MAC0_RXDV	TRACE_CLK
228	ETH_Ucc																	
227	ETH_RX-																	
226	ETH_RX+																	
225	ETH_TX-																	
224	ETH_TX+																	
223	ETH_GND																	
222	ETH_IREF																	
221	ETH_EXTR																	
220	HSE1_OSC_OUT																	
219	HSE1_OSC_IN																	
218	GND																	
217			PA17		READY[1]	READY[3]	CAN0_TX	UART1_TX		TMR1_ETR	TMR2_CH0N		ARCO_OUT2P		MIL1_ENB	MIL0_ENB	MAC0_RXD0	
216			PA18		READY[0]	READY[2]	CAN0_RX	UART1_RX		TMR1_CH0P	TMR2_CH0P		ARCO_IN6N		MIL1_RXBP	MIL0_RXBP	MAC0_RXD1	
215			PA19		DATA[63]					TMR1_CH0N	TMR2_ETR		ARCO_IN6P		MIL1_RXBN	MIL0_RXBN	MAC0_RXD2	TRACE_CLK
214			PA20		DATA[62]					TMR1_CH1P	TMR3_BRK		ARCO_IN7N		MIL0_TXAN	MIL1_TXAN	MAC0_RXD3	TRACE_DATA0
213	Ucc1																	
212			PA21		DATA[61]					TMR1_CH1N	TMR3_CH3N		ARCO_IN7P		MIL0_TXAP	MIL1_TXAP	MAC0_RXERR	TRACE_DATA1
211			PA22		DATA[60]					TMR1_CH2P	TMR3_CH3P		ARCO_OUT3N		MIL0_ENA	MIL1_ENA	MAC0_RXCLK	TRACE_DATA2
210			PA23		DATA[59]			UART0_TX	SSP0_TX	TMR1_CH2N	TMR3_CH2N		ARCO_OUT3P		MIL0_RXAP	MIL1_RXAP	MAC0_TXD0	TRACE_DATA3
209			PA24		DATA[58]			UART0_RX	SSP0_CLK	TMR1_CH3P	TMR3_CH2P		ARCO_IN0N	MIL0_RXAN	MIL1_RXAN	MAC0_TXD1		
208	GND																	
207	SPW_TXD-																	
206	SPW_TXD+																	
205	SPW_TXS-																	
204	SPW_TXS+																	
203	SPW_RXS-																	
202	SPW_RXS+																	
201	SPW_RXD-																	
200	SPW_RXD+																	
199	SPW_GND																	
198	GND																	
197	SPW_Ucc																	
196	SPW_EXTR																	
195	SPW_IREF																	
194	Ucc1																	
193	Ucc1																	

Рисунок 8 – Функции выводов 193 ÷ 256 в различных режимах

5 Указания по применению и эксплуатации

При ремонте аппаратуры и измерении параметров замену микросхем необходимо проводить только при отключенных источниках питания.

Инструмент для пайки (сварки) и монтажа не должен иметь потенциал, превышающий 0,3 В относительно шины "Общий".

Типовая схема включения микросхем приведена на рисунке 166.

Порядок подачи и снятия напряжения питания и входных сигналов на микросхемы:

- подача (включение микросхем) – общий, питание батарейного домена U_{CCB} , напряжение питания U_{CC} , U_{CC1} , входные сигналы или одновременно;
- снятие (выключение микросхем) – в обратном порядке или одновременно.

Металлизированные площадки основания корпуса А1 – А8, предназначенные для монтажа навесных элементов, электрически соединены с выводами питания 81, 103, 124, 213, 233.

Металлизированные площадки основания корпуса В1 – В8, предназначенные для монтажа навесных элементов, электрически соединены с выводами «Общий» 75, 97, 198, 218, 238.

Крышка корпуса электрически изолирована от выводов микросхемы.

Вывод 256 корпуса микросхемы электрически соединен с подложкой кристалла. Вывод 256 соединять с шиной «Общий».

Рекомендуется конденсаторы С7 – С14 емкостью 0,1 мкФ распаивать непосредственно на металлизированные площадки А1 – А8, В1 – В8 основания корпуса.

Неиспользуемые выводы, настроенные или имеющие по умолчанию функцию цифрового входа, должны быть доопределены до допустимых логических уровней напряжения через резисторы доопределения.

6 Архитектура микроконтроллера

6.1 Питание

6.1.1 Домены питания

Микроконтроллер имеет несколько доменов питания, описание которых приведено в таблице 3.

Таблица 3 – Домены питания

Обозначение	Допустимые значения, В		Описание
	Не менее	Не более	
VUcc	2,0	5,5	Батарейное питание микросхемы. Используется для питания батарейного домена при отсутствии основного питания Ucc. Питание изолировано от Ucc. Переключение между VUcc и Ucc происходит автоматически. Если в системе батарейное питание не требуется, вывод VUcc должен быть объединен с Ucc
Ucc	3,0	5,5	Основное питание микросхемы. Используется для питания портов ввода-вывода
Ucc1	2,2	5,5	Входное питание встроенных LDO регуляторов, формирующих питание для цифровой части. На вход питания Ucc1 может подаваться основное питание Ucc, либо питание ниже, чем Ucc, но не менее 2,2 В и не более Ucc
ETHUcc	3,0	5,5	Питание физических приемо-передатчиков Ethernet. Должно совпадать с питанием по уровню с Ucc±0,2 В, не выходя за допустимые значения. Используется для снижения шумов по Ucc
SPWUcc	3,0	5,5	Питание физических приемо-передатчиков SpaceWire. Должно совпадать с питанием по уровню с Ucc±0,2 В, не выходя за допустимые значения. Используется для снижения шумов по Ucc
ADC_Ucc	3,0	5,5	Основное питание аналоговых блоков АЦП. Должно совпадать с питанием по уровню с Ucc±0,2 В, не выходя за допустимые значения. Используется для снижения шумов по Ucc
DAC_Ucc	3,0	5,5	Основное питание аналоговых блоков ЦАП. Должно совпадать с питанием по уровню с Ucc±0,2 В, не выходя за допустимые значения. Используется для снижения шумов по Ucc
BDUcc	-	-	Питание батарейного домена. Выбирается из VUcc или Ucc (не требуется подавать извне): <ul style="list-style-type: none"> – при отсутствии Ucc выбирается VUcc; – при появлении Ucc > Urog спустя время Troresetn переключается на Ucc; – при снижении Ucc < Urog переключается на VUcc с минимальной задержкой
DUcc_B	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для питания батарейного домена. Входным напряжением LDO регулятора является BDUcc
DUcc_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для блока умножителей частоты и генераторов
DUcc0,1,2	1,62	1,98	Питание основной цифровой части, сформированное с помощью внутреннего LDO регулятора
HV	3,0	5,5	Высоковольтное напряжение программирования однократно программируемой памяти OTP. При обычной работе (не программирование) должно совпадать с питанием по уровню с Ucc±0,2 В. Но не выходя за допустимые значения
	6,5	7,5	Высоковольтное напряжение программирования однократно программируемой памяти OTP. При программировании памяти

6.1.2 Сброс при включении питания

Блок сброса (схема POR – Power-on-Reset) предназначен для выработки аппаратного сброса при включении основного питания или просадке его ниже критического уровня. При включении питания, пока уровень U_{CC} не превысил значение U_{POR} (типичное значение 2,75 В), сигнал сброса $U_{CC}RESET$ будет нулевым (сброс). После превышения U_{CC} уровня U_{POR} сигнал $U_{CC}RESET$ в течение времени $T_{PORRESETn}$ (типичное значение 10 мс) удерживается в нулевом состоянии, затем переключается в единицу. Задержка $T_{PORRESETn}$ формируется на счетчике, работающем с тактовой частотой от генератора LSI. Если по каким-либо причинам произошел отказ генератора LSI, схема сформирует сигнал $U_{CC}RESET$ с задержкой много меньшей, чем $T_{PORRESETn}$.

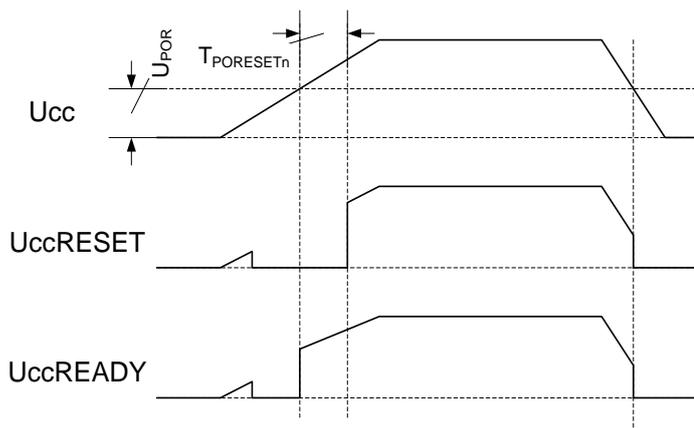


Рисунок 9 – Сигнал сброса при включении основного питания или просадке его ниже критического уровня

При снижении уровня U_{CC} ниже значения U_{POR} , сигнал $U_{CC}RESET$ вырабатывается с минимальной задержкой.

6.1.3 Схема защиты от перенапряжения POVR

При превышении напряжением питания U_{CC} уровня U_{OVER} (типичное значение 5,75 В) формируется сигнал сброса $OVRSTn$. Таким образом, при превышении напряжением питания предельно-допустимого уровня происходит аппаратный сброс.

Схема POVR может быть заблокирована сигналом OVREN. В этом случае, сигнал $OVRSTn$ всегда равен U_{CC} , Факт срабатывания схемы POVR фиксируется в регистре REG_60 батарейного домена и может быть проанализирован при последующем запуске системы.

6.1.4 Перекрытие рабочих диапазонов питания схемами POR и POVR

Чтобы исключить возможность продолжения работы схемы в диапазоне ниже или выше допустимого уровня напряжения питания, обеспечивается перекрытие рабочих диапазонов питания и уровней срабатывания схем POR и POVR. При отбракованном тестировании схемы POR и POVR отключаются, и тестирование выполняется на уровнях ниже нижней границы срабатывания схемы POR и выше верхней границы срабатывания схемы POVR. После чего проводится проверка работы схем POR и POVR, подтверждающая, что их области срабатывания укладываются в допустимые диапазоны. Таким образом, при любых технологических отклонениях либо произойдет сброс, либо микросхема будет работать в допустимых режимах использования. При этом нельзя рассчитывать на применение микросхемы в условиях отличных от предельно-допустимых.



Рисунок 10 – Рабочий диапазон питания

6.1.5 Схема переключения основного и батарейного питаний (Ucc и BUcc)

Сигнал UccRESET является ключом для переключения питания батарейного домена. Таким образом, при отсутствии основного питания Ucc батарейный домен запитан от вывода BUcc. При включении основного питания и превышении им уровня U_{POR} через время T_{PORSTN} питание батарейного домена переключится на основное питание. При снижении основного питания ниже уровня U_{POR} питание батарейного домена автоматически переключится на BUcc. Так как переключение с Ucc на BUcc происходит не мгновенно, возникает короткий момент времени, при котором на батарейный домен подается питание с уровнем ниже U_{POR} . Но диапазон допустимых питаний батарейного домена BUcc (не менее 2,0 и не более 5,5 В) перекрывает диапазон значений U_{POR} .

6.1.6 Блок монитора основного и батарейного питаний (Ucc и BUcc)

Для определения абсолютных значений уровней напряжения питания Ucc и BUcc используется блок PVD, обеспечивающий определение уровня напряжения с типовой точностью ± 100 мВ. Результат определения уровней напряжения отображается в блоке PWR_CNTR. При этом блок PWR_CNTR может быть настроен таким образом, чтобы при превышении или снижении некоторого заданного уровня напряжения вырабатывался сигнал прерывания для программной обработки данного события. При определении значения напряжения BUcc потребление осуществляется от источника BUcc, не зависимо от наличия или отсутствия Ucc. Таким образом, анализ BUcc желательно не делать непрерывным, а осуществлять периодически с коротко-временным включением/отключением анализа BUcc.

6.1.7 Управление питанием доменов DUcc

Микроконтроллер содержит несколько доменов DUcc (Таблица 4), которые включают в себя основные блоки микроконтроллера.

Таблица 4 – Домены питания DUcc

Название	Предельно-допустимые значения, В		Описание
	не менее	не более	
DUcc_B	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для батарейного домена
DUcc_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для блока умножителей частоты и генераторов
DUcc0,1,2	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для основной цифровой части (процессорное ядро, память, периферия)

Управление питанием доменов осуществляется в регистре батарейного домена. Также может быть отключено все питание DUcc, кроме батарейного домена, переводом схемы в режим Standby. Биты управления питанием реализованы в батарейном домене.

6.1.8 Блок монитора тока потребления доменов DUcc

В каждом блоке регулятора может быть включен блок датчика превышения тока. Датчик превышения тока вырабатывает сигнал CURFLG при превышении током уровня задаваемого входом CURDUCC[2:0]. Флаг CURFLG формируется только на время превышения тока. Блок датчика превышения тока разрешается битом CUREN. Кроме того, регулятор формирует выходной аналоговый сигнал CURLVL, обозначающий текущий уровень тока потребления. Данный сигнал заводится на отдельный канал АЦП и может быть преобразован в цифровое значение, соответствующее току потребления.

6.2 Тактовые частоты

Микросхема имеет несколько первичных источников тактирования. При включении питания автоматически запускаются генераторы LSI и HSI. Процессорное ядро начинает выполнять загрузочную программу, а затем и пользовательскую на частоте HSI. Переключение на другие источники тактирования выполняется программно в ходе выполнения пользовательской программой в соответствии с заложенным пользователем алгоритмом.

6.2.1 Встроенный низкоскоростной генератор LSI

При включении питания запускается встроенный RC-генератор LSI с частотой в диапазоне от 20 до 60 кГц. На основании этого генератора высчитывается задержка T_{PORSTN} .

Сам генератор и его управление расположены в батарейном домене. Генератор может продолжать работать в STANDBY режиме, в том числе и при наличии только питания BUcc.

6.2.2 Встроенный высокоскоростной генератор HSI

При включении питания запускается встроенный RC-генератор HSI с частотой в диапазоне от 6 до 10 МГц. На данной частоте запускается процессорное ядро.

6.2.3 Внешний высокоскоростной генератор HSE0 и HSE1

Для формирования внешней стабильной высокой частоты (от 1 до 30 МГц) применяются генераторы HSE0 и HSE1. Наличие двух генераторов обеспечивает возможность стабильной работы при отказе одного из генераторов. С помощью блока монитора тактовых частот может быть определен факт отказа одного из генераторов, выполнен аварийный переход на HSI-генератор и программное восстановление работы системы на оставшемся работающем генераторе.

Также генераторы HSE имеют встроенные фильтры помех, исключающие прохождение коротких импульсов или «просечек». Данный фильтр предназначен для исключения кратковременных единичных высокоскоростных импульсов, которые могут быть не обнаружены схемой монитора частоты, которая имеет большее время реакции.

Генератор HSE имеет флаг готовности, который вырабатывается при включении генератора и выхода его в нормальный режим работы.

6.2.4 Внешний низкоскоростной часовой генератор LSE

Для формирования внешней стабильной низкой частоты (от 20 до 40 кГц) применяется генератор LSE. В микроконтроллере нет второго дублирующего генератора LSE. Отказ генератора LSE может быть обнаружен с помощью блока монитора тактовых частот. При отказе генератора может быть выполнен аварийный переход на генератор HSI и программное восстановление работы системы. Также генератор LSE имеет встроенные фильтры помех, исключающие прохождение коротких импульсов или «просечек». Генератор LSE имеет флаг готовности, который вырабатывается при включении генератора и выхода его в нормальный режим работы. Генератор может продолжать работать в STANDBY режиме, в том числе и при наличии только питания BUcc.

6.2.5 Блок умножения тактовой частоты PLL0, PLL1 и PLL2

Для формирования высоких тактовых частот (до 150 МГц) используются схемы умножения тактовых частот. В качестве источников опорной частоты для PLL выступают генераторы HSI, HSE0, HSE1 и их производные. С помощью PLL формируется частота $F_{OUT} = F_{IN} * N / ((Q+1) * (DV+1))$ (подробнее см. рисунок). PLL имеет флаг готовности, который вырабатывается при включении схемы умножителя и выходе её в нормальный режим работы, либо при перезапуске PLL с новыми коэффициентами умножения частоты. С помощью блока монитора частоты, можно определить значение частоты по отношению к другим источникам тактирования, обнаружить отказ PLL (снижение или отсутствие частоты, и превышение выходной частоты), выполнить аварийный переход на генератор HSI и программно восстановить работу системы на других PLL или генераторах.

6.2.6 Монитор частоты внешних генераторов и PLL (в блоке CLKCNTNTR)

Блок монитора частоты внешних генераторов и PLL предназначен для определения значения тактовых частот по отношению друг к другу, определения следующих событий сбоя в работе генераторов и блоков PLL:

- Снижение частоты ниже предела;
- Пропадание частоты (снижение ниже более жесткого предела);
- Превышение частоты выше предела.

Монитор, в зависимости от настроек, может по-разному реагировать на различные события в тактовых частотах:

- Вырабатывать сигнал предупреждения;
- Аппаратно переключать схему тактирования на другой источник тактирования;
- Осуществлять сброс микроконтроллера через блок FT_CNTR.

Определение соотношения частот между собой происходит путем сравнения значений счетчиков, считающих на разных частотах. Сравнивая это соотношение с различными пределами, определяется сбой. Для обнаружения факта сбоя, в зависимости от реального соотношения частот и заданных пределов их допустимого соотношения, может потребоваться некоторое время. Таким образом, при возникновении сбоя (EVENT) или отказа (ERROR) блоку монитора потребуется некоторое время для реакции на это событие. Для тактовой частоты MAX_CLK также реализован блок монитора частоты, но он не имеет возможности аппаратного переключения на частоту HSI, и позволяет только обнаруживать факты сбоя или отказа.

6.3 Сигналы Сброса

Структурная схема формирования сигналов сброса представлена на рисунке .

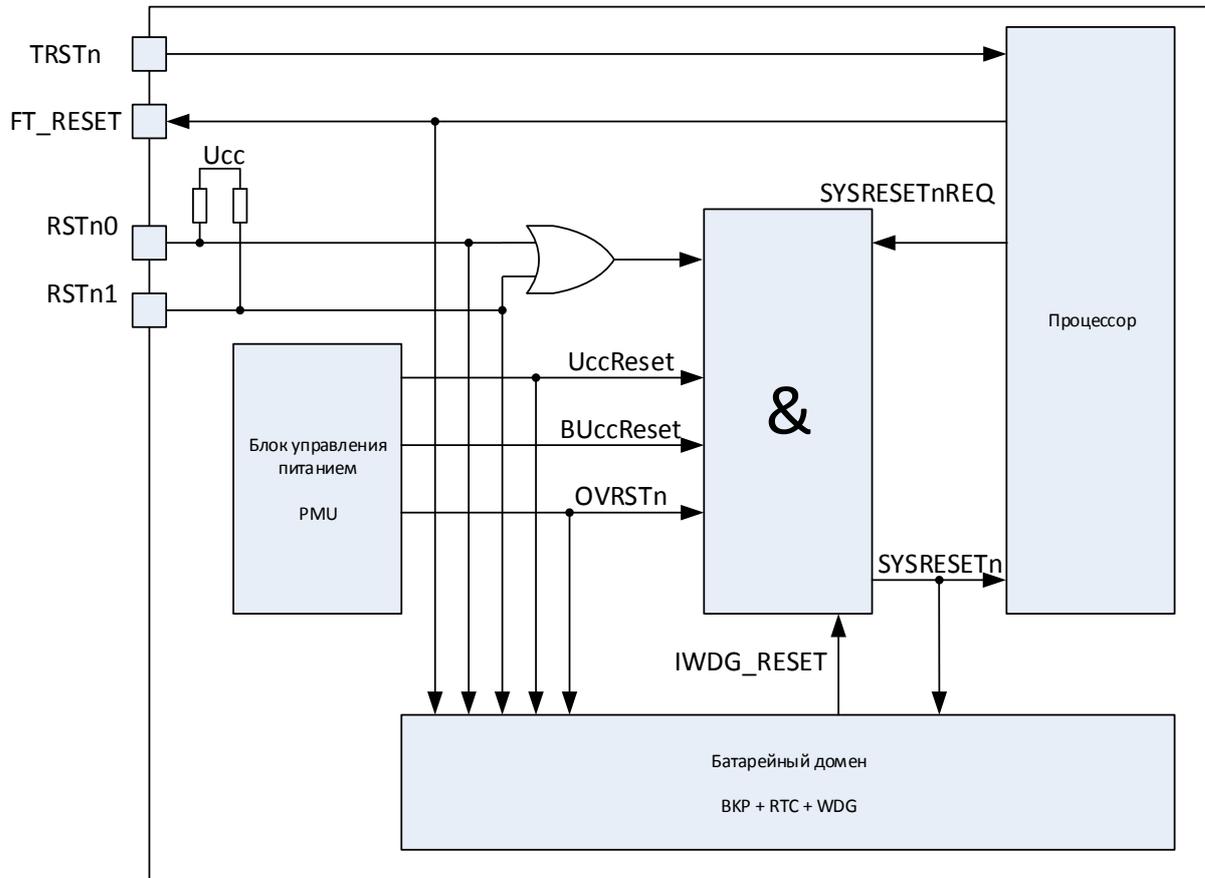


Рисунок 11 – Формирование сигналов сброса

6.3.1 Сигналы внешнего сброса RSTn0 и RSTn1

Микросхема может быть сброшена внешними сигналами с выводов RSTn0 и RSTn1. Сброс осуществляется при одновременной подаче низкого уровня на оба входа. Входы RSTn0 и RSTn1 имеет встроенную подтяжку к питанию с эквивалентным сопротивлением 50 кОм. Факт появления каждого сигнала RSTn0 и RSTn1 фиксируется в батарейном домене в битах S_RESET0n и S_RESET1n.

6.3.2 Сигнал сброса по питанию UccRESET

Начиная с включения питания Ucc и до превышения уровня U_{POR} в течение времени T_{PORESETn}, а также при снижении Ucc ниже уровня U_{POR} вырабатывается сигнал сброса UccRESET. Этот сигнал сбрасывает микросхему в исходное состояние, за исключением режима STANDBY. В этом режиме микросхема может быть сброшена только сбросом BDUccRESET или сигналами пробуждения из режима STANDBY (сигнал WAKEUP или флаг ALRF будильника часов реального времени). Факт появления сигнала UccRESET фиксируется в батарейном домене в бите S_UccRESET.

6.3.3 Сигнал сброса по питанию DUccRESET

Домены цифрового питания DUcc сбрасываются сигналом готовности DUCCREADY при отключении регуляторов. При выключении регуляторов сигнал DUCCREADY доопределяет в ноль сигнал SYSRESETn. Факт появления сигнала DUccRESET фиксируется в батарейном домене в бите S_SYSRSTn.

6.3.4 Сигнал программного запроса сброса SYSRESETnREQ

Микросхема может быть сброшена программным запросом со стороны процессорного ядра. Факт появления сигнала SYSRESETnREQ фиксируется в батарейном домене в бите S_SYSRSTn.

6.3.5 Сигналы аварийного сброса FT_RESET

Микросхема может быть сброшена блоком управления сбоями при обнаружении различных событий, также ряд событий всегда вызывает сброс микросхемы. Для этого необходимо вывод FT_RESET соединить с выводами внешнего сброса RSTn0 и RSTn1 вне микросхемы. При этом факт появления сигнала FT_RESETn фиксируется в батарейном домене в бите S_FT_RESET.

6.3.6 Сигналы сброса сторожевого таймера IWDG_RESET

Микросхема может быть сброшена сторожевым таймером WDG. Факт появления сигнала IWDG_RESET фиксируется в батарейном домене в бите S_WDG_RESET.

6.3.7 Сигнал сброса по перенапряжению POVR

При превышении питанием U_{CC} уровня U_{OVER} (не менее 5,5 В и не более 5,7 В) формируется сигнал сброса OVRSTn. Таким образом, при превышении напряжением питания предельно-допустимых границ микроконтроллер может быть аппаратно сброшен.

Схема POVR при проведении отбраковочного тестирования может быть заблокирована сигналом EXTPOVR. В этом случае сигнал OVRSTn всегда равен U_{CC}, что позволяет протестировать схему при питании выше уровня U_{OVER}. Отключать схему POVR в рабочем режиме не рекомендуется.

Факт срабатывания схемы POVR фиксируется в регистре батарейного домена в бите S_OVRSTn и может быть проанализирован при последующем запуске системы.

6.3.8 Сигнал сброса отладочного интерфейса TRSTn

Сигнал TRSTn служит для сброса отладочного интерфейса JTAG. Если отладочный интерфейс не используется данный вывод должен быть подтянут к логической единице внешним резистором. Установка в первоначальное значение отладочного интерфейса так же происходит по сигналу U_{CC}RESET.

6.4 Режимы работы микроконтроллера

Выбор режима запуска микроконтроллера определяется при включении микроконтроллера или после любого сигнала сброса. Режим запуска задается комбинацией сигналов на входах MODE[7:0]. При этом режим закодирован кодом Хемминга (7,4), таким образом, биты MODE[3:0] задают режим, а биты MODE[7:4] задают проверочные ЕСС биты. При выборе режима запуска автоматически исправляются одиночные ошибки в комбинации режима запуска, при обнаружении двойных ошибок микроконтроллер остается в режиме ожидания в загрузочной программе. Для определения режима запуска микроконтроллер конфигурирует выводы MODE[7:0] (PE[31:24]) в цифровой режим на вход, включает внутренние резисторы подтяжки ~50 кОм на землю. После чего выдерживается пауза в 300 мкс для переопределения потенциалов на выводе MODE, и выполняется тройное считывание значений выводов MODE для выполнения мажоритарного контроля. Затем выполняется проверка ЕСС смажорированного значения MODE. И результирующее значение записывается в биты MODE регистров ВКР_REG_60_TMRx. Проводится контроль записанного значения (отсутствие разницы в регистрах TMR0, 1, 2 батарейного домена), и на основании конечного значения MODE осуществляется переход в выбранный режим. При обнаружении неисправимой ошибки микроконтроллер остается в режиме ожидания в загрузочной программе.

Таблица 5 – Режимы работы микроконтроллера

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
0000	0000	WAIT_BOOT_JA	Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_A
0111	0001	ОТР+JB	Запуск из ОТР (0x0100_0000) с включенным интерфейсом отладки через выводы JTAG_B
1011	0010	ОТР+JA	Запуск из ОТР (0x0100_0000) с включенным интерфейсом отладки через выводы JTAG_A
1100	0011	EXTBUS_8_ECC+JB	Запуск из внешней памяти с (0x1000_0000) сконфигурированной в минимальный режим: <ul style="list-style-type: none"> – Шина данных D[7:0] (PE[5:0]+PD[31:30]) – Шина адреса A[15:0] (PD[13:0]+PC[31:30]) – Сигнал nOE (PD[23]) с последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_B
1101	0100	EXTBUS_8_ECC+JA	Запуск из внешней памяти с (0x1000_0000) сконфигурированной в минимальный режим: <ul style="list-style-type: none"> – Шина данных D[7:0] (PE[5:0]+PD[31:30]) – Шина адреса A[15:0] (PD[13:0]+PC[31:30]) – Сигнал nOE (PD[23]) с последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_A
1010	0101	EXTBUS_CFG+JB	Запуск из внешней памяти с (0x1000_0000) с чтением конфигурации в режиме: <ul style="list-style-type: none"> – Шина данных D[7:0] (PE[5:0]+PD[31:30]) – Шина адреса A[10:3] (PD[8:1]) – Сигнал nOE (PD[23]) Читается DATA[7:0] = ECC[3:0]+CFGx[3:0], где: <ul style="list-style-type: none"> – CFG0 = Режим запуска (8, 16, 32, 64) – CFG1 = Режим ECC (нет, послед., параллельн.) – CFG2 = BASE_ECC[3:0] – CFG3 = BASE_ECC[7:4] ... – CFG9 = BASE_ECC[31:28] с включенным интерфейсом отладки через выводы JTAG_B
0110	0110	EXTBUS_CFG+JA	Запуск из внешней памяти с (0x1000_0000) с чтением конфигурации в режиме: <ul style="list-style-type: none"> – Шина данных D[7:0] (PE[5:0]+PD[31:30]) – Шина адреса A[10:3] (PD[8:1]) – Сигнал nOE (PD[23]) Читается DATA[7:0] = ECC[3:0]+CFGx[3:0], где: <ul style="list-style-type: none"> – CFG0 = Режим запуска (8, 16, 32, 64) – CFG1 = Режим ECC (нет, послед., параллельн.) – CFG2 = BASE_ECC[3:0] – CFG3 = BASE_ECC[7:4] ... – CFG9 = BASE_ECC[31:28] с включенным интерфейсом отладки через выводы JTAG_A

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
0001	0111	SPI0+JB	Загрузка последовательно из внешней памяти по SPI интерфейсу с включенным интерфейсом отладки через выводы JTAG_B
1110	1000	SPI1+JA	Загрузка последовательно из внешней памяти по SPI интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
1001	1001	SPI2+JB	Загрузка последовательно из внешней памяти по SPI интерфейсу с включенным интерфейсом отладки через выводы JTAG_B
0101	1010	SPI3+JA	Загрузка последовательно из внешней памяти по SPI интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
0010	1011	UART0+JB	Загрузка последовательно из внешней памяти по UART интерфейсу с включенным интерфейсом отладки через выводы JTAG_B
0011	1100	UART0+JA	Загрузка последовательно из внешней памяти по UART интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
0100	1101	ETT_K7_TEST	Тест предназначен для кратковременных и длительных испытаний на безотказность.
1000	1110	-	Резерв
1111	1111	TEST_MODE+JB	Тестовый режим микросхемы для отбраковочного тестирования с включенным интерфейсом отладки через выводы JTAG_B
-	Двойная ошибка	WAIT_BOOT_JB	Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_B

Выбор выводов для отладки определяется значением в троированных регистрах батарейного домена REG_60_TMRx битами MODE[0] и битом DISABLE_JTAG. При выборе того или иного интерфейса данные выводы жестко переопределяются на выполнение функции JTAG.

6.4.1 Режим OTP+JA

После определения данного режима бутовая программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x0100_0000 (SP);
- устанавливает адрес таблицы векторов в значение 0x0100_0000 (VTOR);
- осуществляет безусловный переход по адресу, считанному из ячейки 0x0100_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

В версии микроконтроллера с памятью ОЗУ вместо OTP предполагается, что ОЗУ памяти программ проинициализировано кодом заранее.

Отладка выполняется через выводы JTAG_A.

6.4.2 Режим OTP+JB

Аналогично OTP_JA, за исключением того, что выполняется отладка через выводы JTAG_B.

6.4.3 Режим EXTBUS_8_ECC+JA

После определения данного режима бутовая программа конфигурирует внешнюю системную шину в режим:

- Шина данных D[7:0] (PE[5:0]+PD[31:30]);
- Шина адреса A[15:0] (PD[13:0]+PC[31:30]);
- Сигнал nOE (PD[23]).

С последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_A.

После определения данного режима бутовая программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);
- устанавливает адрес таблицы векторов в значение 0x1000_0000 (VTOR);
- осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

6.4.4 Режим EXTBUS_8_ECC+JB

Аналогично режиму EXTBUS_8_ECC+JA, за исключением того, что отладка выполняется через выводы JTAG_B.

6.4.5 Режим EXTBUS_CFG+JA

После определения данного режима бутовая программа конфигурирует внешнюю системную шину в режим:

- Шина данных D[7:0] (PE[5:0]+PD[31:30]);
- Шина адреса A[10:3] (PD[8:1]);
- Сигнал nOE (PD[23]).

С включенным интерфейсом отладки через выводы JTAG_A.

После этого из ячеек по адресам 0x1000_0400, 0x1000_0408, ... 0x1000_0448 читается DATA[7:0] = ECC[3:0]+CFGx[3:0], и выполняется контроль ECC. При обнаружении одиночной ошибки бутовая программа исправляет её, при обнаружении двойной ошибки – формирует сигнал сброса.

После проверки, бутовая программа конфигурирует внешнюю системную шину в зависимости от CFGx:

- Шина данных (в зависимости от CFG0):
 - CFG0 = 1 – 8-ми битная шина;
 - CFG0 = 2 – 16-ти битная шина;
 - CFG0 = 3 – 32-х битная шина;
 - CFG0 = 4 – 64-х битная шина.
- Шина адреса A[20:0] (PD[18:0]+PC[31:30])
- Сигнал nOE (PD[23])
- Контроль ECC (в зависимости от CFG1):
 - CFG1 = 1 – без ECC;
 - CFG1 = 2 – с последовательной ECC. Базовый адрес ECC устанавливается из CFG9- CFG2:
 - CFG2 = BASE_ECC[3:0];
 - CFG3 = BASE_ECC[7:4];
 - ...
 - CFG9 = BASE_ECC[31:28].
 - CFG1 = 3 – с параллельной ECC.

С включенным интерфейсом отладки через выводы JTAG_A.

После переопределения внешней системной шины бутовая программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);
- устанавливает адрес таблицы векторов в значение 0x1000_0000 (VTOR);
- осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

6.4.6 Режим EXTBUS_CFG+JB

Аналогично режиму EXTBUS_8_ECC+JA, за исключением того, что выполняется отладка через выводы JTAG_V.

6.4.7 Режим SPI0+JB

После определения данного режима бутовая программа конфигурирует контроллер SSP0:

- PA [7] – nSTATUS
- PA [8] – DCLK
- PA [9] – CONF_DONE
- PA [10] – DATA

и отладкой через выводы JTAG_V. В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

Режим SPI0+JB необходим для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение.

В качестве источника тактовой частоты SSP0 используется внутренний RC-генератор HSI с частотой 8 МГц.

6.4.7.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

- Частота сигнала DCLK – 100 кГц;
- Количество бит данных – 8;
- Формат синхронного обмена – SPI фирмы Motorola, SPO = 0, SPH = 0.

6.4.7.2 Протокол обмена по SPI

После конфигурирования выводов бутовая программа ожидает высокого уровня на выводе nSTATUS, что свидетельствует о готовности к работе внешней EEPROM. После обнаружения высокого логического уровня из EEPROM последовательно считывается информационный заголовок:

Информационный заголовок				
32 бит	32 бит	32 бит	16 бит	16 бит
OFFS*	OFFS + 4	OFFS + 8	OFFS + 12	OFFS + 14
SRAM ADDR	START ADDR	PROG LEN	CRC CNT LEN	CRC 16

*В некоторых EEPROM начальные адреса зарезервированы для конфигурации.

- 32-х разрядный адрес (SRAM ADDR), по которому будет расположен образ копируемой программы.
- 32-х разрядный адрес (START ADDR), которому будет передано управление после копирования программы.
- 32-х разрядная длина (PROG LEN) образа программы в словах (32 бит) без учета CRC16.
- 16-ти разрядная длина данных в словах (CRC CNT LEN), по которым вычисляется CRC16 в образе программы.

- CRC16 информационного заголовка без учета поля CRC16.

После чтения информационного заголовка верифицируются считанное и рассчитанное значение CRC16. При несовпадении формируется сигнал сброса. После успешной проверки CRC16 бутовая программа последовательно считывает образ программы:

Образ программы			
CRC CNT LEN *(32 бит)	16 бит	CRC CNT LEN *(32 бит)	16 бит
OFFS+16			
DATA	CRC16	DATA	CRC16

- CRC CNT LEN-слов данных программы и записывает их, начиная с адреса образа программы (SRAM ADDR).
- CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).
- CRC CNT LEN-слов данных программы и записывает их.
- CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).
- и т.д., пока внутренний счетчик не станет равным PROG LEN.

В данном случае, образ программы соответствует ситуации, когда половина данных программы участвует в подсчете CRC16 (CRC CNT LEN = PROG LEN / 2). Таким образом, в зависимости от CRC CNT LEN устанавливается необходимая избыточность информационных данных (CRC16).

Все данные EEPROM имеют порядок байт "little-endian", все байты имеют обратный порядок бит.

6.4.7.3 Вычисление CRC16

Для вычисления CRC16 используется полином вида: $x^{16} + x^{15} + x^2 + 1$ (0xA001) с начальным значением CRC16 0xFFFF:

```
uint16_t crc16_update (uint16_t crc, uint8_t a)
{
    int i;

    crc ^= a;
    for (i = 0; i < 8; ++i)
    {
        if (crc & 1)
            crc = (crc >> 1) ^ 0xA001;
        else
            crc = (crc >> 1);
    }
    return crc;
}
```

6.4.8 Режим SPI1+JA

Аналогично режиму SPI0+JB, за исключением того, что выполняется отладка через выводы JTAG_A.

6.4.9 Режим SPI2+JB

После определения данного режима бутовая программа конфигурирует контроллер SSP0:

- PA [7] – TX
- PA [8] – CLK
- PA [9] – FS
- PA [10] – RX

и отладкой через выводы JTAG_V. В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

Данный режим необходим для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение.

В качестве источника тактовой частоты SSP0 используется внутренний RC-генератор HSI с частотой 8 МГц.

6.4.9.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

- Частота сигнала CLK – 100 кГц;
- Количество бит данных – 8;
- Формат синхронного обмена – SPI фирмы Motorola, SPO = 0, SPH = 0.

6.4.9.2 Протокол обмена по SPI

После конфигурирования выводов бутовая программа из внешней EEPROM по адресу 0x000_000 считывает 32-х разрядный адрес смещения информационного заголовка OFFS. Если смещение отлично от нуля, то операция чтения прерывается и возобновляется по адресу OFFS. Далее последовательно считывается информационный заголовок:

Информационный заголовок					
32 бит	32 бит	32 бит	32 бит	16 бит	16 бит
0x000_000	OFFS	OFFS + 4	OFFS + 8	OFFS + 12	OFFS + 14
OFFS	SRAM ADDR	START ADDR	PROG LEN	CRC CNT LEN	CRC 16

- 32-х разрядный адрес (SRAM ADDR), по которому будет расположен образ копируемой программы.
- 32-х разрядный адрес (START ADDR), которому будет передано управление после копирования программы.
- 32-х разрядная длина (PROG LEN) образа программы в словах (32 бит) без учета CRC16.
- 16-ти разрядная длина данных в словах (CRC CNT LEN), по которым вычисляется CRC16 в образе программы.
- CRC16 информационного заголовка, включая смещение информационного заголовка (OFFS), без учета поля CRC16.

После чтения информационного заголовка верифицируются считанное и рассчитанное значение CRC16. При несовпадении формируется сигнал сброса. После успешной проверки CRC16 бутовая программа последовательно считывает образ программы:

Образ программы			
CRC CNT LEN *(32 бит)	16 бит	CRC CNT LEN *(32 бит)	16 бит
OFFS+16			
DATA	CRC16	DATA	CRC16

- CRC CNT LEN-слов данных программы и записывает их, начиная с адреса образа программы (SRAM_ADDR).
- CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).
- CRC CNT LEN-слов данных программы и записывает их.
- CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).
- И т.д., пока внутренний счетчик не станет равным PROG_LEN.

В данном случае образ программы соответствует ситуации, когда половина данных программы участвует в подсчете CRC16 ($CRC\ CNT\ LEN = PROG\ LEN / 2$). Таким образом, в зависимости от CRC CNT LEN устанавливается необходимая избыточность информационных данных (CRC16).

Все данные EEPROM имеют порядок байт "little-endian".

6.4.9.3 Вычисление CRC16

Для вычисления CRC16 используется полином вида: $x^{16} + x^{15} + x^2 + 1$ (0xA001) и начальным значением CRC16 0xFFFF:

```
uint16_t crc16_update (uint16_t crc, uint8_t a)
{
    int i;

    crc ^= a;
    for (i = 0; i < 8; ++i)
    {
        if (crc & 1)
            crc = (crc >> 1) ^ 0xA001;
        else
            crc = (crc >> 1);
    }
    return crc;
}
```

6.4.10 Режим SPI3+JA

Аналогично режиму SPI2+JB за исключением того, что выполняется отладка через выводы JTAG_A.

6.4.11 Режим UART0+JA

После определения данного режима бутовая программа конфигурирует контроллер UART0 TX(PE[15]), RX(PE[16]) с отладкой через выводы JTAG_A.

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении бутовой программы.

Данный режимы предоставляет достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение. Кроме того, существует возможность задания внешним устройством

скорости обмена. Помимо доступа к ОЗУ может быть осуществлен доступ и к другим адресным диапазонам.

В качестве источника тактовой частоты UART0 используется внутренний RC-генератор HSI с частотой 8 МГц. Так как имеется разброс значений частоты HSI, то требуется этап подбора значения делителя частоты UART0 для синхронизации с внешним устройством.

6.4.11.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- Начальная скорость – 9600 бод;
- Количество бит данных – 8;
- Четность – нет;
- Количество Stop бит – 1;
- Загрузчик не использует FIFO UART0;
- Загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master;
- Данные передаются младшим битом вперед.

6.4.11.2 Протокол обмена по UART

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, устройству Master доступны следующие команды:

6.4.11.3 Команды UART загрузчика

Таблица 6 – Список команд UART загрузчика

Команда	Код	ASCII Символ	Описание
CMD_SYNC	0x00		Пустая команда. Загрузчик ее принимает, но ничего по ней не делает
CMD_CR	0x0D		Выдача приглашения устройству Master
CMD_BAUD	0x42	'B'	Установка скорости обмена
CMD_LOAD	0x4C	'L'	Загрузка массива байт
CMD_VFY	0x59	'Y'	Выдача массива байт
CMD_RUN	0x52	'R'	Запуск программы на выполнение

6.4.11.4 Команда CMD_SYNC

Пустая команда.

Загрузчик (Slave) ее принимает, но ничего по ней не делает. Код команды соответствует символу синхронизации.

Таблица 7 – Команда CMD_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master Выдает код команды CMD_SYNC	Slave Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды

6.4.11.5 Команда CMD_CR

Выдача приглашения устройству Master.

Таблица 8 – Команда CMD_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master Выдает код команды CMD_CR	Slave Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды. Выдает код команды CMD_CR. Выдает код 0x0A Выдает код 0x3E (ASCII символ '>')

6.4.11.6 Команда CMD_BAUD

Установка скорости обмена.

Таблица 9 – Команда CMD_BAUD

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод]
Формат команды:	
Master Выдает код команды CMD_BAUD	Slave Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master Выдает параметр	Если параметр принят с ошибками, выдает код ошибки ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Выдает код команды CMD_BAUD. Устанавливает новое значение скорости обмена

6.4.11.7 Команда CMD_LOAD

Загрузка массива байт в память микроконтроллера.

Таблица 10 – Команда CMD_LOAD

Код команды	CMD_LOAD = 0x4C
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1	Адрес памяти приемника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master Выдает код команды CMD_LOAD	Slave Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master Выдает параметр 1	Slave Если хотя бы один из параметров принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master Выдает параметр 2	Slave Если хотя бы один из параметров принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_LOAD
Master Выдает массив байт младшим байтом вперед	Slave Принимает массив байт. Если хотя бы один байт принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании приёма массива выдает код ответа REPLY_OK = 0x4B ('K')

6.4.11.8 Команда CMD_VFY

Выдача массива байт из памяти микроконтроллера.

Таблица 11 – Команда CMD_VFY

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master Выдает код команды CMD_VFY	Slave Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master Выдает параметр 1	Slave Если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master Выдает параметр 2	Slave Если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_VFY. Выдает массив байт младшим байтом вперед. По окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

6.4.11.9 Команда CMD_RUN

Запуск программы на выполнение.

Таблица 12 – Команда CMD_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес таблицы векторов загруженной программы
Формат команды:	
Master Выдает код команды CMD_RUN	Slave Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Выдает параметр	Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_RUN. Устанавливает значение MSP и PC согласно таблице векторов (NVIC не перепрограммируется) и, таким образом, Slave завершает свое выполнение

6.4.11.10 Синхронизация с внешним устройством

Начальные условия.

На этапе синхронизации с внешним устройством (Master) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость, он переходит в диспетчер команд и выдает приглашение (3 байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>')) Master-у.

Master завершает выдачу синхросимвол и теперь может подавать команды согласно протоколу обмена.

6.4.11.11 Прием параметров команды

Параметры команд – это 4-х байтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

Если при приеме параметра обнаружена аппаратная ошибка (UART установил в '1' какой-либо из флагов ошибки), то прием параметров не прекращается.

Анализ всех видов ошибок, связанных с передачей параметров, загрузчик производит только после принятия всех параметров команды.

6.4.11.12 Сообщения об ошибках

Сообщения об ошибках – это 2-х байтные последовательности символов. Первый символ всегда 0x45 ('E'). Второй символ определяет тип ошибки.

После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения должен прекратить передачу байт, относящихся к текущей команде.

После принятия сообщения об ошибке Master должен подавать команду CMD_CR до тех пор, пока не получит корректный ответ, соответствующий этой команде.

Возможны следующие сообщения об ошибках: ERR_CHN, ERR_CMD, ERR_BAUD.

6.4.11.13 Ошибка ERR_CHN

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

Выдается, если UART установил в '1' один из аппаратных флагов ошибки при приеме очередного байта.

6.4.11.14 Ошибка ERR_CMD

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

6.4.11.15 Ошибка ERR_BAUD

Принята неизвестная команда.

Код ошибки 0x62 ('b').

Выдается диспетчером команд, если по принятому от устройства Master значению скорости обмена невозможно вычислить корректное значение делителя частоты UART.

6.4.12 Режим UART0+JB

Аналогично режиму UART0+JA, за исключением того, что используются выводы TX(PA[7]), RX(PA[8]) и выполняется отладка через выводы JTAG_B.

6.4.13 Режим TEST_MODE+JB

В данном режиме на выводы JTAG_B вместо TAP контроллера процессорных ядер подключится технологический TAP контроллер, позволяющий перевести микросхему в тестовые режимы.

6.4.14 Режим WAIT_BOOT_JA

Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_A.

6.4.15 Режим WAIT_BOOT_JB

Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_V.

6.4.16 Статус загрузчика

Последние четыре слова в ОЗУ используются для передачи ошибок, которые могли возникнуть в процессе загрузки контроллера. Таким образом, после завершения работы бутовой программы и передачи управления, ошибки могут быть считаны из ОЗУ для корректирования режима и минимизации ошибок при последующих запусках.

Таблица 13 – Поля статуса загрузчика

Адрес ОЗУ	Размер, бит	Обозначение	Описание
0x2000_7FF0	8	INIT_ERR_CNTR	Число повторных инициализаций выводов MODE[7:0]. Значение INIT_ERR_CNTR >2 свидетельствует об ошибке в контроллере порта.
0x2000_7FF1	8	MODE_0	Считанное значение MODE[0]
0x2000_7FF2	8	MODE_1	Считанное значение MODE[1]
0x2000_7FF3	8	MODE_2	Считанное значение MODE[2]
0x2000_7FF4	8	MODE_MAJ	Значение MODE после мажоритарного контроля MODE[0], MODE[1] и MODE[2]
0x2000_7FF5	8	MODE_MAJ_ERR	Ошибка мажоритарного контроля: 0 – Нет различия между MODE[0], MODE[1] и MODE[2] 1 – Есть различия между MODE[0], MODE[1] и MODE[2]
0x2000_7FF6	8	SYNDROM	Синдром: 1 – Ошибка в MODE [5] 2 – Ошибка в MODE [6] 3 – Ошибка в MODE [0] 4 – Ошибка в MODE [7] 5 – Ошибка в MODE [1] 6 – Ошибка в MODE [2] 7 – Ошибка в MODE [3]
0x2000_7FF7	8	PARITY_BIT	Четность MODE
0x2000_7FF8	8	ECC_ERR_TYPE	Тип ошибки при контроле ECC: 0 – Нет ошибки 1 – Одиночная ошибка 2 – Двойная ошибка 3 – Другие ошибки нечетной кратности
0x2000_7FF9	8	MODE_CORR	Скорректированное значение MODE. При обнаружении двойных или других ошибок нечетной кратности, MODE_CORR совпадает с MODE_MAJ
0x2000_7FFA	8	MODE_EXTBUS	Исправленное значение конфигурации (ECC+CFG0) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA
0x2000_7FFB	8	MODE_ECC_EXTBUS	Исправленное значение конфигурации (ECC+CFG1) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA
0x2000_7FFC	32	ECC_BASE_EXTBUS	Исправленное значение базового адреса ECC CFG(9-2) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA

6.5 Организация памяти

Микроконтроллер имеет Фон-Неймановскую архитектуру памяти, т.е. все адресное пространство (память программ и память данных) отображены в общее адресное пространство размером 4 Гбайт. Но при этом память разделена на две секции: CODESECTION (от 0x00000000 до 0x1FFFFFFF) для кода программ и DATASECTION (от 0x20000000 до 0xDFFFFFFF) для данных. Доступ к CODESECTION осуществляется через шины IBUS и DBUS. Доступ к DATASECTION осуществляется через шину SBUS. Доступ DMA контроллера осуществляется через шину MBUS и может быть выполнен в диапазоне DATASECTION. При этом код программы может располагаться в DATASECTION, но при его выполнении снижается производительность из-за возникающих конфликтов по доступу за инструкциями и данными. Данные могут располагаться в CODESECTION в областях RAMD или Внешней памяти ОЗУ, отображаемой в эту секцию.

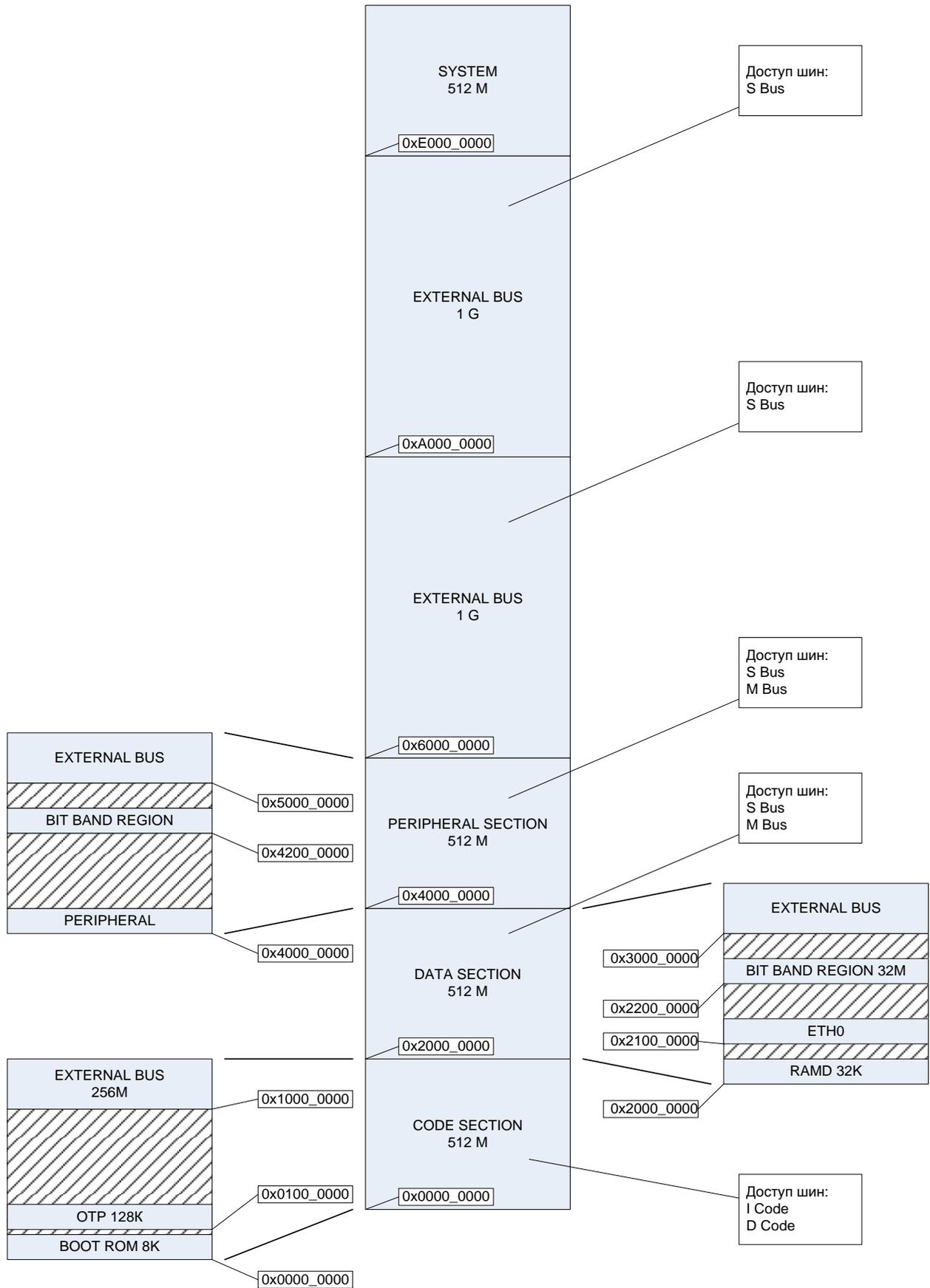


Рисунок 12 – Распределение адресного пространства

Возможная коммутация агентов в блоке BUSMATRIX представлена на рисунке .

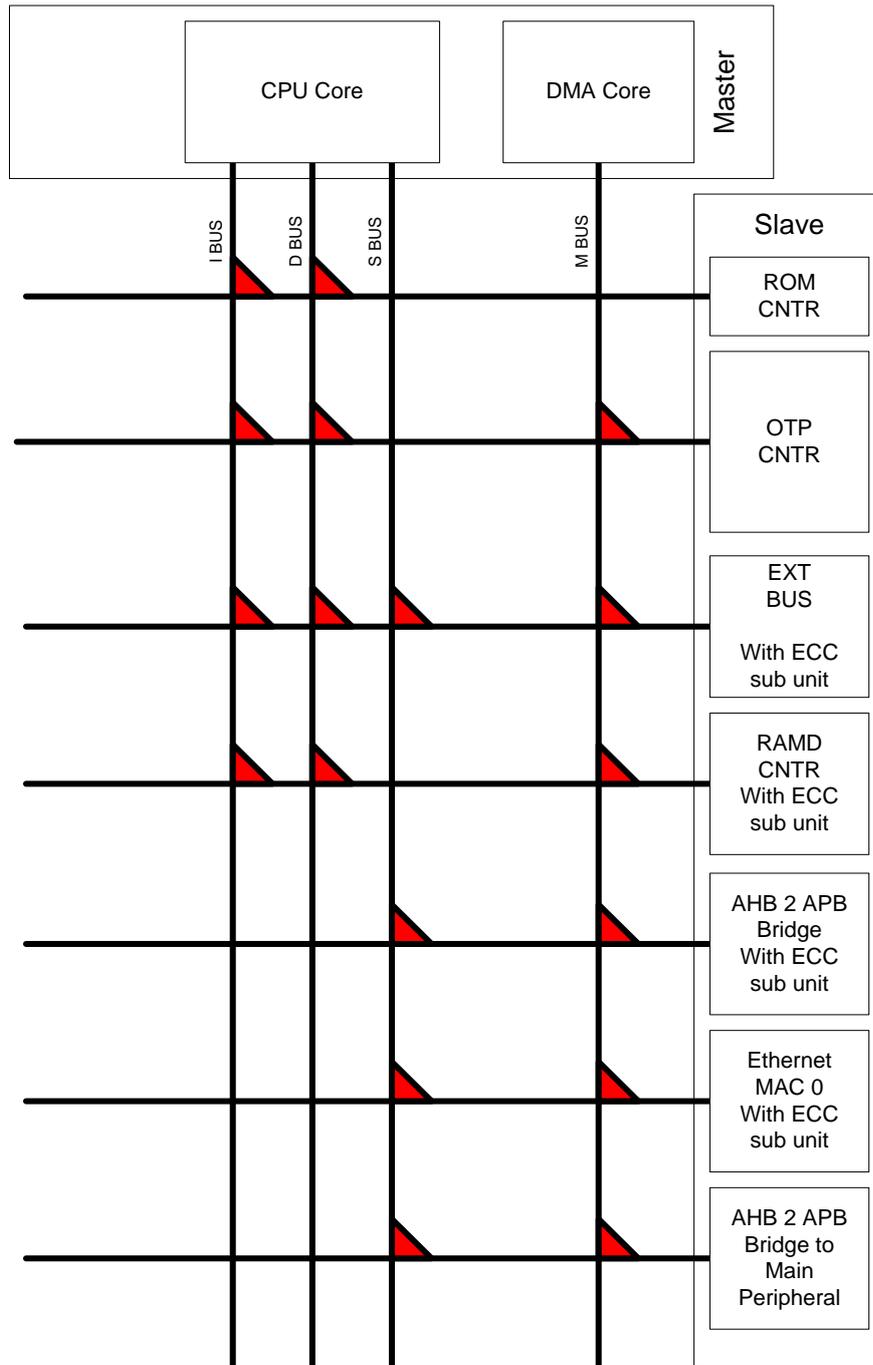


Рисунок 13 – Схема внутренних межсоединений

6.5.1 Распределение областей памяти

Таблица 14 – Распределение адресного пространства

Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
0x0000_0000	8K	BOOT ROM	I, D	Область памяти загрузочного ПЗУ. Из этой памяти микроконтроллер начинает работу, определяет дальнейший режим функционирования и выполняет необходимые настройки
0x0100_0000	128K	OTP	I, D	Однократно программируемая память программ, выполненная по технологии anti-fuse. Основная память программ для пользовательской программы.
0x0200_0000	32K	-	-	-
0x0300_0000	128M			
0x1000_0000	256M	EXT BUS	I, D, S, M	Область памяти для отображения внешней системной шины
0x2000_0000	32K	RAMD	S, M	Статическое ОЗУ в DATASECTION, предназначено для данных
0x2100_0000		E0	S, M	Регистры блока контроллера ETHERNETMAC 0
0x2100_8000		E0	S, M	Статическое ОЗУ блока контроллера ETHERNETMAC 0
0x2101_0000		-	-	-
0x2101_8000		-	-	-
0x2102_0000				
0x2103_0000				
0x2200_0000	32M	BIT BAND REGION RAMD	*	Функционально выделенная область памяти на уровне процессорного ядра, позволяющая упростить битовые манипуляции в RAMD
0x4000_0000	1M	PERIPHERAL	S, M	Область отображения регистров периферии шины APB. Подробнее см. таблицу 15
0x4200_0000	32M	BIT BAND REGION PERIPHERAL	*	Функционально выделенная область памяти на уровне процессорного ядра, позволяющая упростить битовые манипуляции в PERIPHERAL
0x5000_0000	256M	EXT BUS	S, M	Область памяти для отображения внешней системной шины
0x6000_0000	1G	EXT BUS	S, M	Область памяти для отображения внешней системной шины
0xA000_0000	1G	EXTBUS	S, M	Область памяти для отображения внешней системной шины
0xE000_0000	512M	SYSTEM	-	Отображение внутренних регистров процессорного ядра. Подробнее см. таблицу 16

6.5.2 Распределение областей памяти для регистров периферии

Таблица 15 – Распределение адресного пространства периферии

	Базовый адрес	Обозначение	Доступ по шинам	Описание
0	0x4000_0000	MDR_CLK	S, M	Контроллер тактовых частот
1	0x4000_1000	MDR_BKP		Контроллер батарейного домена и часов реального времени
2	0x4000_2000	MDR_PWR		Контроллер блока управления питанием
3	0x4000_3000	MDR_FTMODE		Контроллер блока диагностики ошибок
4	0x4000_4000	MDR_WDT		Контроллер сторожевого таймера
5	0x4000_5000	MDR_EBC		Контроллер внешней шины
6	0x4000_6000	MDR_OTP		Контроллер OTP памяти
7	0x4000_7000	-		Зарезервировано
8	0x4000_8000	MDR_RAMD		Контроллер ОЗУ в области DATA
9	0x4000_9000	MDR_ROM		Контроллер ПЗУ
10	0x4000_A000	-		Зарезервировано
11	0x4000_B000	MDR_SCACHE		Контроллер кэш памяти RAMD
12	0x4000_C000	-		Зарезервировано
13	0x4008_0000	MDR_PORTA		Контроллер порта А
14	0x4008_1000	MDR_PORTB		Контроллер порта В
15	0x4008_2000	MDR_PORTC		Контроллер порта С
16	0x4008_3000	MDR_PORTD		Контроллер порта D
17	0x4008_4000	MDR_PORTE		Контроллер порта E
18	0x4008_5000	-		Зарезервировано
19	0x4008_6000	-		Зарезервировано
20	0x4008_7000	-		Зарезервировано
21	0x4008_8000	MDR_SPW0		Контроллер SpaceWire0
22	0x4008_9000	-		Зарезервировано
23	0x4008_A000	MDR_TMR0		Контроллер Таймера 0
24	0x4008_B000	MDR_TMR1		Контроллер Таймера 1
25	0x4008_C000	MDR_TMR2		Контроллер Таймера 2
26	0x4008_D000	MDR_TMR3		Контроллер Таймера 3
27	0x4008_E000	-		Зарезервировано
28	0x4008_F000	-		Зарезервировано
29	0x4009_0000	MDR_CAN0		Контроллер CAN0
30	0x4009_1000	-		Зарезервировано
31	0x4009_2000	-		Зарезервировано
32	0x4009_3000	-		Зарезервировано
33	0x4009_4000	-		Зарезервировано
34	0x4009_5000	MDR_SSP0		Контроллер SSP0
35	0x4009_6000	-		Зарезервировано
36	0x4009_7000	-		Зарезервировано
37	0x4009_8000	-		Зарезервировано
38	0x4009_9000	MDR_UART0		Контроллер UART0
39	0x4009_A000	MDR_UART1		Контроллер UART1
40	0x4009_B000	-		Зарезервировано
41	0x4009_C000	-		Зарезервировано
42	0x4009_D000	-	Зарезервировано	

	Базовый адрес	Обозначение	Доступ по шинам	Описание
43	0x4009_E000	MDR_ARCRX0	S, M	Контроллер ARINC0 RX
44	0x4009_F000	MDR_ARCRX0		DATA
45	0x400A_0000	MDR_ARCTX0		Контроллер ARINC0 TX
46	0x400A_1000	MDR_ARCTX0		DATA
47	0x400A_2000	-		Зарезервировано
48	0x400A_3000	-		Зарезервировано
49	0x400A_4000	-		Зарезервировано
50	0x400A_5000	-		Зарезервировано
51	0x400A_6000	MDR_MILDAT0		Контроллер MIL0 данные
52	0x400A_7000	MDR_MILCNR0		Контроллер MIL0 управление
53	0x400A_8000	MDR_MILDAT1		Контроллер MIL1 данные
54	0x400A_9000	MDR_MILCNR1		Контроллер MIL1 управление
55	0x400A_A000	MDR_ADC0		Контроллер ADC0
56	0x400A_B000	MDR_ADC1		Контроллер ADC1
57	0x400A_C000	MDR_DAC0		Контроллер DAC0
58	0x400A_D000	MDR_DAC1		Контроллер DAC1
59	0x400A_E000	-		Зарезервировано
60	0x400A_F000	-		Зарезервировано
61	0x400B_0000	-		Зарезервировано
62	0x400B_1000	-		Зарезервировано
63	0x400B_2000	-		Зарезервировано

6.5.3 Распределение областей памяти для регистров периферии ядра

Таблица 16 – Распределение внутреннего адресного пространства ядра

Базовый адрес	Размер, байт	Обозначение	Описание
0xE000_E008	-	System Control Block	Системные регистры
0xE000_E010		System Timer	Регистры системного таймера
0xE000_E100		NVIC	Регистры контроллера прерываний
0xE000_ED00		System Control Block	Системные регистры
0xE000_ED90		MPU	Регистры блока защиты памяти
0xE000_EF00		NVIC	Регистры контроллера прерываний
0xE000_EF30		FPU	Регистры блока вычислений с плавающей запятой
0xE004_0000		TPIU	Блок управления портом трассировки TPIU
0xE004_1000		ETM	Блок формирования трассировки ядра ETM
0xE004_2000		MDR_DMA	Блок прямого доступа в память
0xE004_3000		-	Зарезервировано
0xE004_4000		MDR_ICACHE	Блок управления I-кэш памятью
0xE004_5000		MDR_DCACHE	Блок управления D-кэш памятью
0xE00F_F000		ROM TABLE	Таблица описания периферии ядра

Описание блоков, не представленных в данном документе, можно найти в Cortex-M4 Technical Reference Manual.

6.6 Помехозащищенное кодирование

В микроконтроллере применяется помехозащищенное кодирование SEC-DED (исправление одиночных и обнаружение двойных ошибок):

- кодом Хемминга (72,64) – для внутренних памятей и памяти на внешней системной шине;
- кодом Хемминга (7,4) – для задания режима работы.

Код Хемминга (7,4) применяется для задания режима работы микроконтроллера, подробнее см. подраздел 6.4 «Режимы работы микроконтроллера».

Кодом Хемминга (72,64) всегда защищаются внутренние памяти, включая кэш-память. ECC кодирование может использоваться и для защиты внешней памяти. С помощью ECC кодируются как непосредственно сами данные, так и адрес расположения данных. В результате автоматически исправляются одиночные ошибки в шине данных, обнаруживаются двойные ошибки в поле данных, а также одиночные или двойные ошибки в поле адреса. Одиночные ошибки в поле адреса не исправляются и считаются неустранимыми ошибками. Кодирование производится при операциях записи в блоках ECC_T. На основании записываемых данных и адреса вычисляется поле HWECC[7:0] и передается совместно с полем данных (далее по тексту генерация ECC). При считывании на основании данных, адреса и поля HRECC[7:0] в блоке ECC_R проверяется полученная информация, и при необходимости исправляется одиночная ошибка в поле данных, либо обнаруживаются одиночные и двойные ошибки в поле адреса и двойные в поле данных (далее по тексту проверка ECC).

6.6.1 ECC кодирование для ПЗУ

При обращении в область загрузочного ПЗУ памяти проверка и, при необходимости, исправление по ECC происходит на границе контроллера памяти ПЗУ и непосредственно самой матрицей ПЗУ. После чего, корректные данные с контрольной суммой передаются процессорному ядру, где они еще раз проверяются по ECC, но уже без исправления, только с целью обеспечения достоверности.

ЕСС биты загрузочной программы вычислены на этапе создания программы и защиты в ПЗУ микроконтроллера на этапе разработки.

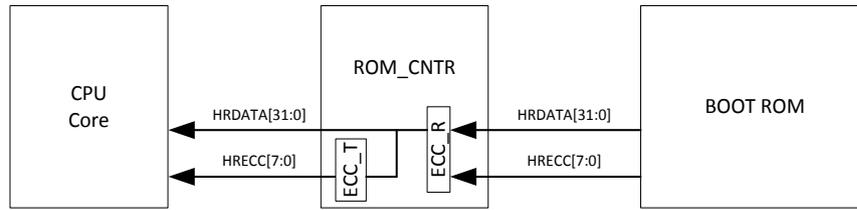


Рисунок 14 – Схема контроля ЕСС на пути от ПЗУ

6.6.2 ЕСС кодирование для OTP памяти программ (1986BE8)

При обращении в область однократно программируемой памяти OTP (1986BE8Т) и ОЗУ (1986BE81Т) памяти программ проверка ЕСС происходит на границе контроллера OTP (ОЗУ) и непосредственно самой матрицей памяти. При этом из матрицы извлекается строка с размером в общей сложности 160 бит (128 бит данных и 32 бита ЕСС). Для всех слов строки проводится проверка и при необходимости исправление данных по ЕСС. Затем данные передаются в кэш данные (DCACHE) или кэш инструкции (ICACHE) и сохраняются в них с ЕСС. Одновременно данные передаются в процессорное ядро. Возвращаемые процессорному ядру данные снабжены ЕСС суммой, которая используется для обеспечения достоверности передаваемых данных. Так как контроллер OTP(ОЗУ) имеет сложную структуру, работающую под управлением автомата состояний, а также для обеспечения диагностики сбоев в блоке контроллера OTP он реализован в LOCKSTEP режиме. Таким образом, сравнение поведений двух копий контроллера позволяет обнаружить сбои, возникшие в самом контроллере OTP(ОЗУ).

Генерация ЕСС бит при программировании OTP (ОЗУ) памяти осуществляется программными средствами разработки программ и вычисляется при программировании памяти.

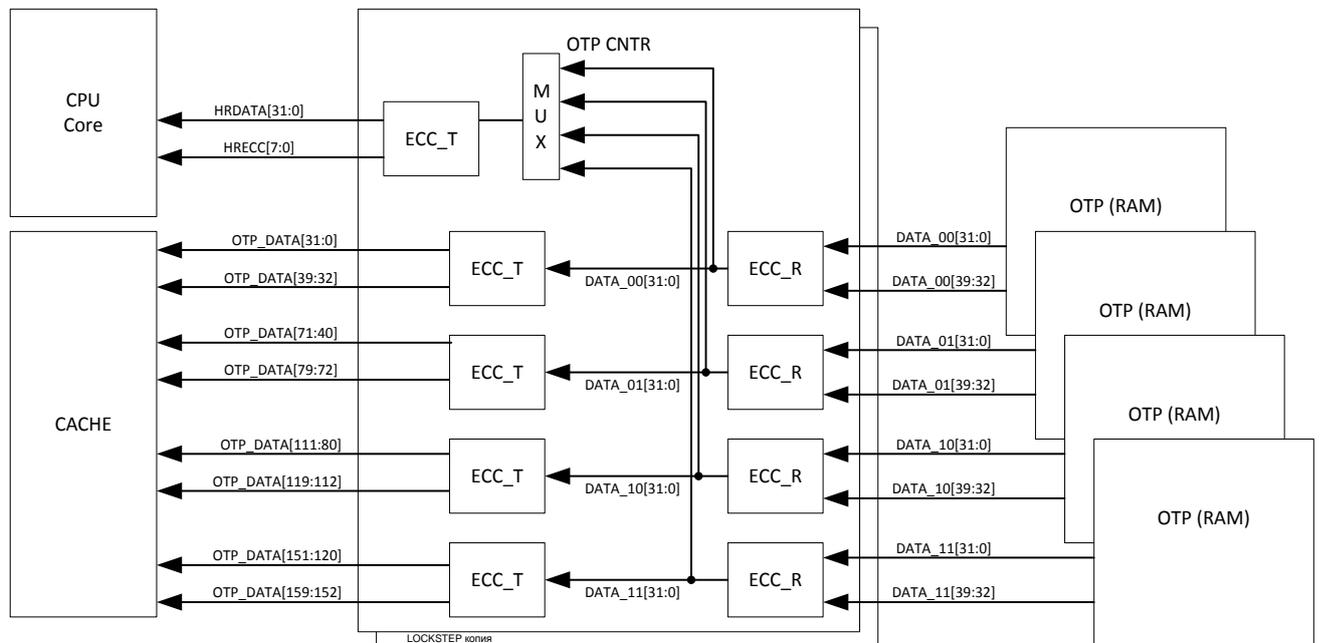


Рисунок 15 – Схема контроля ЕСС на пути от OTP (ОЗУ) памяти программ

6.6.1 ECC кодирование для ОЗУ памяти программ (1986BE81)

6.6.1.1 ECC кодирование для ОЗУ данных RAMD

При обращении в область ОЗУ данных генерация и проверка ECC происходит на границе контроллера памяти и непосредственно самой матрицы.

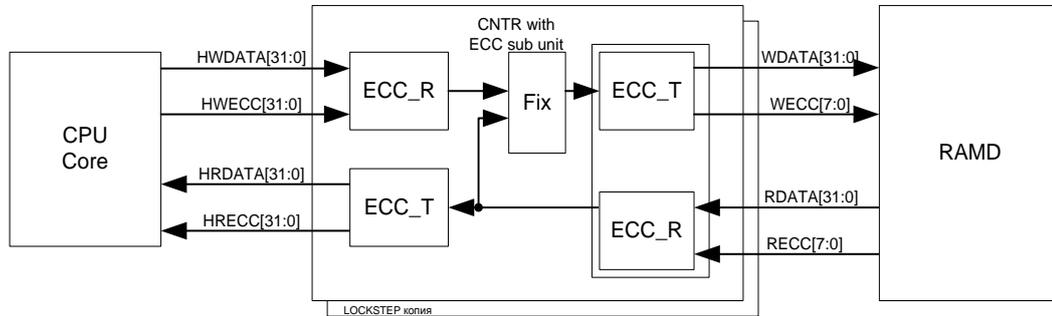


Рисунок 16 – Схема контроля ECC для RAMD

В ряде случаев при записи в память сначала выполняется чтение, это связано с необходимостью обработки байтов и полуслов при работе с памятью, имеющей 32-х битную организацию с защитой ECC. Так, например, при записи байта в ОЗУ необходимо считать из памяти ранее записанное слово с ECC, проверить его ECC, модифицировать в проверенном слове записываемый байт, сгенерировать новое ECC и записать его обратно в память. Также необходимо иметь в виду, что после включения питания микроконтроллера память содержит произвольные значения и требует проведение первоначальной процедуры инициализации.

6.6.1.2 ECC кодирование для внешней шины

При обращении в область EXT_BUS существует два варианта организации ECC:

- параллельный (аналогичный организации ECC для RAM);
- последовательный (когда ECC биты расположены в отдельной области памяти).

При работе с внешней памятью, в зависимости от режима ECC, биты считываются одновременно с данными (параллельная организация), либо в два этапа (последовательная организация: сначала считываются данные, затем ECC), и затем производится проверка ECC (подробнее см. 7.7.1 «Организация доступа к внешней шине EXTBUS»).

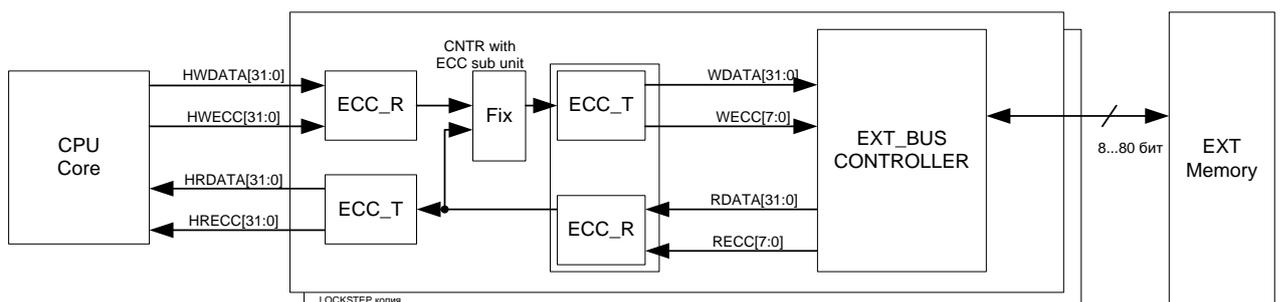


Рисунок 17 – Схема контроля ECC для внешней шины

Данные, считанные по внешней шине через шину инструкций IBus или шину данных DBus, могут быть закэшированы в кэш. Для этого при возвращении данных процессору к ним снова генерируется ECC для записи в кэш-память. При работе кэш-памяти когерентность закэшированных данных не обеспечивается. Также необходимо иметь в виду, что после включения питания внешняя память может содержать произвольные значения и требует дополнительной процедуры инициализации.

6.6.2 ECC кодирование для периферии

Периферийные блоки, имеющие в своем составе большие массивы памяти, защищены ECC. Периферийные блоки также могут иметь собственные механизмы исправления и обнаружения ошибок.

6.6.3 Н-матрица ECC по коду Хемминга (72, 64)

Таблица 17 – Н-матрица ECC кода HSIA70a

Проверочные биты	Байты данных																																										
	0								1								2								3																		
Bits	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
S0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0					
S1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1			
S2	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0		
S3	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
S4	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	0	0	0	
S5	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	0		
S6	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
S7	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Проверочные биты	Байты данных																																																	
	4								5								6								7																									
Bits	0	1	2	3	4	5	6	7	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63										
S0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0					
S1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0				
S2	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1		
S3	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
S4	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	1	0	0	1	0	0	0	0		
S5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	
S6	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	
S7	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.6.4 Н-матрица ECC по коду Хемминга (8, 4)

Таблица 18 – Матрица для кода Хемминга (8, 4)

MODE[]	7	6	5	4	3	2	1	0
S1	0	0	1	0	1	0	1	1
S2	0	1	0	0	1	1	0	1
S3	1	0	0	0	1	1	1	0
S0	1	1	1	1	1	1	1	1

S0 – бит чётности. Соответствующий ему MODE[4] вырезается при сохранении в поле MODE[6:0] регистров ВКР REG_60_TMRx.

6.6.5 Функция вычисления ECC для записи в память

Вычисление 8-ми битной ECC суммы для 32-х битного слова в памяти осуществляется на основании самого слова (unsigned int DATA) и адреса (unsigned int ADR) расположения слова.

SECC = getecc (((unsigned long long)ADR)<<32) | DATA);

Тело функции getecc:

```
unsigned int getecc(unsigned long long inw)
{
    unsigned char ecc = 0;
    unsigned long long inputw;
```

```
unsigned long long H[8];
int i,j;
unsigned char res = 0;
```

```
H[0] = (unsigned long long) 0x0738C808099264FF;
H[1] = (unsigned long long) 0x38C808099264FF07;
H[2] = (unsigned long long) 0xC808099264FF0738;
H[3] = (unsigned long long) 0x08099264FF0738C8;
H[4] = (unsigned long long) 0x099264FF0738C808;
H[5] = (unsigned long long) 0x9264FF0738C80809;
H[6] = (unsigned long long) 0x64FF0738C8080992;
H[7] = (unsigned long long) 0xFF0738C808099264;
```

```
for (i=0; i < 8; i++)
{
    inputw = H[i] & inw;
    res = 0;
    for (j=0; j < 64; j++)
    {
        res = res ^ ((inputw >> j) & 0x0000000000000001);
    }
    ecc = ecc | res << i;
}
return ecc;
```

6.7 Система команд процессорного ядра

Инструкция	Выполняемое действие	Изменяемые флаги	Примечание
Вызов и возврат из функций			
BL label	PC ← адрес label; LR ← адрес возврата;	-	Используется для вызова функций
BX Rn	PC ← Rn;	-	Используется для перехода по адресу из регистра Rn. Может использоваться для возврата из функции если Rn = LR
Загрузка констант			
ADR Rd, label	Rd ← адрес label;	-	
LDR Rd, =constant	Rd ← constant;	-	Псевдоинструкция, аналогична MOV, MVN
MOV Rd, constant	Rd ← constant;	NZC	$0 \leq \text{constat} \leq 255$
MVN Rd, constant	Rd ← ~ constant;	NZC	$0 \leq \text{constat} \leq 255$
LDRB Rd, [address mode]	Rd ← ;		

6.8 Интерфейс отладки

Для отладки приложений используется интерфейс JTAG/SWD.

При выборе отладочного интерфейса JTAG_A или JTAG_B выводы, мультиплексированные с выводами отладочного интерфейса, переопределяются в выводы JTAG независимо от выполняемых ими функций.

Таблица 19 – Сигналы интерфейсов отладки

Сигнал JTAG	Сигналы SWD	JTAG_A=1	JTAG_B=1
nTRST	-	PA[0]	PB[4]
TCK	SWCLK	PA[1]	PB[5]
TMS	SWDIO	PA[2]	PB[6]
TDI	-	PA[3]	PB[7]
TDO	SWV	PA[4]	PB[8]

6.9 Тестовые режимы работы

Микросхема переключается в тестовый режим при высоком уровне на выводе JTAG_EN или MODE = 1111.

В тестовом режиме к выводам JTAG_A подключается тестовый TAP-контроллер. Загрузкой инструкций IR в TAP-контроллер микросхема переводится в тестовые режимы, указанные в таблице. При переходе в тестовый режим работа микроконтроллера останавливается.

Таблица 20 – Тестовые режимы работы

IR[7:0]	TYPE	TTMODE	Название тестового режима	Описание
0x00			Boundary Scan	
0x14			IDCODE	
0x13	SCAN	0	MAIN_ATPG	Режим тестирования основной логики МК
0x02		1	ADC0_OPEN	Режим раскрытия ADC0
0x03		2	BKP_OPEN	Режим раскрытия BKP домена и тестирования LSE, LSI, WDG, BKP
0x04		3	HSE_OPEN	Режим раскрытия HSE, PLL
0x05		4	ADC1_OPEN	Режим раскрытия ADC1
0x06		5	DAC_OPEN	Режим раскрытия DAC
0x07	MEM	6	OTP_OPEN	Режим раскрытия OTP (1986VE8Т)
0x07	MEM	6	RAM_OPEN	Режим раскрытия RAM(1986VE81Т)
0x09	MEM	8	RAMD_OPEN	Режим раскрытия RAMD
0x0A	MEM	9	ROM_OPEN	Режим раскрытия ROM
0x0B		10	ETH_OPEN	Режим раскрытия ETH PHY
0x0C		11	SPW_OPEN	Режим раскрытия SPW
0x0D	MEM	12	ETH_MEM_OPEN	Режим раскрытия ETH памяти
0x0E	MEM	13	ARC_MEM_OPEN	Режим раскрытия ARC памяти
0x0F	MEM	14	MIL_MEM_OPEN	Режим раскрытия MIL памяти
0x01		16	LDO_OPEN	Режим раскрытия LDO
0x17	MEM	17	CAN_MEM_OPEN	Режим раскрытия CAN памяти
			Резерв	

6.9.1 Режим тестирования ОЗУ

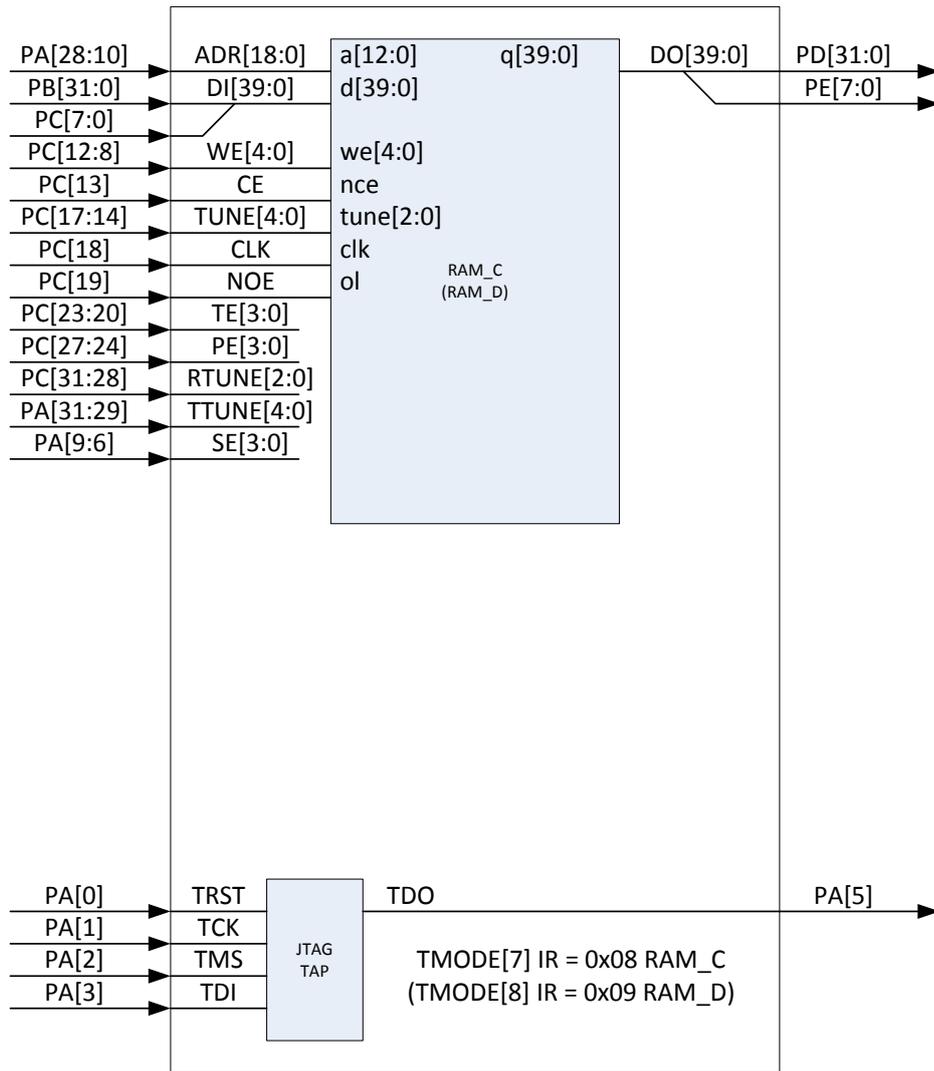


Рисунок 18 – Режим тестового раскрытия блоков основной памяти RAM

6.9.2 Режим тестирования Масочного ПЗУ (BOOTROM)

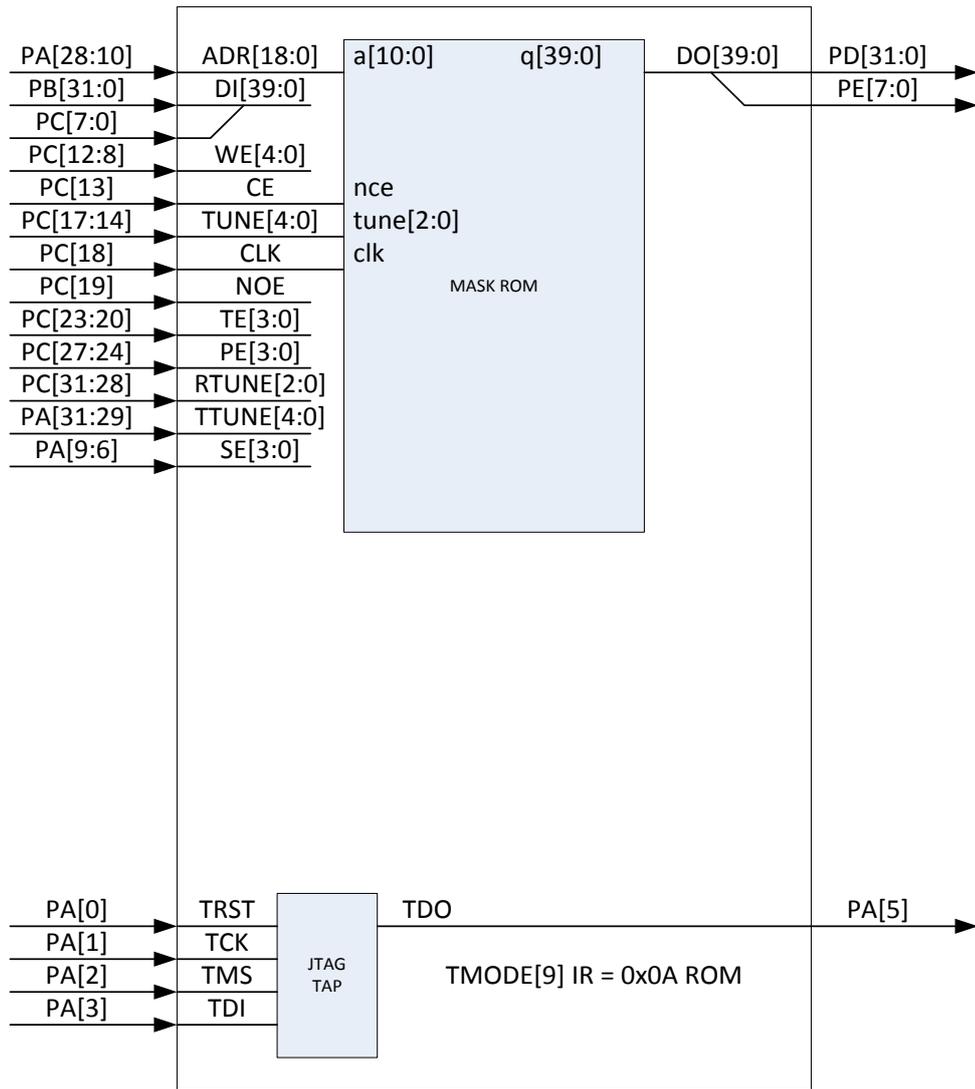


Рисунок 19 – Режим тестового раскрытия блоков основной памяти ROM

6.9.3 Режим тестирования OTP (для микросхем 1986BE8Т)

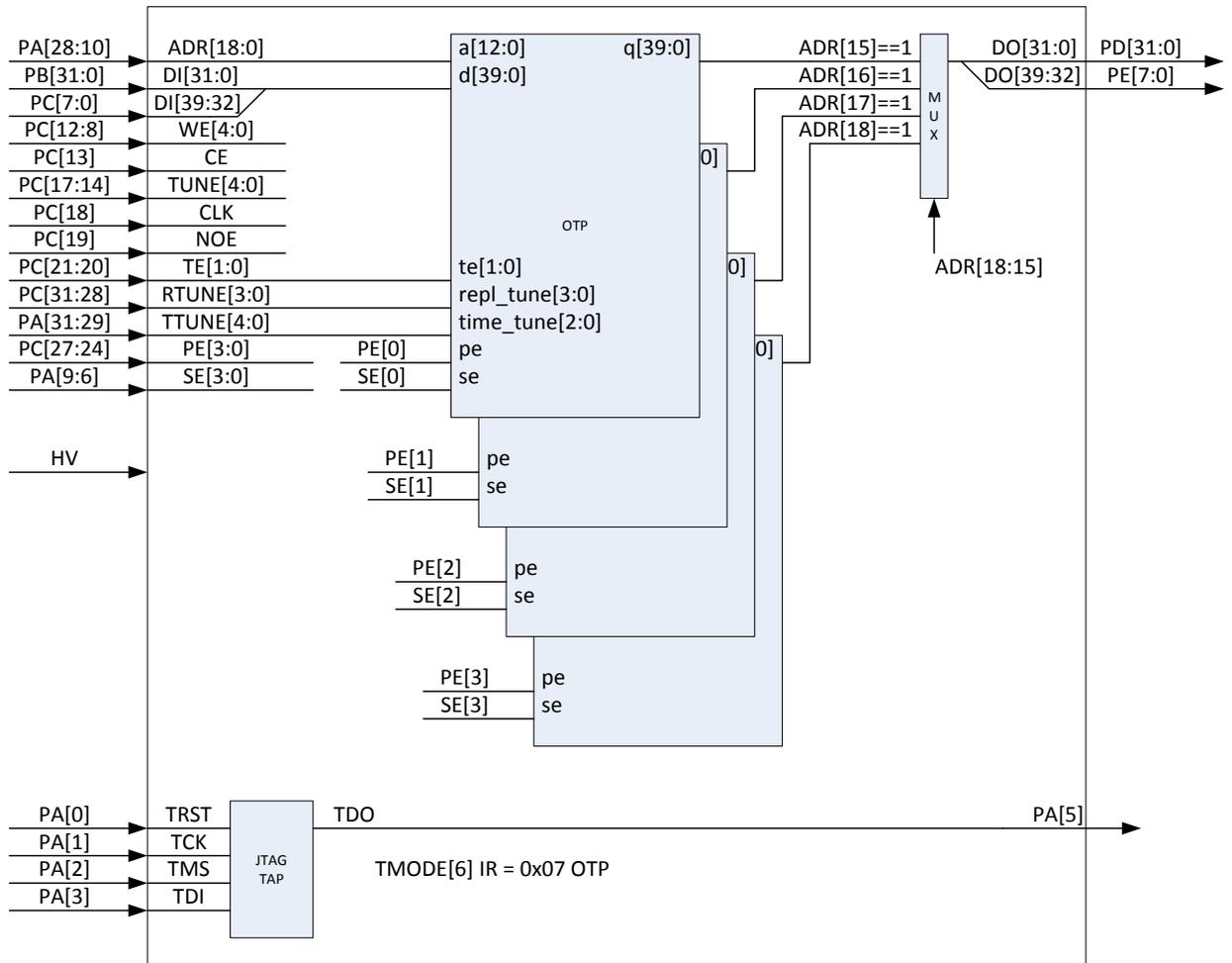


Рисунок 20 – Режим тестового раскрытия блоков основной памяти OTP

6.9.4 Режим тестирования RAM (для микросхем 1986BE81Т)

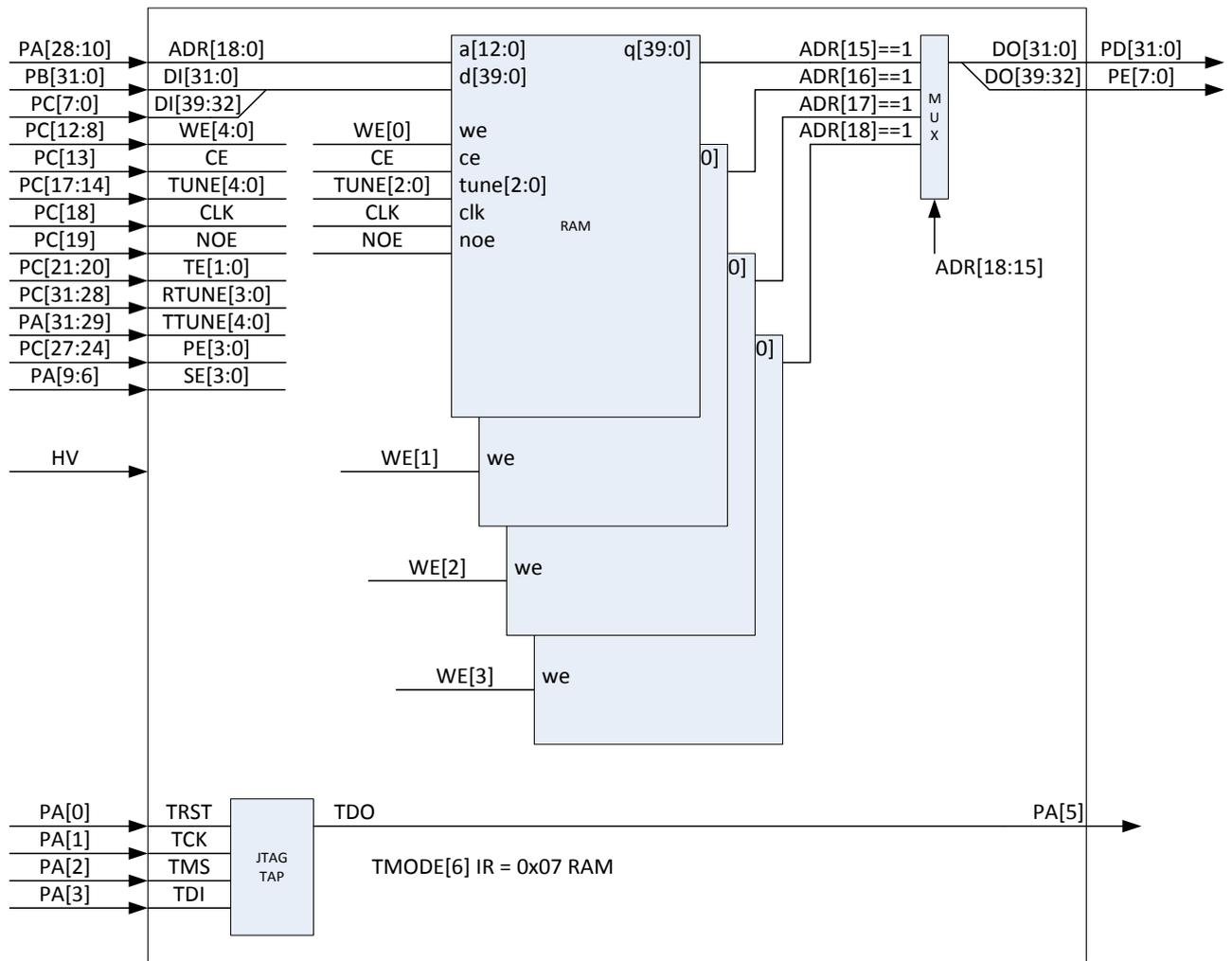


Рисунок 21 – Режим тестового раскрытия блоков основной памяти RAM

6.9.5 Режим тестирования RAM_CAN

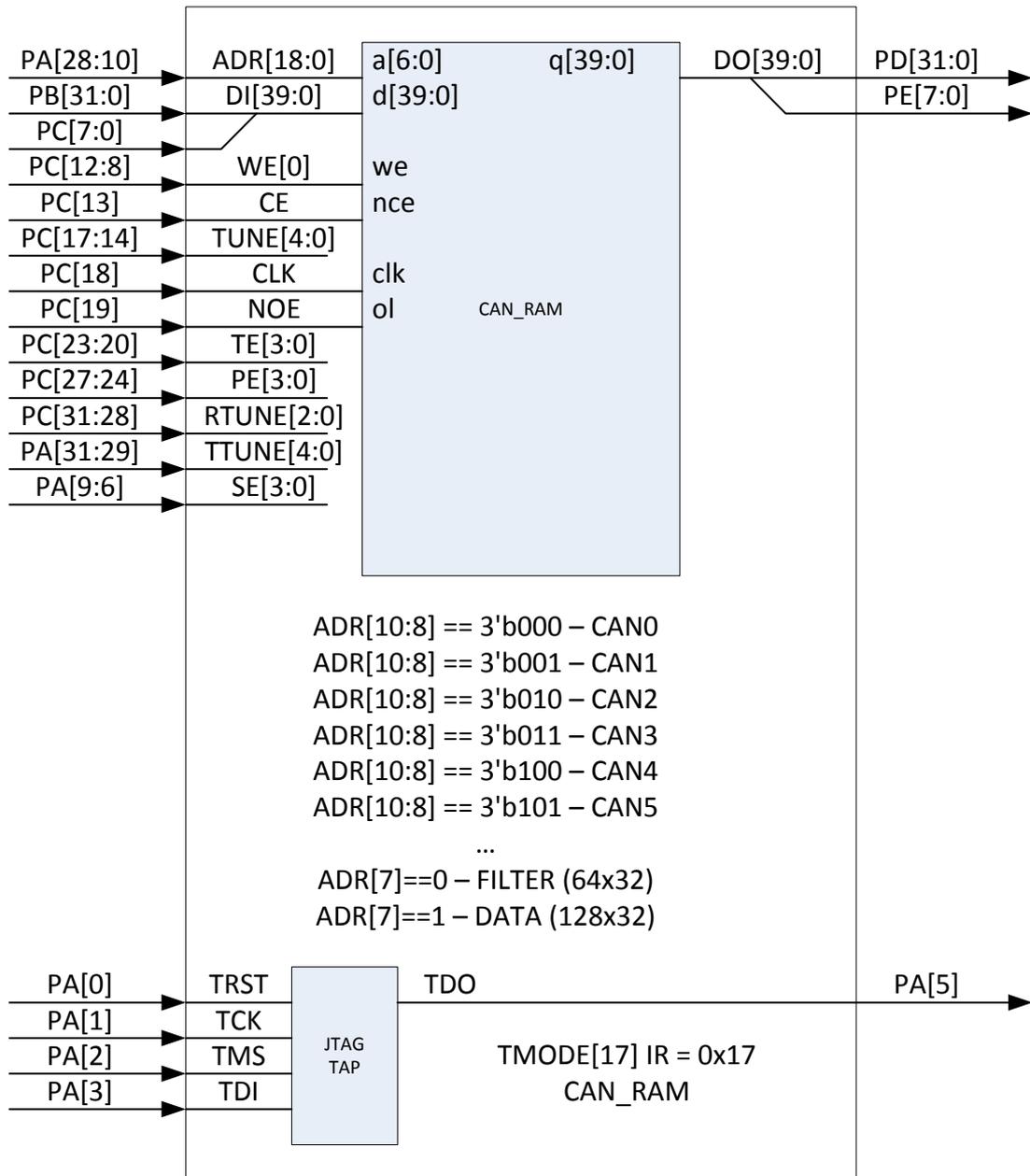


Рисунок 22 – Режим тестового раскрытия блоков памяти CAN

6.9.6 Режим тестирования ETH_MEM

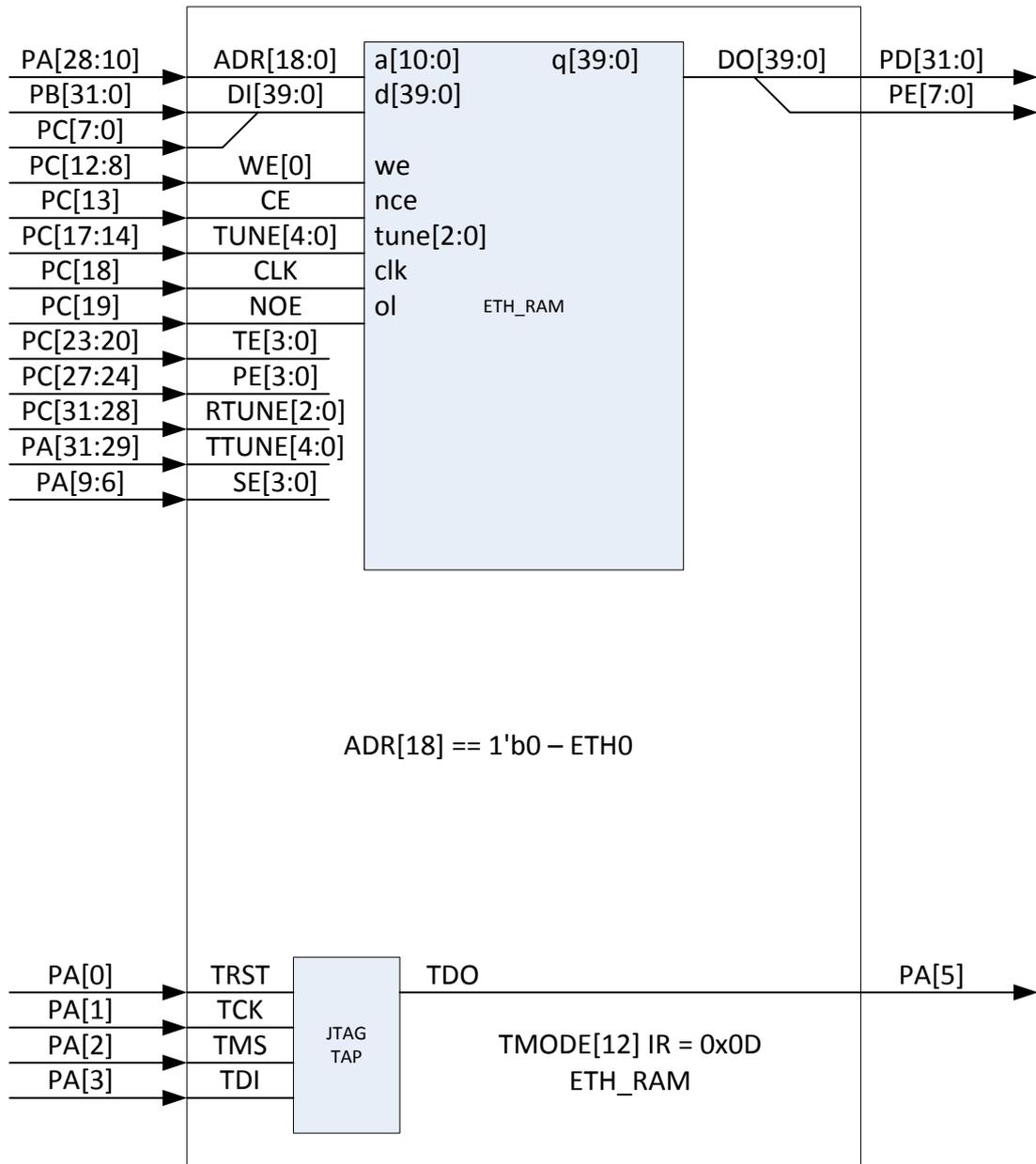


Рисунок 23 – Режим тестового раскрытия блоков памяти Ethernet

6.9.7 Режим тестирования RAM_MIL

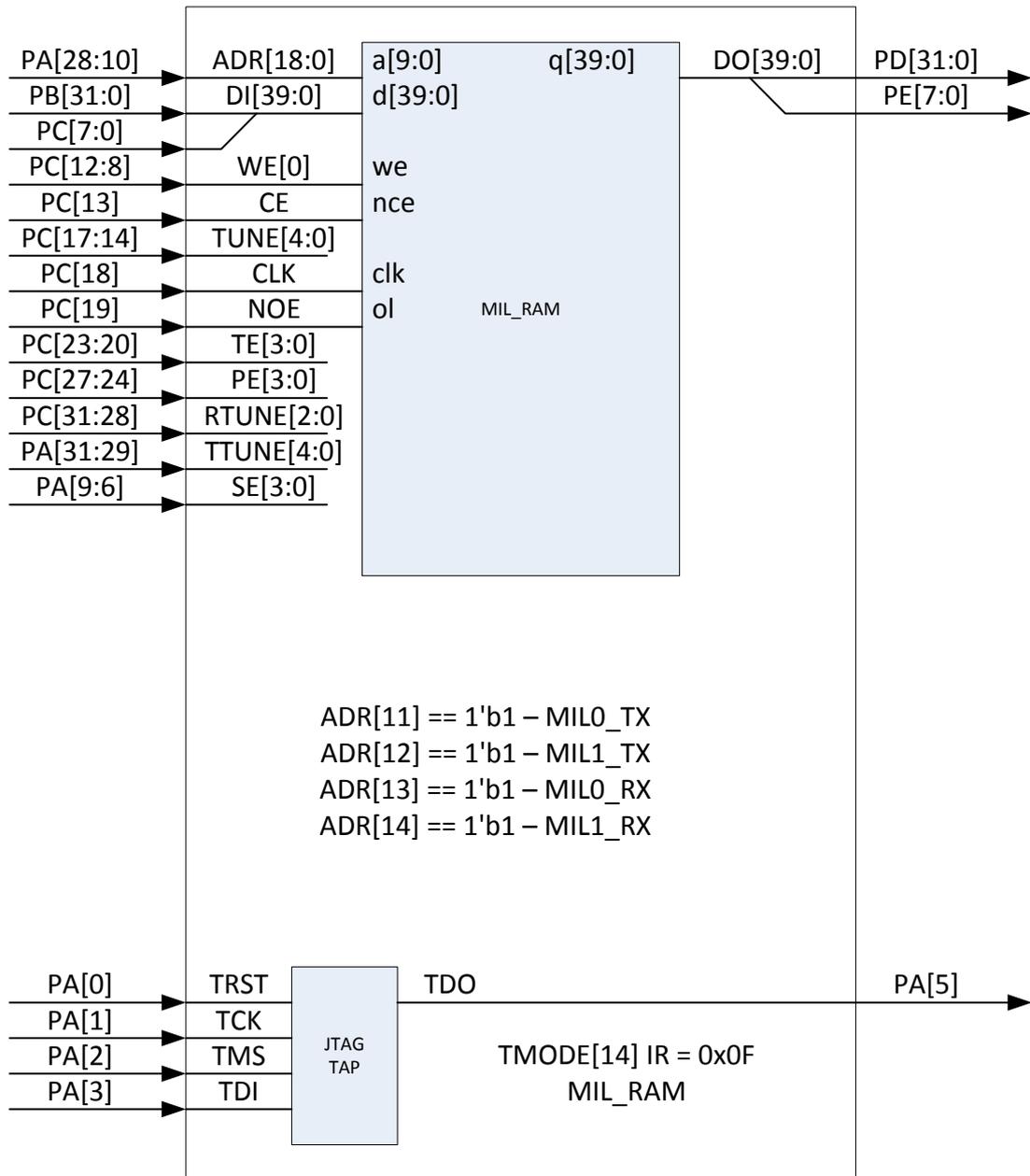


Рисунок 24 – Режим тестового раскрытия блоков памяти MIL

6.9.8 Режим тестирования RAM_ARC

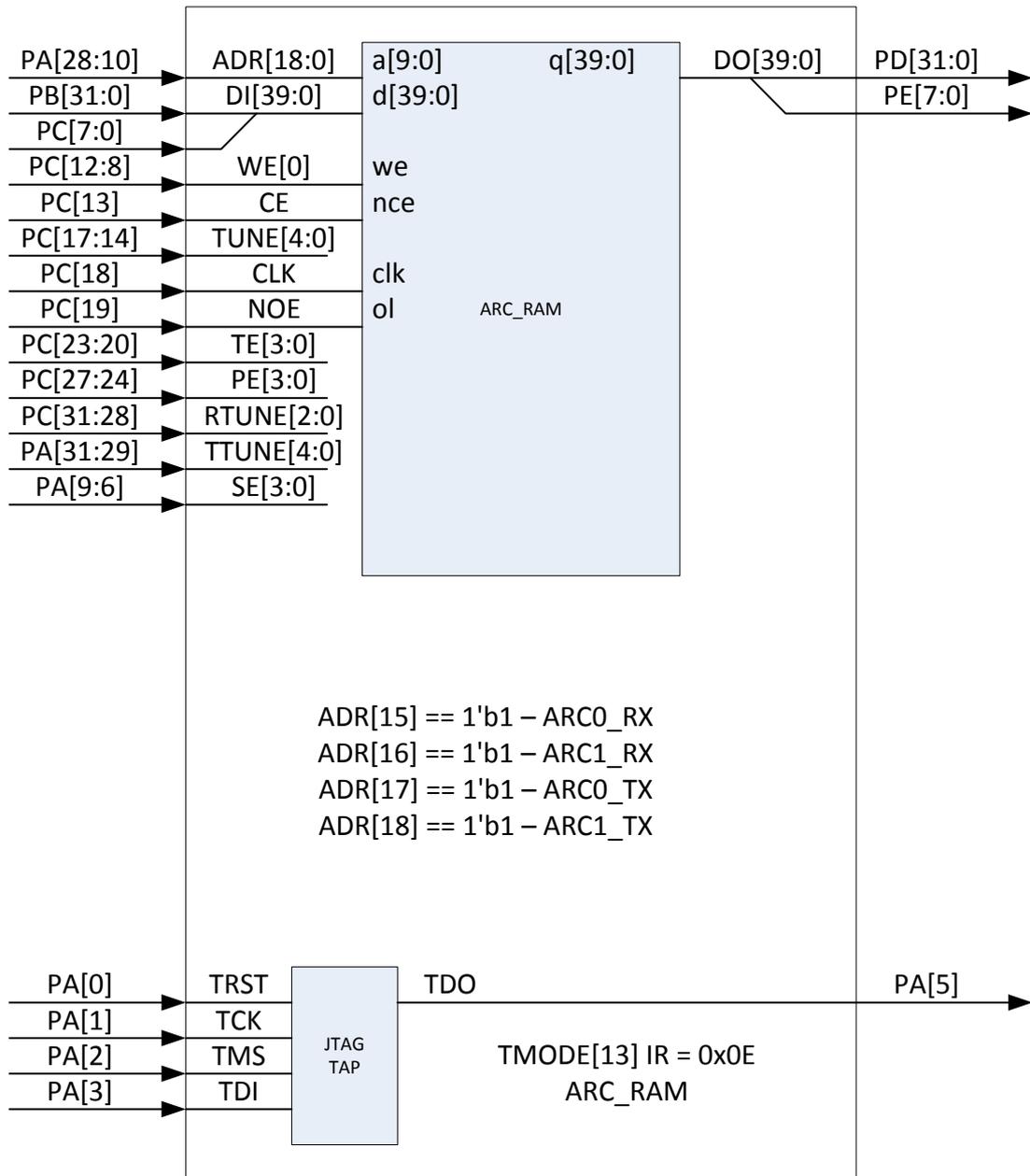


Рисунок 25 – Режим тестового раскрытия блоков памяти ARINC

6.9.9 Режим тестирования генераторов и PLL

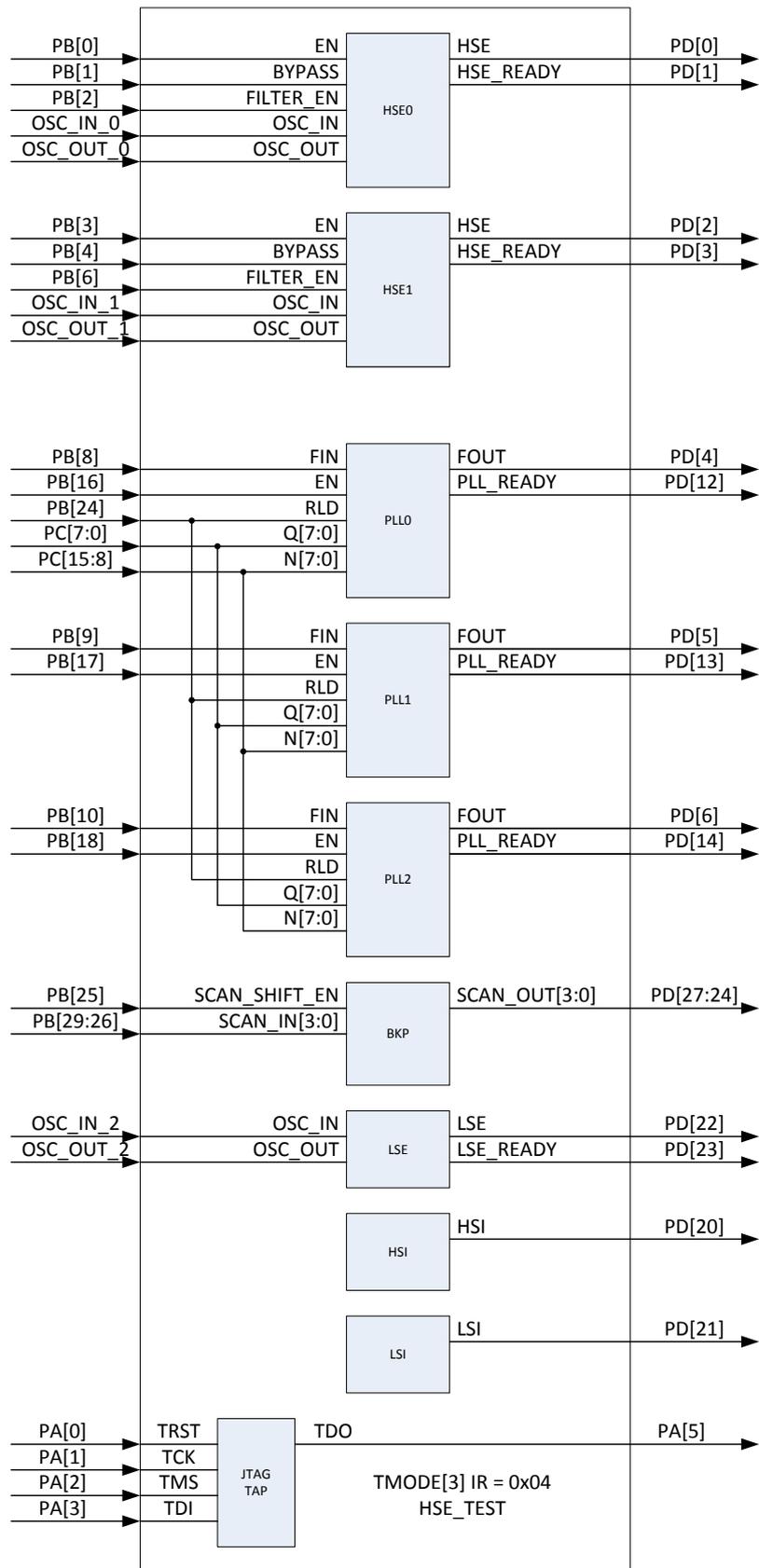


Рисунок 26 – Режим тестового раскрытия блоков PLL и генераторов

6.9.10 Режим тестирования АЦП

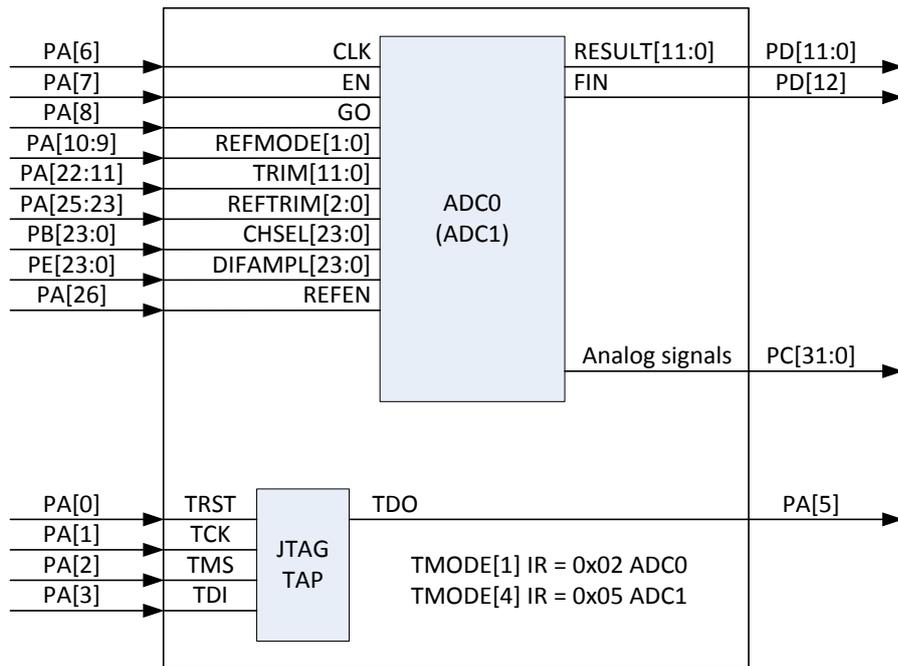


Рисунок 27 – Режим тестового раскрытия блоков АЦП

6.9.11 Режим тестирования DAC

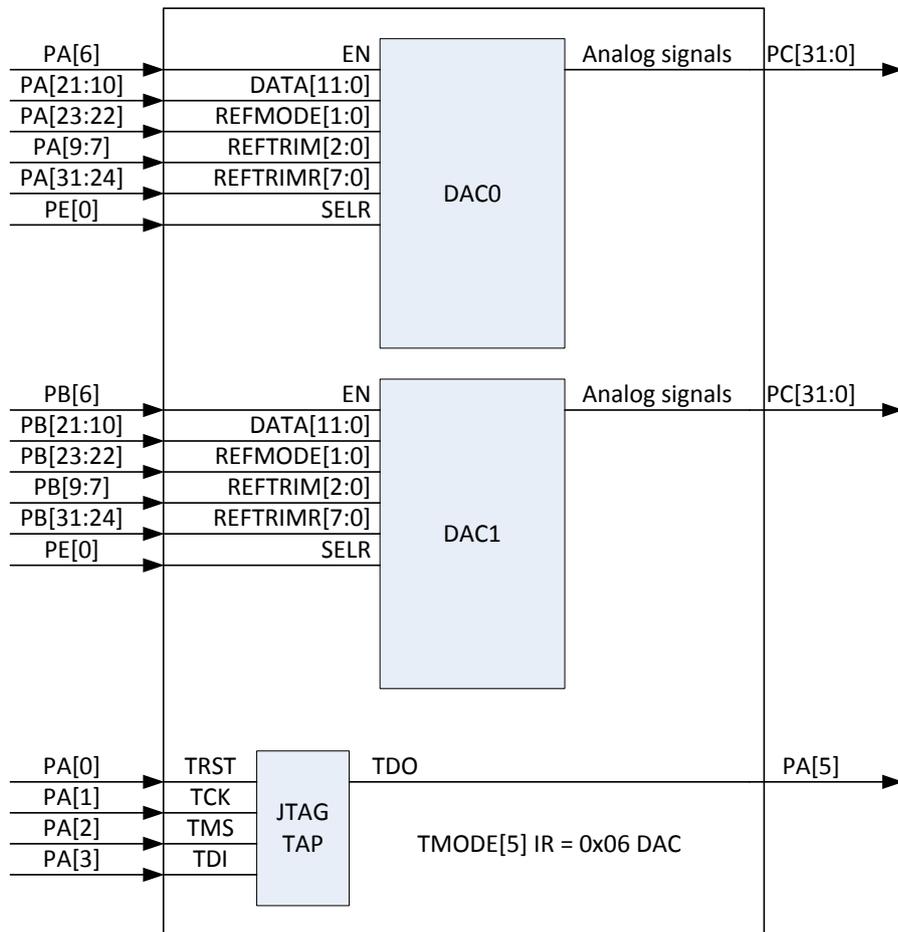


Рисунок 28 – Режим тестового раскрытия блоков памяти ЦАП

6.9.12 Режим тестирования приемопередатчиков SPW

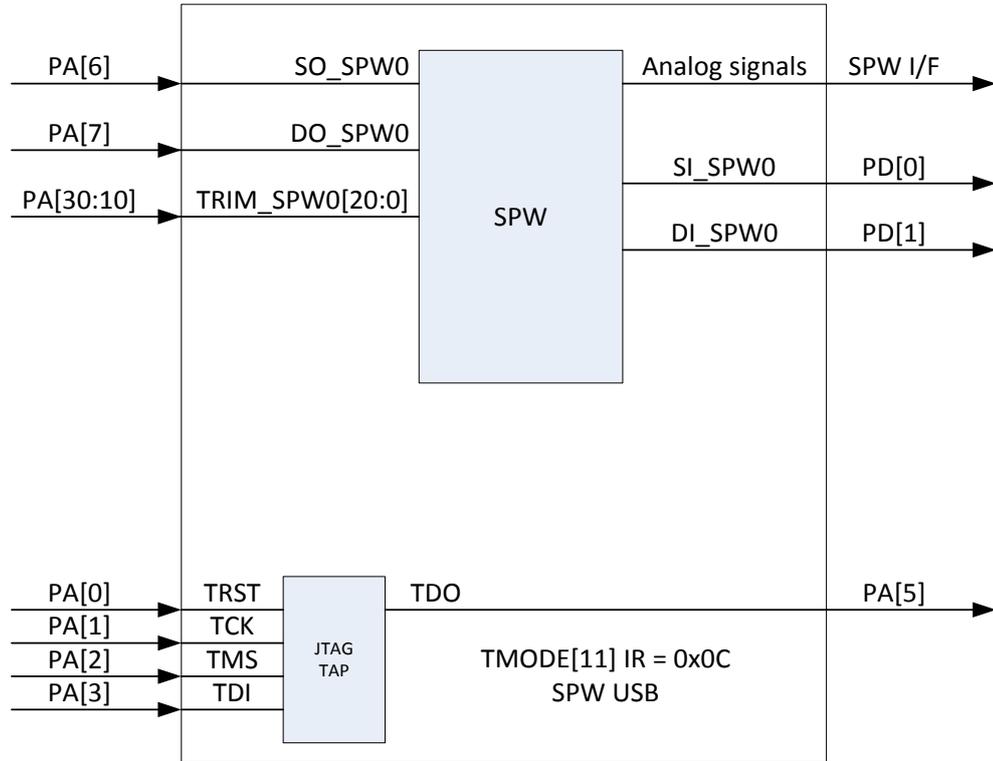


Рисунок 29 – Режим тестового раскрытия блоков передатчиков SpaceWire

6.9.13 Режим тестирования ETНPНУ

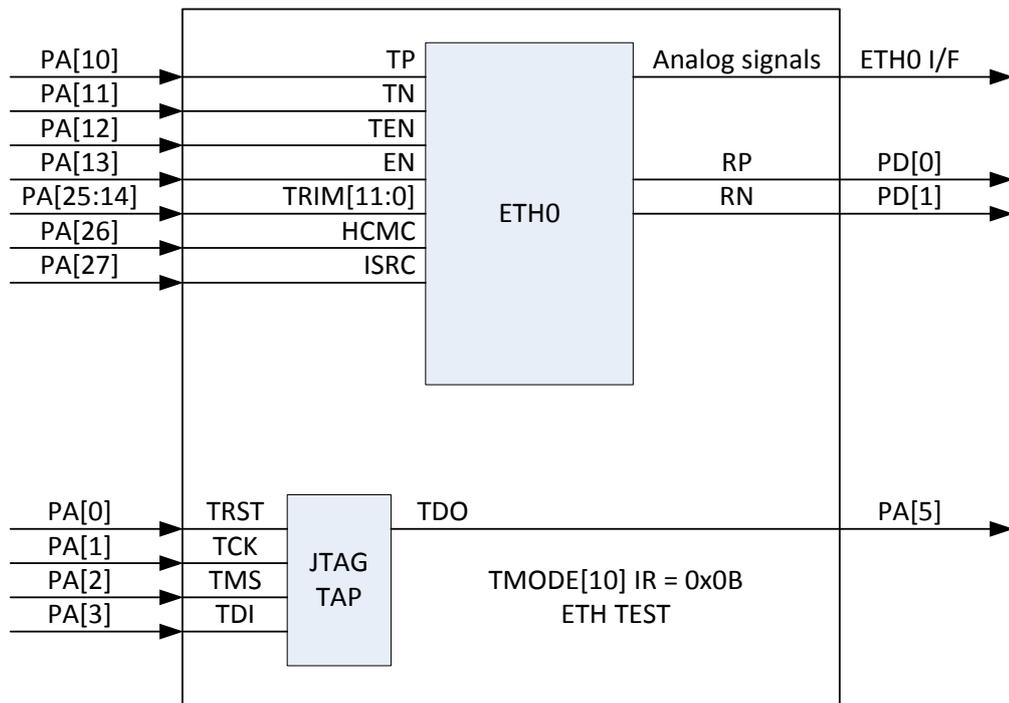


Рисунок 30 – Режим тестового раскрытия блоков передатчиков Ethernet

6.9.14 Режим тестирования PMU (VoltageDetect)

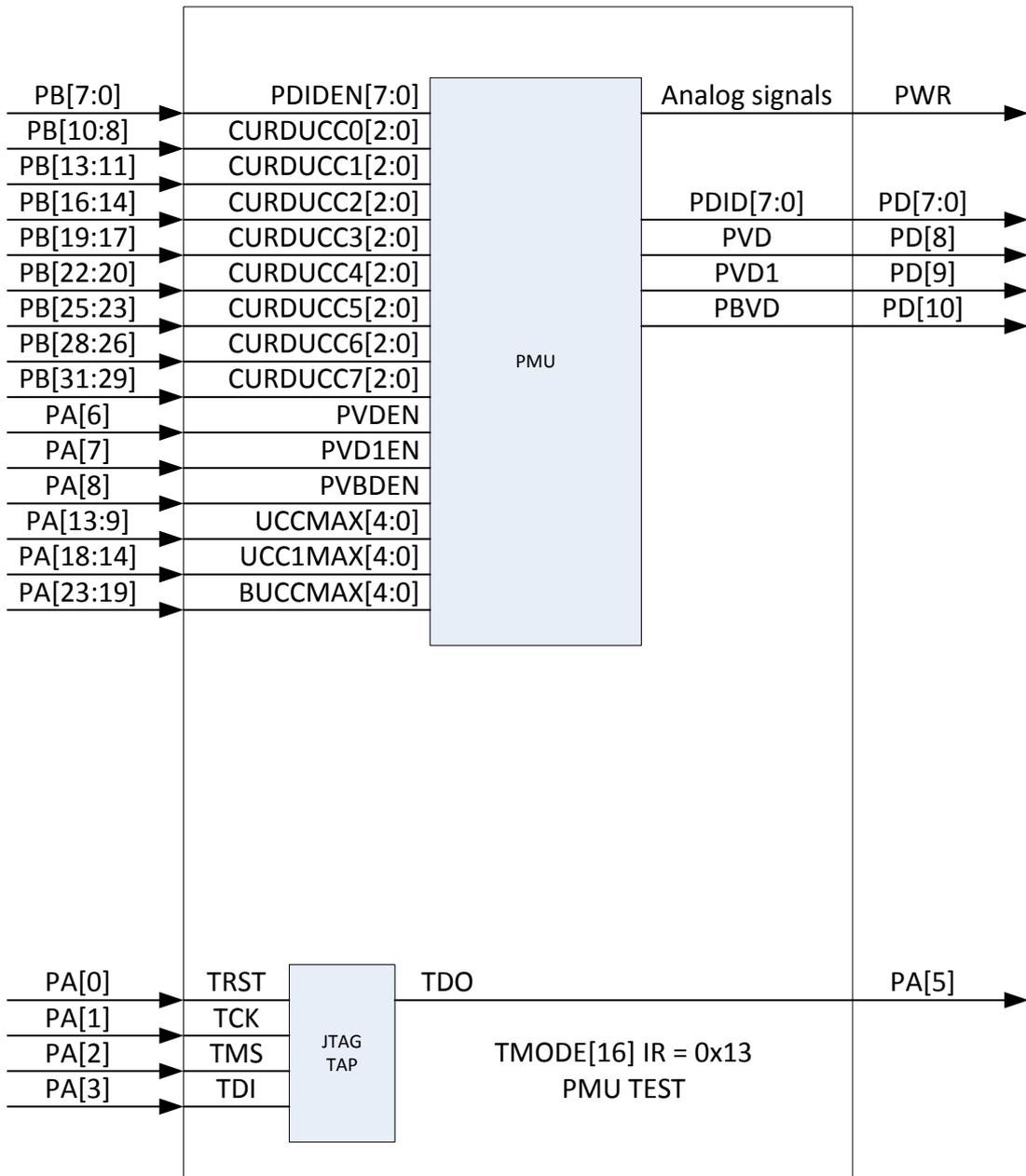


Рисунок 31 – Режим тестового раскрытия блок управления питанием

6.9.15 Режим тестирования ВКР

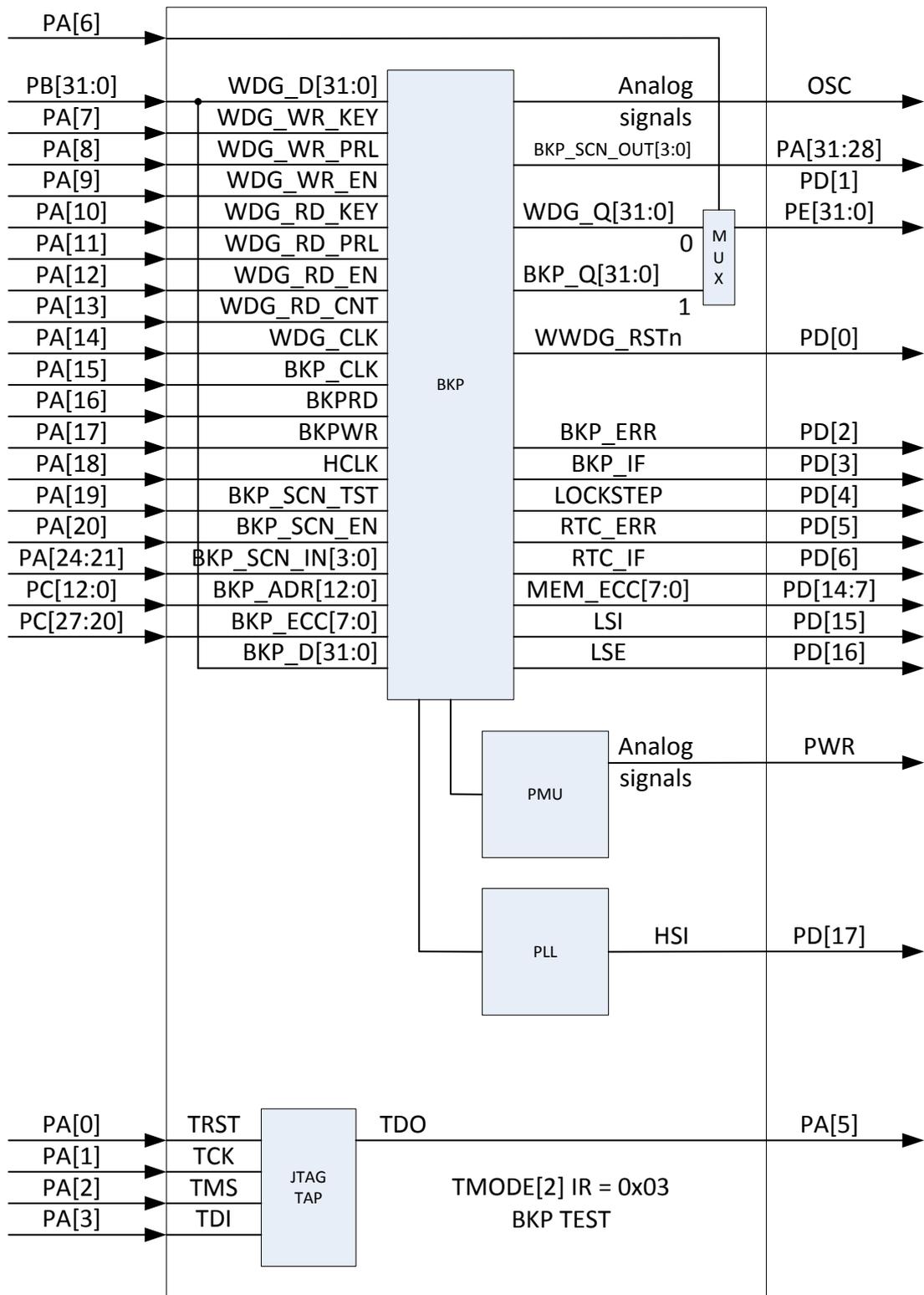


Рисунок 32 – Режим тестового раскрытия блока батарейного домена

7 Программная модель микроконтроллера

В данном разделе приводятся принципы программирования и описание регистров микроконтроллера.

7.1 Идентификация микросхемы

Идентификация возможна только для микросхем 1986BE8 с OTP-памятью. Для идентификации микросхем в последние 16 слов OTP-памяти записана идентификационная информация.

Ячейки памяти с адресом 0x0100_0000 и смещениями с 0x0001_FFC0 по 0x0001_FFFF зарезервированы и не могут использоваться для хранения пользовательской программы. Эти регистры содержат идентификационную информацию.

7.1.1 Описание полей идентификации

Таблица 21 – Описание регистров

Базовый адрес	Смещение	Название	Описание
0x0100_0000	0x0001_FFC0 0x0001_FFC4 0x0001_FFC8 0x0001_FFCC	WAFER_INFO	Номер партии пластин, номер пластины в партии
	0x0001_FFD0	PACKAGE_INFO	Технологический номер партии микросхем, номер микросхемы
	0x0001_FFD4 0x0001_FFD8 0x0001_FFDC 0x0001_FFE0 0x0001_FFE4 0x0001_FFE8 0x0001_FFEC 0x0001_FFF0 0x0001_FFF4 0x0001_FFF8 0x0001_FFFC	–	Резерв, может содержать произвольную информацию, но корректную по ECC

WAFER_INFO (4 слова = 16 байт) состоит из WAFER_LOT_ID (15 байт) и WAFER_ID (1 байт). Запись в эти поля производится на этапе измерения пластины.

- WAFER_LOT_ID – номер партии пластин, 15 символов в ASCII кодировке.
- WAFER_ID – номер пластины в партии пластин, натуральное число от 1 до 25.

PACKAGE_INFO (1 слово = 4 байта) состоит из PACKAGE_LOT_ID (2 байта) и PACKAGE_ID (2 байта). Запись в эти поля производится на этапе измерения микросхемы в корпусе.

- PACKAGE_LOT_ID – технологический номер партии микросхем, натуральное число от 0 до 99.
- PACKAGE_ID – номер микросхемы в партии микросхем, натуральное число от 0 до 4000.

Таблица 22 – Описание полей идентификации

Обозначение поля	WAFER_INFO														PACKAGE_INFO					
	WAFER_LOT_ID														WAFER_ID		PACKAGE_LOT_ID		PACKAGE_ID	
Адрес	0x0101_FFC0	0x0101_FFC1	0x0101_FFC2	0x0101_FFC3	0x0101_FFC4	0x0101_FFC5	0x0101_FFC6	0x0101_FFC7	0x0101_FFC8	0x0101_FFC9	0x0101_FFCA	0x0101_FFCB	0x0101_FFCC	0x0101_FFCD	0x0101_FFCE	0x0101_FFCF	0x0101_FFD0	0x0101_FFD1	0x0101_FFD2	0x0101_FFD3
Кодировка	ASCII String[15]														unsigned char	unsigned short		unsigned short		
Пример	"M12345"														0x13	0x0059		0x1234		

Расшифровка данных примера: микросхема из партии пластин «M12345», из пластины № 13, из производственной партии № 59, микросхема с номером № 1234.

Для уникальной идентификации микросхемы необходимо использовать все поля.

7.2 Контроллер тактовых частот (CLKCNTR)

7.2.1 Формирование и управление тактовыми сигналами

Блок управления тактовыми сигналами является важным элементом обеспечения работоспособности микроконтроллера, так как при отсутствии тактовых сигналов микроконтроллер не может выполнять возложенные на него функции. Блок CLKCNTR обеспечивает управление всеми тактовыми сигналами микросхемы. Структурная схема формирования тактовых частот представлена на рисунке .

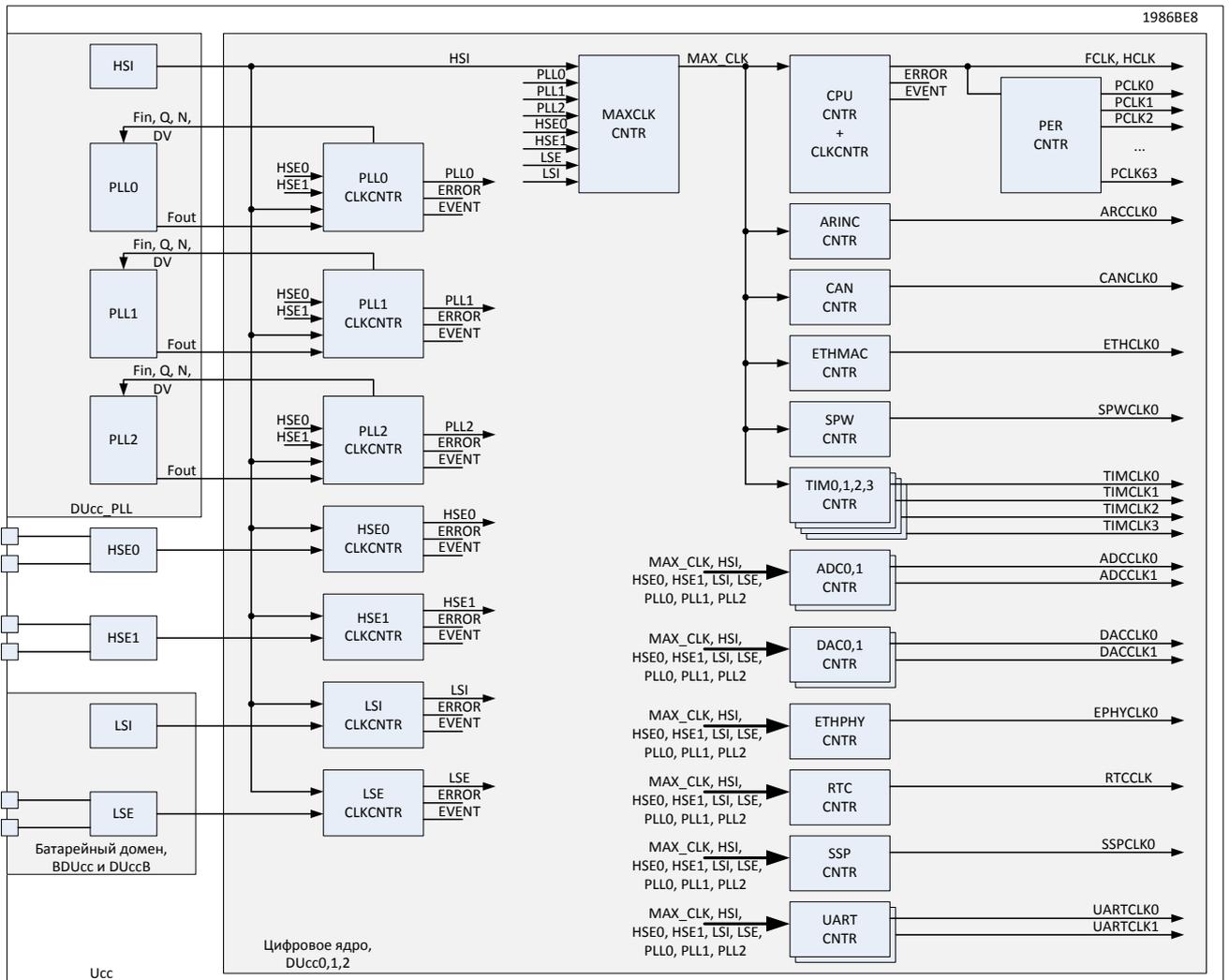


Рисунок 33 – Формирование тактовых частот

При выполнении загрузочной программы и при запуске пользовательской программы микроконтроллер тактируется встроенным генератором HSI. Генератор HSI считается надежным и стабилизированным, отказ генератора HSI приводит к полному отказу микроконтроллера. Включение и переключение на другие источники тактирования, а также разрешение тактирования периферийных блоков осуществляется программно в пользовательской программе. Система тактовой синхронизации построена по принципу полной синхронности основных элементов, связанных с процессорными ядрами (блоки памяти ОЗУ, контроллер DMA, мосты, и интерфейсные по отношению к процессору части всех периферийных блоков). Т.е. эти части микросхемы всегда работают на одной тактовой частоте. Ряд периферийных блоков (CAN, TIMER, MIL, ARINC, ETHERNET) могут быть синхронными процессорному ядру, но при этом могут работать как на большей, так и на меньшей тактовой частоте. В этом случае не требуется дополнительной логики по пересинхронизации. Ряд периферийных блоков (SSP, UART, ADC, DAC, RTC) могут работать на несинхронных частотах. Для обеспечения корректной работы всех синхронных модулей формируется частота MAXCLK. Далее синхронные блоки могут работать на частоте MAXCLK или на прореженной частоте MAXCLK. При этом процессорные ядра и связанные с ним блоки также могут работать на прореженной частоте (все на одной и той же). Принцип формирования и соотношения тактовых частот в микроконтроллере приведен на рисунке .

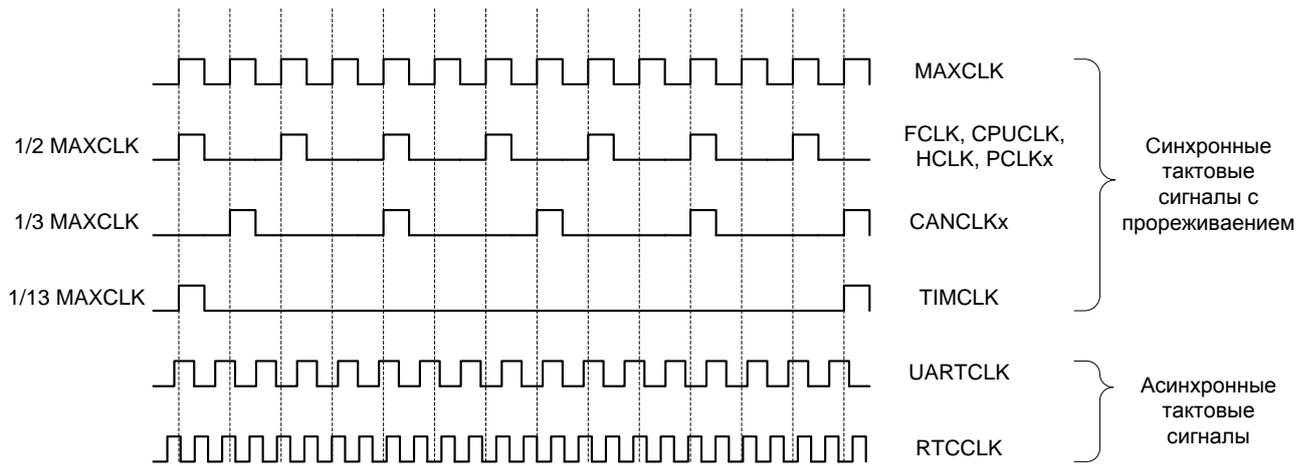


Рисунок 34 – Пример тактовых частот в микроконтроллере

В ряде периферийных блоков (SSP, UART, USB) вводятся ограничения на соотношения рабочих частот этих блоков и частоты PCLK интерфейса с процессорным ядром. Сигнал CPUCLK может совпадать с MAXCLK, но не должен превышать 64 МГц.

7.2.2 Схема контроля тактовых частот

Генератор HSI может использоваться как аварийная замена других источников тактирования. Для этого частота HSI и проверяемая частота от другого источника подается на блок Checker, в котором они сравниваются, и, в зависимости от настроенных критериев, определяются четыре события:

- незначительное увеличение тактовой частоты;
- значительное увеличение тактовой частоты;
- незначительное снижение тактовой частоты;
- значительно снижение (исчезновение) тактовой частоты.

В зависимости от настроек в данных ситуациях может быть выполнено автоматическое переключение на генератор HSI.

Генератор HSI может быть программно отключен (не рекомендуется) для снижения энергопотребления. Отключение генератора HSI выполняется в регистрах управления батарейного домена. Отключение генератора HSI делает невозможным автоматический контроль наличия тактовой частоты от других источников. Включение генератора HSI может быть выполнено программно либо по сигналу сброса по питанию U_{CC}.

Структурная схема блока Checker представлена на рисунке 35.

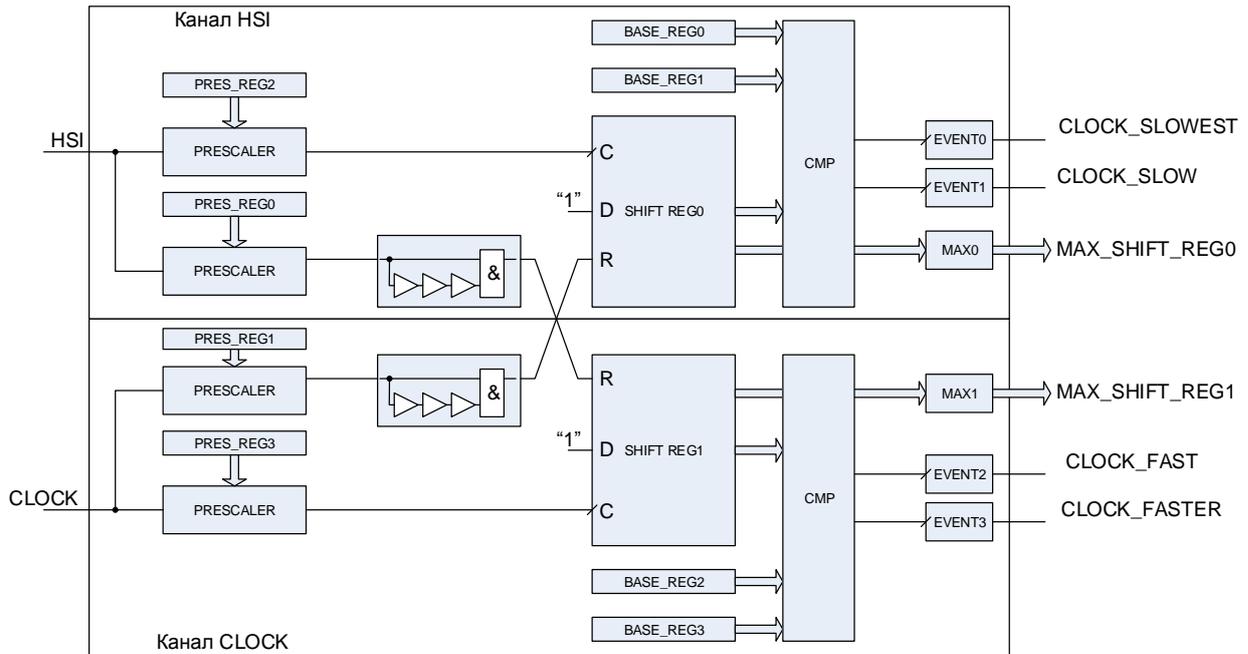


Рисунок 35 – Схема блока контроля тактовых частот

При настройке блока необходимо выбрать предделители сигналов HSI и CLOCK такими, чтобы соотношение полученных частот укладывалось в диапазон от 1 до 15.

$$\text{если } f_{HSI} < f_{CLOCK}, \begin{cases} \frac{f_{HSI}}{\frac{PRES_REG2+1}{f_{CLOCK}}} = K \\ \frac{f_{CLOCK}}{\frac{PRES_REG3+1}{f_{HSI}}} = M \end{cases} \quad (1)$$

$$\text{если } f_{HSI} > f_{CLOCK}, \begin{cases} \frac{f_{HSI}}{\frac{PRES_REG2+1}{f_{CLOCK}}} = K \\ \frac{f_{CLOCK}}{\frac{PRES_REG3+1}{f_{HSI}}} = M \end{cases} \quad (2)$$

где f_{HSI} – частота сигналов HSI;

f_{CLOCK} – частота сигналов CLOCK;

$K \in [1..15]$ – фактическое соотношение частот, вычисляемое в блоке HSI. На схеме обозначено как SHIFT_REG0;

$M \in [1..15]$ – фактическое соотношение частот, вычисляемое в блоке CLOCK. На схеме обозначено как SHIFT_REG1.

K_0 – расчетное соотношение частот K в блоке HSI.

M_0 – расчетное соотношение частот M в блоке CLOCK.

При отклонении частоты CLOCK от расчетного значения коэффициенты K и M отклоняются от своих расчетных значений K_0 и M_0 .

При снижении частоты CLOCK значение K изменяется от K_0 до 15. Максимальное значение K сохраняется в поле MAX_CHK_SHIFT_REG0 соответствующего регистра ..._STAT.

При достижении K пороговых значений, заданных полями BASE_REG0 и BASE_REG1, вырабатываются события EVENT0 и EVENT1 соответственно.

$$\begin{cases} f_{CLOCK} < \frac{f_{HSI} \cdot PRES_REG1^*}{(PRES_REG2+1) \cdot BASE_REG0} \Rightarrow EVENT0 \\ f_{CLOCK} < \frac{f_{HSI} \cdot PRES_REG1^*}{(PRES_REG2+1) \cdot BASE_REG1} \Rightarrow EVENT1 \end{cases} \quad (3)$$

* PRES_REG1 – при $f_{HSI} > f_{CLOCK}$,

PRES_REG1 + 1 – при $f_{HSI} < f_{CLOCK}$.

При увеличении частоты CLOCK значение М изменяется от M0 до 15. Максимальное значение М сохраняется в поле MAX_CHK_SHIFT_REG1 соответствующего регистра ..._STAT.

При достижении М пороговых значений, заданных полями BASE_REG2 и BASE_REG3, вырабатываются события EVENT2 и EVENT3 соответственно.

$$\begin{cases} f_{CLOCK} > \frac{f_{HSI} \cdot (PRES_REG3+1) \cdot BASE_REG2}{PRES_REG0^*} \Rightarrow EVENT2 \\ f_{CLOCK} > \frac{f_{HSI} \cdot (PRES_REG3+1) \cdot BASE_REG3}{PRES_REG0^*} \Rightarrow EVENT3 \end{cases} \quad (4)$$

* PRES_REG0 – при $f_{HSI} < f_{CLOCK}$,

PRES_REG0 + 1 – при $f_{HSI} > f_{CLOCK}$.

Непосредственно в блоке Checker на возникающие события может быть настроено аппаратное переключение на тактовую частоту генератора HSI.

Возникающие события также обрабатываются в блоке FT_CNTR, и на эти события могут быть настроены следующие действия:

- сигнал предупреждения;
- аппаратный сброс микроконтроллера.

Для иллюстрации принципов работы блока возможные комбинации настроек приведены в таблице 23.

Таблица 23 – Примеры работы блока Checker

Частота HSI	Ожидаемая частота CLOCK	Реальная частота CLOCK	PRES REG2	PRES REG0	PRES REG1	PRES REG3	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примеры
8 МГц	80 МГц	80 МГц	1	1	100	1	12	15	12	15	K0 = 10 M0 = 10 Нет событий
8 МГц	80 МГц	100 МГц	1	1	100	1	12	15	12	15	K0 = 10 M0 = 10 K = 8 M = 12 EVENT2Fast Clock
8 МГц	80 МГц	120 МГц	1	1	100	1	12	15	12	15	K0 = 10 M0 = 10 K = 6 M = 15 EVENT2Fast Clock EVENT3Faster Clock
8 МГц	80 МГц	66 МГц	1	1	100	1	12	15	12	15	K0 = 10 M0 = 10 K = 12 M = 7 EVENT0Slow Clock

Частота HSI	Ожидаемая частота CLOCK	Реальная частота CLOCK	PRES REG2	PRES REG0	PRES REG1	PRES REG3	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примеры
8 МГц	80 МГц	0 МГц (нет частоты)	1	1	100	1	12	15	12	15	K0 = 10 M0 = 10 K = 15(∞) M = 0 EVENT0Slow Clock EVENT1Slowest Clock
8 МГц	32 кГц	32 кГц	48	1600	1	1	12	15	12	15	K0 = 5 M0 = 6 Нет событий
8 МГц	32 кГц	60 кГц	48	1600	1	1	12	15	12	15	K0 = 5 M0 = 6 K = 3 M = 12 EVENT 2Fast Clock
8 МГц	32 кГц	75 кГц	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 K = 2 M = 15 EVENT 2Fast Clock EVENT 3Faster Clock
8 МГц	32 кГц	15 кГц	48	1600	1	1	12	15	12	15	K0 = 5 M0 = 6 K = 11 M = 3 EVENT 0Slow Clock
8 МГц	32 кГц	0 кГц (нет частоты)	48	1600	1	1	12	15	12	15	K0 = 5 M0 = 6 K = 15(∞) M = 0 EVENT0Slow Clock EVENT1Slowest Clock

Максимальное достигнутое значение регистра *SHIFT_REG* каждого из каналов можно считать программно. При чтении регистра *SHIFT_REG* считывается теневой регистр, в котором отражается максимальное достигнутое регистром *SHIFT_REG* значение за период с последнего сброса теневого регистра. Сброс теневого регистра осуществляется программно, записью управляющего бита CLR_CHK_SHIFT_REG.

Блок Checker установлен на входах тактовых частот от внешних источников HSE0, HSE1 и LSE, встроенного LSI генератора и всех PLL. Также отдельно стоит блок Checker на выходе схемы формирования тактовых сигналов процессорных ядер.

Блоки Checker проверяют наличие и значения частот тактовых сигналов, поступающих только в цифровую часть микроконтроллера. Блок часов реального времени RTC может быть сконфигурирован на использование тактовой частоты от LSE или LSI генератора, при этом частота на RTC поступает напрямую с генератора и не контролируется блоком Checker. В результате исчезновение частоты LSE приведет к остановке часов реального времени RTC. Этот аспект особенно важен при работе микросхемы в режиме STANDBY с пробуждением от RTC. Так как в этом случае при отказе генератора и остановке часов пробуждение будет невозможно.

Частота LSI генератора поступает напрямую на схему формирования задержки $T_{PORESETn}$, но в блоке формирования сигнала UccRESET аппаратно контролируется формирование этой задержки, и в случае отсутствия частоты сигнал UccRESET будет сгенерирован, но с много меньшей задержкой.

7.2.3 Описание регистров

Таблица 24 – Описание регистров

Базовый адрес	Смещение	Обозначение	Описание
0x4000_0000	0x0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	MAX_CLK	Регистр настройки частоты MAX_CLK
Тактирование процессорного ядра			
	0x0000_0008	CPU_CLK	Регистр управления тактовой частотой процессорных ядер
	0x0000_000C	PER0_CLK	Регистр управления тактированием периферии
	0x0000_0010	PER1_CLK	Регистр управления тактированием периферии
	0x0000_0014	CPU_CHK0	Регистр настройки монитора процессорной частоты CPU_CLK, регистр PRES_REG0, PRES_REG1
	0x0000_0018	CPU_CHK1	Регистр настройки монитора процессорной частоты CPU_CLK, регистр PRES_REG2, PRES_REG3
	0x0000_001C	CPU_CHK2	Регистр настройки монитора процессорной частоты CPU_CLK, регистр BASE_REG0,1,2,3
	0x0000_0020	CPU_STAT	Регистр состояния тактовой частоты процессорных ядер
Генератор LSI			
	0x0000_0024	LSI_CLK	Регистр настройки частоты LSI
	0x0000_0028	LSI_CHK0	Регистр настройки монитора частоты LSI, регистр PRES_REG0
	0x0000_002C	LSI_CHK1	Регистр настройки монитора частоты LSI, регистр PRES_REG1
	0x0000_0030	LSI_CHK2	Регистр настройки монитора частоты LSI, регистр BASE_REG0
	0x0000_0034	LSI_STAT	Регистр состояния тактовой частоты LSI
Генератор HSI			
	0x0000_0038	HSI_STAT	Регистр состояния тактовой частоты HSI
Генератор LSE			
	0x0000_003C	LSE_CLK	Регистр настройки частоты LSE
	0x0000_0040	LSE_CHK0	Регистр настройки монитора частоты LSE, регистр PRES_REG0
	0x0000_0044	LSE_CHK1	Регистр настройки монитора частоты LSE, регистр PRES_REG1
	0x0000_0048	LSE_CHK2	Регистр настройки монитора частоты LSE, регистр BASE_REG0
	0x0000_004C	LSE_STAT	Регистр состояния тактовой частоты LSE
Генератор HSE0			
	0x0000_0050	HSE0_CLK	Регистр настройки генератора HSE0
	0x0000_0054	HSE0_CHK0	Регистр настройки монитора частоты HSE0, регистр PRES_REG0
	0x0000_0058	HSE0_CHK1	Регистр настройки монитора частоты HSE0, регистр PRES_REG1
	0x0000_005C	HSE0_CHK2	Регистр настройки монитора частоты HSE0, регистр BASE_REG0
	0x0000_0060	HSE0_STAT	Регистр состояния тактовой частоты HSE0

Базовый адрес	Смещение	Обозначение	Описание
Генератор HSE1			
	0x0000_0064	HSE1_CLK	Регистр настройки генератора HSE1
	0x0000_0068	HSE1_CHK0	Регистр настройки монитора частоты HSE1, регистр PRES_REG0
	0x0000_006C	HSE1_CHK1	Регистр настройки монитора частоты HSE1, регистр PRES_REG1
	0x0000_0070	HSE1_CHK2	Регистр настройки монитора частоты HSE1, регистр BASE_REG0
	0x0000_0074	HSE1_STAT	Регистр состояния тактовой частоты HSE1
Блок умножения PLL0			
	0x0000_0078	PLL0_CLK	Регистр настройки PLL0
	0x0000_007C	PLL0_CHK0	Регистр настройки монитора частоты PLL0, регистр PRES_REG0
	0x0000_0080	PLL0_CHK1	Регистр настройки монитора частоты PLL0, регистр PRES_REG1
	0x0000_0084	PLL0_CHK2	Регистр настройки монитора частоты PLL0, регистр BASE_REG0
	0x0000_0088	PLL0_STAT	Регистр состояния тактовой частоты PLL0
Блок умножения PLL1			
	0x0000_008C	PLL1_CLK	Регистр настройки PLL1
	0x0000_0090	PLL1_CHK0	Регистр настройки монитора частоты PLL1, регистр PRES_REG0
	0x0000_0094	PLL1_CHK1	Регистр настройки монитора частоты PLL1, регистр PRES_REG1
	0x0000_0098	PLL1_CHK2	Регистр настройки монитора частоты PLL1, регистр BASE_REG0
	0x0000_009C	PLL1_STAT	Регистр состояния тактовой частоты PLL1
Блок умножения PLL2			
	0x0000_00A0	PLL2_CLK	Регистр настройки PLL2
	0x0000_00A4	PLL2_CHK0	Регистр настройки монитора частоты PLL2, регистр PRES_REG0
	0x0000_00A8	PLL2_CHK1	Регистр настройки монитора частоты PLL2, регистр PRES_REG1
	0x0000_00AC	PLL2_CHK2	Регистр настройки монитора частоты PLL2, регистр BASE_REG0
	0x0000_00B0	PLL2_STAT	Регистр состояния тактовой частоты PLL2
	0x00B4 – 0x0114		Зарезервировано
Тактирование CAN			
	0x0000_0118	CAN0_CLK	Управление тактированием контроллера CAN0
	0x0000_011C – 0x0000_012C		Зарезервировано
Тактирование TIMER			
	0x0000_0130	TIM0_CLK	Управление тактированием контроллера TIMER0
	0x0000_0134	TIM1_CLK	Управление тактированием контроллера TIMER1
	0x0000_0138	TIM2_CLK	Управление тактированием контроллера TIMER2
	0x0000_013C	TIM3_CLK	Управление тактированием контроллера TIMER3

Базовый адрес	Смещение	Обозначение	Описание
	0x0000_0140 – 0x0000_0144		Зарезервировано
Тактирование MIL (МКПД)			
	0x0000_0148 – 0x0000_0154		Зарезервировано
Тактирование ARINC-249			
	0x0000_0158 – 0x0000_0164		Зарезервировано
Тактирование ETHERNET			
	0x0000_0168 – 0x0000_016C		Зарезервировано
	0x0000_0170	EPHY0_CLK	Управление тактированием контроллера EthernetPHY0
	0x0000_0174		Зарезервировано
Тактирование SPACE WIRE			
	0x0000_0178	SPW0_CLK	Управление тактированием контроллера SpaceWire0
	0x0000_017C – 0x0184		Зарезервировано
Тактирование UART			
	0x0000_0188	UART0_CLK	Управление тактированием контроллера UART0
	0x0000_018C	UART1_CLK	Управление тактированием контроллера UART1
	0x0000_0190 – 0x019C		Зарезервировано
Тактирование SSP			
	0x0000_01A0	SSP0_CLK	Управление тактированием контроллера SSP0
	0x01B8		Зарезервировано
Тактирование ADC			
	0x0000_01BC	ADC0_CLK	Управление тактированием контроллера ADC0
	0x0000_01C0	ADC1_CLK	Управление тактированием контроллера ADC1
Тактирование DAC			
	0x0000_01C4	DAC0_CLK	Управление тактированием контроллера DAC0
	0x0000_01C8	DAC1_CLK	Управление тактированием контроллера DAC1
Тактирование RTC			
	0x0000_01CC	RTC_CLK	Управление тактированием контроллера RTC

7.2.3.1 Регистр KEY

Base ADDR=	0x4000_0000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока CLKCNTNTR

7.2.3.2 Регистр MAX_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0004							
REG Name:		MAX_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Зарезервировано															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Зарезервировано												SELECT[3:0]			

Бит	Имя	Значение	Описание
31...4	-	-	Зарезервировано
3...0	SELECT[3:0]		Номер источника тактовой частоты 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSI/2 4'b 00010 – Генератор HSE0 4'b 00011 – Генератор HSE0/2 4'b 00100 – Генератор HSE1 4'b 00101 – Генератор HSE1/2 4'b 00110 – Генератор LSI 4'b 00111 – Генератор LSE 4'b 01000 – Генератор PLL0 4'b 01001 – Генератор PLL1 4'b 01010 – Генератор PLL2 4'b 01011 – Зарезервировано 4'b 01100 – Зарезервировано 4'b 01101 – Зарезервировано 4'b 01110 – Зарезервировано 4'b 01111 – Зарезервировано

7.2.3.3 Регистр CPU_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0008							
REG Name:		CPU_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					CLR_CHK_EVENT3	CLR_CHK_EVENT2	CLR_CHK_EVENT1	CLR_CHK_EVENT0	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...27	-	-	Зарезервировано
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты FCLK блоком CLK_CHECKER 0 – частота не контролируется 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота) 0 – ничего не делать при событии EVENT3 1 – перейти на частоту HSI при событии EVENT3 Имеет смысл только при EN_CHK = 1 Для FCLK аварийное переключение не предусмотрено
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота) 0 – ничего не делать при событии EVENT2 1 – перейти на частоту HSI при событии EVENT2 Имеет смысл только при EN_CHK = 1 Для FCLK аварийное переключение не предусмотрено
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота) 0 – ничего не делать при событии EVENT1 1 – перейти на частоту HSI при событии EVENT1 Имеет смысл только при EN_CHK = 1 Для FCLK аварийное переключение не предусмотрено
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты) 0 – ничего не делать при событии EVENT0 1 – перейти на частоту HSI при событии EVENT0 Имеет смысл только при EN_CHK = 1 Для FCLK аварийное переключение не предусмотрено
15...0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты FCLK(CPU_CLK). FCLK = MAX_CLK/(DIV+1)

7.2.3.4 Регистр PER0_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_000C							
REG Name:		PER0_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
.	.	MDR_CAN0_EN	.	.	MDR_TMR3_EN	MDR_TMR2_EN	MDR_TMR1_EN	MDR_TMR0_EN	.	MDR_SPW0_EN	.	.	.	MDR_PORTE_EN	MDR_PORTD_EN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDR_PORTC_EN	MDR_PORTB_EN	MDR_PORTA_EN

Бит	Имя	Значение	Описание
31	-		
30	-		
29	MDR_CAN0_EN		Бит разрешения тактирования частотой CAN0 0 – тактирование запрещено 1 – тактирование разрешено
28	-		
27	-		
26	MDR_TMR3_EN		Бит разрешения тактирования частотой TMR3 0 – тактирование запрещено 1 – тактирование разрешено
25	MDR_TMR2_EN		Бит разрешения тактирования частотой TMR2 0 – тактирование запрещено 1 – тактирование разрешено
24	MDR_TMR1_EN		Бит разрешения тактирования частотой TMR1 0 – тактирование запрещено 1 – тактирование разрешено
23	MDR_TMR0_EN		Бит разрешения тактирования частотой TMR0 0 – тактирование запрещено 1 – тактирование разрешено
22	-		
21	MDR_SPW0_EN		Бит разрешения тактирования частотой SpaceWire 0 – тактирование запрещено 1 – тактирование разрешено
20	-		
19	-		
18	-		
17	MDR_PORTE_EN		Бит разрешения тактирования частотой PORTE 0 – тактирование запрещено 1 – тактирование разрешено
16	MDR_PORTD_EN		Бит разрешения тактирования частотой PORTD 0 – тактирование запрещено 1 – тактирование разрешено
15	MDR_PORTC_EN		Бит разрешения тактирования частотой PORTC 0 – тактирование запрещено 1 – тактирование разрешено
14	MDR_PORTB_EN		Бит разрешения тактирования частотой PORTB 0 – тактирование запрещено 1 – тактирование разрешено
13	MDR_PORTA_EN		Бит разрешения тактирования частотой PORTA 0 – тактирование запрещено 1 – тактирование разрешено
12...0	-		

7.2.3.5 Регистр PER1_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0010									
REG Name:		PER1_CLK															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
					MDR_DAC1_EN	MDR_DAC0_EN	MDR_ADC1_EN	MDR_ADC0_EN		MDR_MIL1_EN		MDR_MIL0_EN					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
		MDR_ARCTX0_EN		MDR_ARCRX0_EN				MDR_UART1_EN	MDR_UART0_EN				MDR_SSP0_EN				

Бит	Имя	Значение	Описание
31...27	-		
26	MDR_DAC1_EN		Бит разрешения тактирования частотой DAC1 0 – тактирование запрещено 1 – тактирование разрешено
25	MDR_DAC0_EN		Бит разрешения тактирования частотой DAC0 0 – тактирование запрещено 1 – тактирование разрешено
24	MDR_ADC1_EN		Бит разрешения тактирования частотой ADC1 0 – тактирование запрещено 1 – тактирование разрешено
23	MDR_ADC0_EN		Бит разрешения тактирования частотой ADC0 0 – тактирование запрещено 1 – тактирование разрешено
22	-		
21	MDR_MIL1_EN		Бит разрешения тактирования частотой MIL1 0 – тактирование запрещено 1 – тактирование разрешено
20	-		
19	MDR_MIL0_EN		Бит разрешения тактирования частотой MIL0 0 – тактирование запрещено 1 – тактирование разрешено
18...14	-		
13	MDR_ARCTX0_EN		Бит разрешения тактирования частотой ARCTX0 0 – тактирование запрещено 1 – тактирование разрешено
12	-		
11	MDR_ARCRX0_EN		Бит разрешения тактирования частотой ARCRX0 0 – тактирование запрещено 1 – тактирование разрешено
10...8	-		
7	MDR_UART1_EN		Бит разрешения тактирования частотой UART1 0 – тактирование запрещено 1 – тактирование разрешено
6	MDR_UART0_EN		Бит разрешения тактирования частотой UART0 0 – тактирование запрещено 1 – тактирование разрешено
5...3			
2	MDR_SSP0_EN		Бит разрешения тактирования частотой SSP0 0 – тактирование запрещено 1 – тактирование разрешено
1...0	-		

7.2.3.6 Регистр CPU_CHK0

Base ADDR=	0x4000_0000	Offset=	0x0000_0014												
REG Name:	CPU_CHK0														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1 RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15...0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0 CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.7 Регистр CPU_CHK1

Base ADDR=	0x4000_0000	Offset=	0x0000_0018												
REG Name:	CPU_CHK1														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0 RST(SHIFT_REG0) = FCLK / (PRES_REG3+1)
15...0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1 CLK(SHIFT_REG1) = FCLK / (PRES_REG1+1)

7.2.3.8 Регистр CPU_CHK2

Base ADDR=		0x4000_0000				Offset=		0x0000_001C							
REG Name:		CPU_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_REG3[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.9 Регистр CPU_STAT

Base ADDR=		0x4000_0000				Offset=		0x0000_0020							
REG Name:		CPU_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												EVENT3	EVENT2	EVENT1	EVENT0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_CHK_SHIFT_REG1[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота) 0 – нет события 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота) 0 – нет события 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота) 0 – нет события 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты) 0 – нет события 1 – есть событие
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.10 Регистр LSI_CLK*

Base ADDR=		0x4000_0000				Offset=		0x0000_0024									
REG Name:		LSI_CLK															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
					1	0		0	0	0	ш Z	ш Z	ш Z	ш Z	ш Z		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
-																	

Бит	Имя	Значение	Описание
31...27	-	-	Зарезервировано
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты LSI 0 – частота не контролируется 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота) 0 – ничего не делать при событии EVENT3 1 – перейти на частоту HSI при событии EVENT3 Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота) 0 – ничего не делать при событии EVENT2 1 – перейти на частоту HSI при событии EVENT2 Имеет смысл только при EN_CHK = 1

* Управление генератором LSI осуществляется через регистры батарейного домена.

Бит	Имя	Значение	Описание
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота) 0 – ничего не делать при событии EVENT1 1 – перейти на частоту HSI при событии EVENT1 Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты) 0 – ничего не делать при событии EVENT0 1 – перейти на частоту HSI при событии EVENT0 Имеет смысл только при EN_CHK = 1
15...0	-		Зарезервировано

7.2.3.11 Регистр LSI_CHK0

Base ADDR=	0x4000_0000	Offset=	0x0000_0028												
REG Name:	LSI_CHK0														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1 $RST(SHIFT_REG1) = HSI / (PRES_REG0+1)$
15...0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0 $CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)$

7.2.3.12 Регистр LSI_CHK1

Base ADDR=	0x4000_0000	Offset=	0x0000_002C												
REG Name:	LSI_CHK1														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0 RST(SHIFT_REG0) = LSI / (PRES_REG3+1)
15...0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1 CLK(SHIFT_REG1) = LSI / (PRES_REG1+1)

7.2.3.13 Регистр LSI_CHK2

Base ADDR=	0x4000_0000	Offset=	0x0000_0030												
REG Name:	LSI_CHK2														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_REG3[7:0]								BASE_REG2[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.14 Регистр LSI_STAT

Base ADDR=		0x4000_0000				Offset=		0x0000_0034								
REG Name:		LSI_STAT														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											LSI_RDY	EVENT3	EVENT2	EVENT1	EVENT0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_CHK_SHIFT_REG1[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...21	-	-	Зарезервировано
20	LSI_READY	-	Флаг готовности генератора LSI 0 – генератор не готов или выключен 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота) 0 – нет события 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота) 0 – нет события 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота) 0 – нет события 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты) 0 – нет события 1 – есть событие
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.15 Регистр HSI_STAT*

Base ADDR=		0x4000_0000				Offset=		0x0000_0038								
REG Name:		HSI_STAT														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											HSI_RDY					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Бит	Имя	Значение	Описание
31...21	-	-	Зарезервировано
20	HSI_READY	-	Флаг готовности генератора HSI 0 – генератор не готов или выключен 1 – генератор готов
19...0	-	-	Зарезервировано

* Управление генератором HSI осуществляется через регистры батарейного домена.

7.2.3.16 Регистр LSE_CLK*

Base ADDR=		0x4000_0000				Offset=		0x0000_003C							
REG Name:		LSE_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					LSE_READY			CLR_CHK_EVENT	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Бит	Имя	Значение	Описание
31...27	-	-	Зарезервировано
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 Запись 1 сбрасывает теневой регистр
20	EN_CHK		Бит разрешения контроля частоты LSE 0 – частота не контролируется 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота) 0 – ничего не делать при событии EVENT3 1 – перейти на частоту HSI при событии EVENT3 Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота)

* Управление генератором LSE осуществляется через регистры батарейного домена.

Бит	Имя	Значение	Описание
			0 – ничего не делать при событии EVENT2 1 – перейти на частоту HSI при событии EVENT2 Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота) 0 – ничего не делать при событии EVENT1 1 – перейти на частоту HSI при событии EVENT1 Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты) 0 – ничего не делать при событии EVENT0 1 – перейти на частоту HSI при событии EVENT0 Имеет смысл только при EN_CHK = 1
15...0	-		Зарезервировано

7.2.3.17 Регистр LSE_CHK0

Base ADDR=	0x4000_0000	Offset=	0x0000_0040												
REG Name:	LSE_CHK0														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1 $RST(SHIFT_REG1) = HSI / (PRES_REG0+1)$
15...0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0 $CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)$

7.2.3.18 Регистр LSE_CHK1

Base ADDR=		0x4000_0000				Offset=		0x0000_0044							
REG Name:		LSE_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0 RST(SHIFT_REG0) = LSE / (PRES_REG3+1)
15...0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1 CLK(SHIFT_REG1) = LSE / (PRES_REG1+1)

7.2.3.19 Регистр LSE_CHK2

Base ADDR=		0x4000_0000				Offset=		0x0000_0048							
REG Name:		LSE_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_REG3[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.20 Регистр LSE_STAT

Base ADDR=		0x4000_0000				Offset=		0x0000_004C								
REG Name:		LSE_STAT														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											LSE_RDY	EVENT3	EVENT2	EVENT1	EVENT0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_CHK_SHIFT_REG1[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...20	-	-	Зарезервировано
20	LSE_READY	-	Флаг готовности генератора LSE 0 – генератор не готов или выключен 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота) 0 – нет события 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота) 0 – нет события 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота) 0 – нет события 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты) 0 – нет события 1 – есть событие
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.21 Регистр HSEn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0050							
REG Name:		HSEn_CLK						0x0000_0064							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		HSE_FILTER_EN	HSEn_BYP	HSEn_ON	CLR_CHK_EVENT3	CLR_CHK_EVENT2	CLR_CHK_EVENT1	CLR_CHK_EVENT0	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Бит	Имя	Значение	Описание
31...30	-	-	Зарезервировано
29	HSE_FILTER_EN	-	Бит разрешения встроенного фильтра высоких частот 0 – фильтр отключен 1 – фильтр включен
28	HSEn_BYP	-	Бит выбора режима работы генератора HSEn 0 – режим работы с внешним резонатором 1 – режим работы с внешним генератором
27	HSEn_ON	-	Бит разрешения работы генератора HSEn 0 – генератор выключен 1 – генератор включен
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты HSEn 0 – частота не контролируется 1 – монитор работает

19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота) 0 – ничего не делать при событии EVENT3 1 – перейти на частоту HSI при событии EVENT3 Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота) 0 – ничего не делать при событии EVENT2 1 – перейти на частоту HSI при событии EVENT2 Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота) 0 – ничего не делать при событии EVENT1 1 – перейти на частоту HSI при событии EVENT1 Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты) 0 – ничего не делать при событии EVENT0 1 – перейти на частоту HSI при событии EVENT0 Имеет смысл только при EN_CHK = 1
15...0	-		Зарезервировано

7.2.3.22 Регистр HSEn_CHK0

Base ADDR=	0x4000_0000	Offset=	0x0000_0054 0x0000_0068												
REG Name:	HSEn_CHK0														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1 $RST(SHIFT_REG1) = HSI / (PRES_REG0+1)$
15...0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0 $CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)$

7.2.3.23 Регистр HSEn_CHK1

Base ADDR=	0x4000_0000	Offset=	0x0000_0058 0x0000_006C												
REG Name:	HSEn_CHK1														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Бит	Имя	Значение	Описание
31...16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0 RST(SHIFT_REG0) = HSEn / (PRES_REG3+1)
15...0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1 CLK(SHIFT_REG1) = HSEn / (PRES_REG1+1)

7.2.3.24 Регистр HSEn_CHK2

Base ADDR=	0x4000_0000	Offset=	0x0000_005C 0x0000_0070												
REG Name:	HSEn_CHK2														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_REG3[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.25 Регистр HSEn_STAT

Base ADDR=		0x4000_0000					Offset=		0x0000_0060 0x0000_0074										
REG Name:		HSEn_STAT																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
											HSEn_RDY	EVENT3	EVENT2	EVENT1	EVENT0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
MAX_CHK_SHIFT_REG1[7:0]								MAX_CHK_SHIFT_REG0[7:0]															

Бит	Имя	Значение	Описание
31...21	-		Зарезервировано
20	HSEn_RDY	-	Флаг готовности генератора HSEn 0 – генератор не готов или выключен 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота) 0 – нет события 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота) 0 – нет события 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота) 0 – нет события 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты) 0 – нет события 1 – есть событие
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.26 Регистр PLLn_CLK

Base ADDR=	0x4000_0000					Offset=	0x0000_0078 0x0000_008C 0x0000_00A0										
REG Name:	PLLn_CLK																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
SELECT[2:0]			PLLn_RLD	PLLn_ON	CLR_CHK_EVENT3	CLR_CHK_EVENT2	CLR_CHK_EVENT1	CLR_CHK_EVENT0	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PLLn_N[7:0]								PLLn_Q[7:0]									

Бит	Имя	Значение	Описание
31...29	SELECT[2:0]		Номер источника тактовой частоты 3'b 000 – Генератор HSI 3'b 001 – Генератор HSI/2 3'b 010 – Генератор HSE0 3'b 011 – Генератор HSE0/2 3'b 100 – Генератор HSE1 3'b 101 – Генератор HSE1/2 3'b 110 и 111 – Зарезервировано
28	PLLn_RLD	-	Бит перезапуска работы PLLn При смене коэффициентов умножения/деления запись 1 в этот бит перезапускает PLLn, при этом сбрасывается флаг готовности PLLn_RDY и только после выхода на режим флаг взводится обратно.
27	PLLn_ON	-	Бит разрешения работы генератора PLLn 0 – PLLn выключена 1 – PLLn включена
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита

22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты PLLn 0 – частота не контролируется 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота) 0 – ничего не делать при событии EVENT3 1 – перейти на частоту HSI при событии EVENT3 Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота) 0 – ничего не делать при событии EVENT2 1 – перейти на частоту HSI при событии EVENT2 Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота) 0 – ничего не делать при событии EVENT1 1 – перейти на частоту HSI при событии EVENT1 Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты) 0 – ничего не делать при событии EVENT0 1 – перейти на частоту HS при событии EVENT0 Имеет смысл только при EN_CHK = 1
15	-		Зарезервировано
14...8	PLLn_N[6:0]		Коэффициент умножения тактовой частоты PLLn - K_{NPLL} $FINT = FIN * (K_{NPLL}) / (K_{QPLL} + 1)$ Значение PLLn_N может быть в диапазоне от 3 до 75. Частота FINT, полученная после умножения FIN на K_{NPLL} и деления на $(K_{QPLL} + 1)$, должна быть в диапазоне от 75 до 150 МГц. При PLLn_N = 0 соответствует умножению на 2
7...5	-		Зарезервировано
4	DV		Бит дополнительного деления PLL $FOUT = FINT / (DV + 1)$
3...0	PLLn_Q[3:0]		Коэффициент деления тактовой частоты PLLn - K_{QPLL} $FINT = FIN * (K_{NPLL}) / (K_{QPLL} + 1)$ Значение PLLn_Q может быть в диапазоне от 0 до 15

7.2.3.27 Регистр PLLn_CHK0

Base ADDR=	0x4000_0000					Offset=	0x0000_007C 0x0000_0090 0x0000_00A4										
REG Name:	PLLn_CHK0																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
PRES_REG0[15:0]																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PRES_REG2[15:0]																	

Бит	Имя	Значение	Описание
31...16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1 $RST(SHIFT_REG1) = HSI / (PRES_REG0+1)$
15...0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0 $CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)$

7.2.3.28 Регистр PLLn_CHK1

Base ADDR=	0x4000_0000					Offset=	0x0000_0080 0x0000_0094 0x0000_00A8										
REG Name:	PLLn_CHK1																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
PRES_REG3[15:0]																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PRES_REG1[15:0]																	

Бит	Имя	Значение	Описание
31...16	PRES_REG3[15:0]		Значение регистра делителя частоты PLLn сброса для счетчика SHIFT_REG0 $RST(SHIFT_REG0) = PLLn / (PRES_REG3+1)$
15...0	PRES_REG1[15:0]		Значение регистра делителя частоты PLLn для счетчика на SHIFT_REG1 $CLK(SHIFT_REG1) = PLLn / (PRES_REG1+1)$

7.2.3.29 Регистр PLLn_CHK2

Base ADDR=	0x4000_0000	Offset=	0x0000_0084 0x0000_0098 0x0000_00AC												
REG Name:	PLLn_CHK2														

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-				BASE_REG3[3:0]				-				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				BASE_REG1[3:0]				-				BASE_REG0[3:0]			

Бит	Имя	Значение	Описание
27...24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...20	-		Зарезервировано
19...16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...12	-		Зарезервировано
11...8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...4	-		Зарезервировано
3...0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.30 Регистр PLLn_STAT

Base ADDR=		0x4000_0000				Offset=		0x0000_0088 0x0000_009C 0x0000_00B0							
REG Name:		PLLn_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PLLn_RDY	EVENT3	EVENT2	EVENT1	EVENT0
MAX_CHK_SHIFT_REG1[7:0]								MAX_CHK_SHIFT_REG0[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Бит	Имя	Значение	Описание
31...21	-		Зарезервировано
20	PLLn_READY	-	Флаг готовности генератора PLLn 0 – генератор не готов или выключен 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота) 0 – нет события 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота) 0 – нет события 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота) 0 – нет события 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты) 0 – нет события 1 – есть событие
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.31 Регистр CANn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0118							
REG Name:		CANn_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты CANCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты CANCLKn CANCLKn = MAX_CLK/(DIV+1)

7.2.3.32 Регистр TIMn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0130 0x0000_0134 0x0000_0138 0x0000_013C							
REG Name:		TIMn_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для TIMCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты TIMCLKn TIMCLKn = MAX_CLK/(DIV+1)

7.2.3.33 Регистр EPHY_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0170								
REG Name:		EPHY_CLK														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SELECT[3:0]														EN_CLK		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для EthernetPHY 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты EPHYCLK EPHYCLK = SELCLK/(DIV+1)

7.2.3.34 Регистр SPW0_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0178									
REG Name:		SPW0_CLK															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
																EN_CLK	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DIV[15:0]																

Бит	Имя	Значение	Описание
31...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для SPWCLK0 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты SPWCLK0 $SPWCLK0 = MAX_CLK / (DIV + 1)$

7.2.3.35 Регистр UARTn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0188							
								0x0000_018C							
REG Name:		UARTn_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для UARTCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты UARTCLKn UARTCLKn = SELCLK/(DIV+1)

7.2.3.36 Регистр SSP0_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_01A0							
REG Name:		SSP0_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для SSPCLK0 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты SSPCLK0 SSPCLK0 = SELCLK/(DIV+1)

7.2.3.37 Регистр ADCn_CLK

Base ADDR=	0x4000_0000				Offset=	0x0000_01BC 0x0000_01C0									
REG Name:	ADCn_CLK														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 0000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для ADCCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты ADCCLKn ADCCLKn = SELCLK/(DIV+1) Деление частоты выполняется путем «прореживания» тактовых импульсов. Необходимо учитывать требование к минимальной длительности тактового импульса

7.2.3.38 Регистр DACn_CLK

Base ADDR=	0x4000_0000				Offset=	0x0000_01C4 0x0000_01C8									
REG Name:	DACn_CLK														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]														EN_CLK	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для DACCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты DACCLKn ADCCLKn = SELCLK/(DIV+1)

7.2.3.39 Регистр RTC_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_01CC							
REG Name:		RTC_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 4'b 00000 – Генератор HSI 4'b 00001 – Генератор HSE0 4'b 00010 – Генератор HSE1 4'b 00011 – Генератор LSI 4'b 00100 – Генератор LSE 4'b 00101 – Генератор PLL0 4'b 00110 – Генератор PLL1 4'b 00111 – Генератор PLL2 4'b 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для RTCCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты RTCCLKn RTCCLKn = SELCLK/(DIV+1)

7.3 Контроллер батарейного домена (ВКРСNTR)

В микроконтроллере реализован батарейный домен, включающий в себя:

- 240 байт пользовательской памяти ВКР_МЕМ,
- четыре регистра управления,
- часы реального времени RTC,
- генератор LSE.

Регистры управления ВКР_МЕМ предназначены для хранения и управления режимами работы микроконтроллера, в том числе управление питанием и т.п. При отсутствии питания U_{сс} питание батарейного домена автоматически переключается на питание от вывода ВU_{сс}, и также автоматически переключается на питание U_{сс} при его появлении.

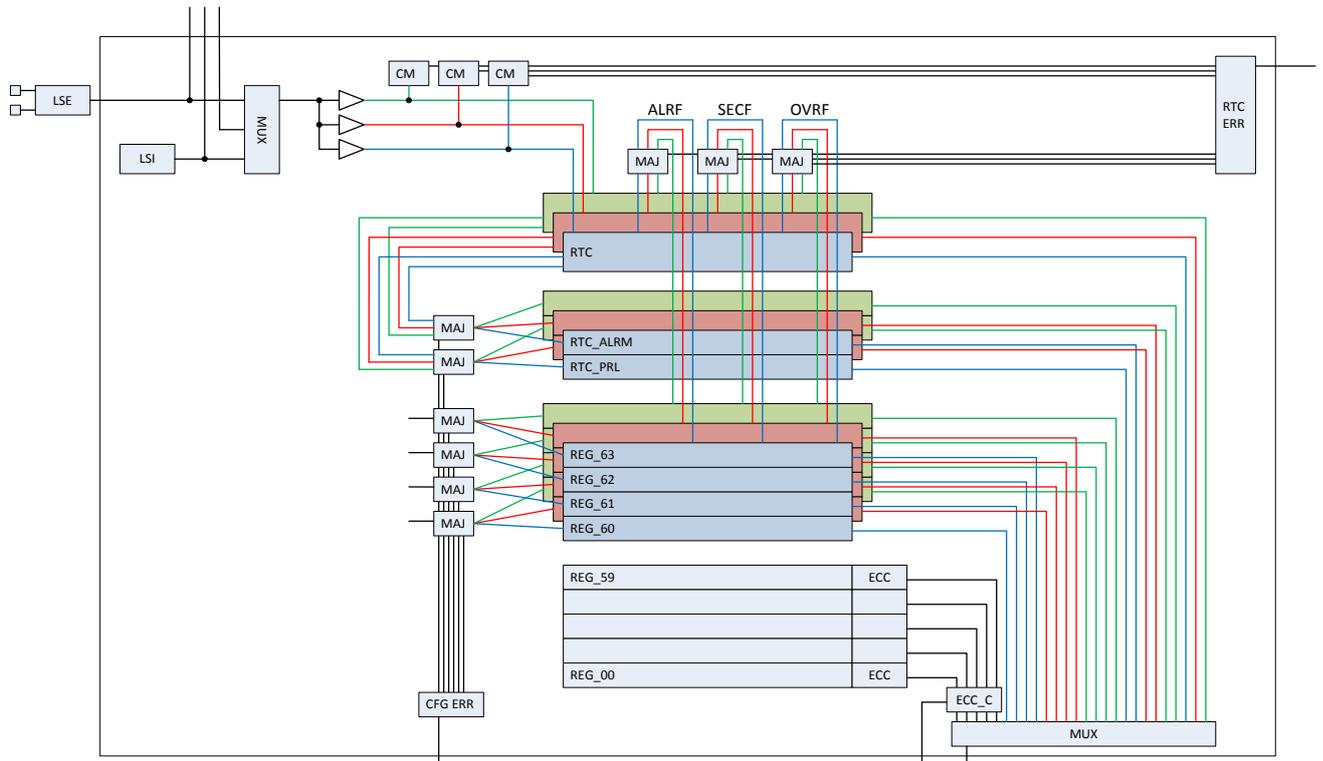


Рисунок 36 – Блок-схема контроллера батарейного домена

В батарейном домене применяются различные механизмы обнаружения и исправления сбоев. Пользовательская память использует биты ECC для автоматического исправления одиночных ошибок и обнаружения двойных. Основная конфигурация и часы реального времени выполнены на троированной логике с непрерывным контролем возникновения ошибок в схемах мажорирования. Мажорирование не исправляет сбоев непосредственно в том канале, в котором он возник, а исправляет выход схемы, влияющий на поведение микросхемы в целом. Таким образом, при возникновении сбоя в одном из каналов схема мажорирования исправит его влияние на систему в целом и будет его обозначать флагом ошибки, вплоть до его исправления программными средствами. Кроме того, для некоторых битов управления, например, STANDBY, действует не мажорированная логика (2 из 3), а логика 3 из 3. Т.е. для того, чтобы перевести микросхему в режим STANDBY необходимо, чтобы бит STANDBY был взведен во всех трех каналах. Если хотя бы в одном из регистров бит не будет взведен или ошибочно сброшен в ноль, то микросхема автоматически выходит из режима STANDBY.

7.3.1 Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTC_EN. В качестве источника тактовой частоты часов реального времени могут выступать генераторы LSI или LSE, либо сигнал RTC_CLK, который формируется в блоке управления тактовыми частотами и может быть выбран только при наличии питания DUcc. LSI может быть выбран при наличии питания Ucc, LSE может быть выбран при наличии Ucc или BUcc. Выбор между источниками осуществляется битами RTC_SEL. При возможном отключении основного источника питания UCC в качестве источника тактовой частоты должен использоваться осциллятор LSE, так как он имеет батарейное питание. Биты управления осциллятором LSE расположены в батарейном домене и, таким образом, при отключении основного питания они не сбрасываются.

Для калибровки тактовой частоты используются биты CAL[7:0]. Значение CAL определяет, какое число тактов из 2^{20} будет замаскировано. Таким образом, с помощью бит CAL[7:0] производится замедление хода часов. Изменение значения бит CAL может быть осуществлено в ходе работы часов реального времени.

Регистр RTC_DIV выступает в роли 20-ти битного предварительного делителя входной тактовой частоты, таким образом, чтобы на его выходе была тактовая частота в 1 Гц. Для задания коэффициента деления регистра RTC_DIV используется регистр RTC_PRL.

Регистр RTC_CNT предназначен для отсчета времени в секундах и работает на выходной частоте делителя RTC_DIV. Регистр RTC_ALR предназначен для задания времени, при совпадении с которым вырабатывается флаг прерывания и пробуждения процессора. Бит STANDBY, отключающий все внутренние регуляторы напряжения, автоматически сбрасывается при превышении RTC_CNT значения в регистре RTC_ALRM.

7.3.2 Описание регистров

Таблица 25 – Описание регистров

Базовый адрес	Смещение	Название	Описание
0x4000_1000	0x0000_00F0	KEY	Регистр ключа, разрешающего модификацию остальных регистров
Память батарейного домена			
	0x0000_0000	REG_00	Регистр батарейной памяти 00
	0x0000_0004	REG_01	Регистр батарейной памяти 01
	0x0000_00EC	REG_59	Регистр батарейной памяти 59
	0x0000_0100	REG_60_TMR0	Регистр батарейной памяти 60 с функциями управления
	0x0000_0104	REG_61_TMR0	Регистр батарейной памяти 61 с функциями управления
	0x0000_0108	REG_62_TMR0	Регистр батарейной памяти 62 с функциями управления
	0x0000_010C	REG_63_TMR0	Регистр батарейной памяти 63 с функциями управления
	0x0000_0110	REG_60_TMR1	Регистр батарейной памяти 60 с функциями управления
	0x0000_0114	REG_61_TMR1	Регистр батарейной памяти 61 с функциями управления
	0x0000_0118	REG_62_TMR1	Регистр батарейной памяти 62 с функциями управления
	0x0000_011C	REG_63_TMR1	Регистр батарейной памяти 63 с функциями управления
	0x0000_0120	REG_60_TMR2	Регистр батарейной памяти 60 с функциями управления
	0x0000_0124	REG_61_TMR2	Регистр батарейной памяти 61 с функциями управления
	0x0000_0128	REG_62_TMR2	Регистр батарейной памяти 62 с функциями управления

Базовый адрес	Смещение	Название	Описание
	0x0000_012C	REG_63_TMR2	Регистр батарейной памяти 63 с функциями управления
Регистры часов реального времени			
	0x0000_0130	RTC_CNT_TMR0	Регистр основного счетчика часов реального времени
	0x0000_0134	RTC_DIV_TMR0	Регистр значения предварительного делителя часов реального времени
	0x0000_0138	RTC_PRL_TMR0	Основание счета предварительного делителя часов реального времени
	0x0000_013C	RTC_ALARM_TMR0	Регистр значения будильник часов реального времени
	0x0000_0140	RTC_CS_TMR0	Регистр управления и состояния часов реального времени
	0x0000_0150	RTC_CNT_TMR1	Регистр основного счетчика часов реального времени
	0x0000_0154	RTC_DIV_TMR1	Регистр значения предварительного делителя часов реального времени
	0x0000_0158	RTC_PRL_TMR1	Основание счета предварительного делителя часов реального времени
	0x0000_015C	RTC_ALARM_TMR1	Регистр значения будильник часов реального времени
	0x0000_0160	RTC_CS_TMR1	Регистр управления и состояния часов реального времени
	0x0000_0170	RTC_CNT_TMR2	Регистр основного счетчика часов реального времени
	0x0000_0174	RTC_DIV_TMR2	Регистр значения предварительного делителя часов реального времени
	0x0000_0178	RTC_PRL_TMR2	Основание счета предварительного делителя часов реального времени
	0x0000_017C	RTC_ALARM_TMR2	Регистр значения будильник часов реального времени
	0x0000_0180	RTC_CS_TMR2	Регистр управления и состояния часов реального времени

7.3.2.1 Регистр KEY

Base ADDR=	0x4000_1000	Offset=	0x0000_00F0													
REG Name:	KEY															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
KEY[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
KEY[15:0]																

Бит	Имя	Значение	Описание
-----	-----	----------	----------

31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока ВКР_CNTR
--------	-----------	-----------	--

7.3.2.2 REG_00...REG_59

Base ADDR=	0x4000_1000	Offset=	0x0000_0000 0x0000_0004 .. 0x0000_00EC												
------------	-------------	---------	---	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

DATA[15:0]															

Бит	Имя	Значение	Описание
31...0	DATA[31:0]		Данные памяти батарейного домена

7.3.2.3 REG_60_TMRx

Base ADDR=		0x4000_1000				Offset=		0x0000_0100							
								0x0000_0110							
								0x0000_0120							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR_IE	CLR_ERR	OVRSTn DSBL	PORSTn DSBL	ERR_63	ERR_LDO	ERR_60	LOCK STEP_EN	DUCCLIM_EN[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISABLE_ JTAG	S_SYSRST	S_OVRSTn	S_RESET1n	S_WDG_ RESET	S_FT_ RESET	S_RESET0n	S_UccRESET	FPOR	MODE[6:0]						

Бит	Имя	Значение	Описание
31	ERR_IE	0	Разрешение прерывания ВКР_IF по ошибкам ERR 0 – прерывание запрещено 1 – прерывание разрешено Сигнал формируется по правилам TMR
30	CLR_ERR	0	Бит сброса флагов ошибки Запись 0 – нет эффекта Запись 1 – сброс флагов ERR_63, ERR_LDO, ERR_60 Флаги не сбрасываются, если в момент их сброса событие сбоя продолжается. Сигнал сброса вырабатывается, если во всех трех регистрах TRMx записаны единицы.
29	OVRSTnDSBL	0	Флаг запрещения сброса по превышению питания OVRSTn 0 – схема сброса работает 1 – схема сброса заблокирована Схема POVR выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает.
28	PORSTnDSBL	0	Флаг запрещения сброса по превышению питания PORSTn 0 – схема сброса работает 1 – схема сброса заблокирована Схема POR выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает.
27	ERR_63	0	Сбой конфигурации регистров REG_63 0 – нет сбоя 1 – есть сбой
26	ERR_LDO	0	Сбой конфигурации регистров REG_61 и REG_62 0 – нет сбоя 1 – есть сбой
25	ERR_60	0	Сбой конфигурации регистров REG_60 0 – нет сбоя 1 – есть сбой
24	-	1	Зарезервировано Взводится в 1 при любом сбросе

Бит	Имя	Значение	Описание
23...16	DUCCLIM_EN[7:0]	11111111	Флаг разрешения работы ограничения тока в доменах DUCC DUCCLIM_EN[0] – LDO0 основного питания DUcc0,1,2 DUCCLIM_EN[1] – LDO1 основного питания DUcc0,1,2 DUCCLIM_EN[2] – LDO2 основного питания DUcc0,1,2 DUCCLIM_EN[3] – LDO3 основного питания DUcc0,1,2 0 – нет ограничения по току 1 – есть ограничения по току (~300 мА на один LDO) DUCCLIM_EN[4] – LDO батарейного домена DUcc_B DUCCLIM_EN[5] – LDO умножителей частоты DUcc_PLL 0 – нет ограничения по току 1 – есть ограничения по току (~50 мА на один LDO)
15	DISABLE_JTAG	0	Запрещение отладочного интерфейса JTAG 0 – JTAG_A и/или JTAG_B разрешены 1 – JTAG_A и/или JTAG_B запрещены
14	S_SYSRSTn	x	Флаг события сброса с вывода SYSRSTn 0 – не было сброса по SYSRSTn 1 – был сброс по SYSRSTn Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
13	S_OVRSTn	0	Флаг события сброса с вывода OVRSTn 0 – не было сброса по OVRSTn 1 – был сброс по OVRSTn Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
12	S_RESET1n	x	Флаг события сброса с вывода RSTn1 0 – не было сброса по RSTn1 1 – был сброс по RSTn1 Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
11	S_WDG_RESET	0	Флаг события сброса с вывода WDG_RESET 0 – не было сброса по WDG_RESET 1 – был сброс по WDG_RESET Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
10	S_FT_RESET	0	Флаг события сброса с вывода FT_RESET 0 – не было сброса по FT_RESET 1 – был сброс по FT_RESET Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
9	S_RESET0n	x	Флаг события сброса с вывода RSTn0 0 – не было сброса по RSTn0 1 – был сброс по RSTn0 Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
8	S_UccRESET	1	Флаг события сброса UccRESET 0 – не было сброса по UccRESET 1 – был сброс по UccRESET Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
7	FPOR	0	Флаг успешного завершения работы загрузочной бутовой программы 0 – бутовая программа не завершена 1 – бутовая программа завершена Возможна только запись 1. Сброс по любому сигналу сброса.
6...0	MODE[6:0]	xxxxxx	Режим запуска микроконтроллера с выводов {MODE[7:5], MODE[3:0]} с исключенным битом четности. См. подраздел «Режимы работы микроконтроллера»

7.3.2.4 REG_61_TMRx

Base ADDR=		0x4000_1000		Offset=		0x0000_0104 0x0000_0114 0x0000_0124									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DUCC3_RDY	EN_DUCC3	TRIM3[2:0]			SRILOW3[2:0]			DUCC2_RDY	EN_DUCC0	TRIM2[2:0]			SRILOW2[2:0]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUCC1_RDY	EN_DUCC1	TRIM1[2:0]			SRILOW1[2:0]			DUCC0_RDY	EN_DUCC0	TRIM0[2:0]			SRILOW0[2:0]		

Бит	Имя	Значение	Описание
31	DUCC3_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO3 регулятора
30	EN_DUCC3		
29:27	TRIM3[2:0]		
26:24	SRILOW3[2:0]		
23	DUCC2_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO2 регулятора
22	EN_DUCC2		
21:19	TRIM2[2:0]		
18:16	SRILOW2[2:0]		
15	DUCC1_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO1 регулятора
14	EN_DUCC1		
13:11	TRIM1[2:0]		
10:8	SRILOW1[2:0]		
7	DUCC0_RDY		Флаг готовности напряжения LDO0 регулятора 0 – регулятор LDO0 не готов 1 – регулятор LDO0 в норме
6	EN_DUCC0		Разрешение выдачи напряжения LDO0 регулятором 0 – разрешена работа LDO0 регулятора 1 – запрещена работа LDO0 регулятора Бит предустанавливается сигналами WAKEUP, ALRF, UCCRESET Схема LDO выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает.

Бит	Имя	Значение	Описание
5:3	TRIM0[2:0]		<p>Подстройка выходного напряжения DUсс регулятора LDO0</p> <p>3'b0 000 = +0,01В (типичное значение)</p> <p>3'b0 001 = +0,05В</p> <p>3'b0 010 = +0,10В</p> <p>3'b0 011 = +0,15В</p> <p>3'b0 100 = - 0,01В</p> <p>3'b0 101 = - 0,05В</p> <p>3'b0 110 = - 0,10В</p> <p>3'b0 111 = - 0,15В</p> <p>Сигнал формируется по правилам TMR</p>
2:0	SRILOW0[2:0]		<p>Настройка режима работы регулятора LDO0</p> <p>3'b0 011 – Частота до 100 МГц (доп. ток ~650 мкА)</p> <p>3'b0 010 – Частота до 50 МГц (доп. ток ~550 мкА)</p> <p>3'b0 001 – Частота до 30 МГц (доп. ток ~450мкА)</p> <p>3'b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типичное значение)</p> <p>3'b0 100 – Частота до 1МГц (доп. ток ~300мкА)</p> <p>3'b0 101 – Частота до 500 кГц (доп. ток ~200 мкА)</p> <p>3'b0 110 – Частота до 100 кГц (доп. ток ~100мкА)</p> <p>3'b0 111 – Частота более 10 кГц (доп. ток ~50 мкА)</p> <p>Сигнал формируется по правилам TMR</p>

7.3.2.5 REG_62_TMRx

Base ADDR=		0x4000_1000				Offset=		0x0000_0108 0x0000_0118 0x0000_0128							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TRIM_REF[2:0]			-	-	-	TRIMR[6]		TRIMR[7]	TRIMR[5:3]			TRIMR [2:0]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUCC5_RDY	EN_DUCC5	TRIM5[2:0]			SRILOW5[2:0]			DUCC4_RDY	EN_DUCC4	TRIM4[2:0]			SRILOW4[2:0]		

Бит	Имя	Значение	Описание
31	-		TRIM_REF[2:0] – подстройка источника опорного напряжения LDO 100 – типовое значение Рекомендуется не изменять типовое значение
30:28	TRIM_REF[2:0]		
27...25	-		TRIMR[7:0] – подстройка источника опорного тока LDO 10000000 – типовое значение Рекомендуется не изменять типовое значение
24	TRIMR[6]		
23			
22	TRIMR[7]		
21:19	TRIMR[5:3]		
18:16	TRIMR[2:0]		
15	DUCC5_RDY		Флаг готовности напряжения для домена DUcc_PLL 0 – нет напряжения DUCC в домене DUcc_PLL 1 – есть напряжение DUCC в домене DUcc_PLL
14	EN_DUCC5		Разрешение выдачи напряжения домен DUcc_PLL 0 – не выдавать напряжение в домен DUcc_PLL 1 – выдавать напряжение в домен DUcc_PLL Бит предустанавливается сигналами WAKEUP, ALRF, UCCRESET Схема LDO выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает.
13:11	TRIM5[2:0]		Подстройка напряжение DUcc в домене DUcc_PLL 3'b0 000 = +0,01В (типовое значение) 3'b0 001 = +0,05В 3'b0 010 = +0,10В 3'b0 011 = +0,15В 3'b0 100 = - 0,01В 3'b0 101 = - 0,05В 3'b0 110 = - 0,10В 3'b0 111 = - 0,15В Сигнал формируется по правилам TMR

Бит	Имя	Значение	Описание
10:8	SRILOW5[2:0]		<p>Настройка режима работы домена DUcc_PLL:</p> <p>3'b0 011 – Частота до 100 МГц (доп. ток ~650 мкА) 3'b0 010 – Частота до 50 МГц (доп. ток ~550 мкА) 3'b0 001 – Частота до 30 МГц (доп. ток ~450мкА)</p> <p>3'b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типичное значение)</p> <p>3'b0 100 – Частота до 1МГц (доп. ток ~300мкА) 3'b0 101 – Частота до 500 кГц (доп. ток ~200 мкА) 3'b0 110 – Частота до 100 кГц (доп. ток ~100мкА) 3'b0 111 – Частота более 10 кГц (доп. ток ~50 мкА)</p> <p>Сигнал формируется по правилам TMR</p>
7	DUCC4_RDY		<p>Флаг готовности напряжения для домена DUccB</p> <p>0 – нет напряжения DUCC в домене DUccB 1 – есть напряжение DUCC в домене DUccB</p>
6	EN_DUCC4		<p>Не имеет значения, батарейный домен DUccB всегда включен. Выключен может быть только сигналом SHDN. Значение данного бита должно совпадать во всех регистрах.</p>
5:3	TRIM4[2:0]		<p>Подстройка напряжение DUcc в домене DUccB</p> <p>3'b0 000 = +0,01В (типичное значение) 3'b0 001 = +0,05В 3'b0 010 = +0,10В 3'b0 011 = +0,15В</p> <p>3'b0 100 = - 0,01В 3'b0 101 = - 0,05В 3'b0 110 = - 0,10В 3'b0 111 = - 0,15В</p> <p>Сигнал формируется по правилам TMR</p>
2:0	SRILOW4[2:0]		<p>Настройка режима работы домена DUccB:</p> <p>3'b0 011 – Частота до 100 МГц (доп. ток ~650 мкА) 3'b0 010 – Частота до 50 МГц (доп. ток ~550 мкА) 3'b0 001 – Частота до 30 МГц (доп. ток ~450мкА)</p> <p>3'b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типичное значение)</p> <p>3'b0 100 – Частота до 1МГц (доп. ток ~300мкА) 3'b0 101 – Частота до 500 кГц (доп. ток ~200 мкА) 3'b0 110 – Частота до 100 кГц (доп. ток ~100мкА) 3'b0 111 – Частота более 10 кГц (доп. ток ~50 мкА)</p> <p>Сигнал формируется по правилам TMR</p>

7.3.2.6 REG_63_TMRx

Base ADDR=	0x4000_1000	Offset=	0x0000_010C 0x0000_011C 0x0000_012C													
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RTC_RST	STANDBY	HSI_TRIM[5:0]						HSI_RDY	HSI_ON	LSI_RDY	LSI_TRIM[4:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSI_ON	.	LSE_RDY	CAL[7:0]								RTC_EN	RTC_SEL[1:0]		LSE_BYP	LSE_ON

Бит	Имя	Значение	Описание
31	RTC_RST		Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются Сигнал формируется по правилам TMR
30	STANDBY		Режим отключения всех регуляторов DUCC: 0 – регуляторы включены и выдают напряжение; 1 – регулятор выключен Триггер сбрасывается флагом ALRF или по низкому уровню на выводе WAKEUP и сигналом сброса BDUCC Схемы LDO выключаются, если во всех трех каналах они выключены, если хотя бы в одном канале они разрешены, то работают.
29...24	HSI_TRIM[5:0]	32	Коэффициент подстройки частоты генератора HSI. 0 – минимальная частота ... 32 – типовое значение частоты ... 63 – максимальна частота Сигнал формируется по правилам TMR
23	HSI_RDY		Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
22	HSI_ON		Бит управления генератором HSI: 0 – генератор выключен; 1 – генератор включен Схемы HSI выключаются, если во всех трех каналах она выключена, если хотя бы в одном канале она разрешена, то работает.
21	LSI_RDY		Флаг выхода генератора LSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор находится в рабочем режиме
20...16	LSI_TRIM[4:0]	32	Коэффициент подстройки частоты генератора LSI. 0 – максимальная частота ... 8 – типовое значение частоты ... 15 и более – минимальная частота Сигнал формируется по правилам TMR
15	LSI_ON		Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен Схемы LSI выключаются, если во всех трех каналах она выключена, если хотя бы в одном канале она разрешена, то работает.
14	FILTER_EN		Фильтрация входных сигналов генератора LSE 0 – фильтр выключен

Бит	Имя	Значение	Описание
			1 – фильтр включен
13	LSE_RDY		Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
12...5	CAL[7:0]		Коэффициент подстройки тактовой частоты часов реального времени, из каждых 2 ²⁰ тактов будет замаскировано CAL тактов: 00000000 – 0 тактов 00000001 – 1 такт 11111111 – 255 тактов Таким образом, при частоте 32768.00000 Гц при CAL = 0 тактовая частота = 32768.00000 Гц при CAL = 1 тактовая частота = 32767,96875 Гц; ... при CAL = 255 тактовая частота = 32760.03125 Гц Сигнал формируется по правилам TMR
4	RTC_EN		Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена Сигнал формируется по правилам TMR
3...2	RTC_SEL[1:0]		Биты выбора источника тактовой синхронизации часов реального времени: 00 – LSI 01 – LSE 10 – RTCCLK (формируется в блоке CLKCNTN) 11 - зарезервировано Сигнал формируется по правилам TMR
1	LSE_BYP		Бит управления генератором LSE: 0 – режим работы с внешним резонатором 1 – режим работы с внешним генератором Сигнал формируется по правилам TMR
0	LSE_ON		Бит разрешения работы LSE генератора 0 – генератор выключен 1 – генератор включен Сигнал формируется по правилам TMR

7.3.2.7 RTC_CNT_TMRx

Base ADDR=		0x4000_1000					Offset=		0x0000_0130							
									0x0000_0150							
									0x0000_0170							
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RTC_CNT[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_CNT[15:0]															

Бит	Имя	Значение	Описание
31...0	RTC_CNT[31:0]		Значение основного счетчика часов реального времени

7.3.2.8 RTC_DIV_TMRx

Base ADDR=		0x4000_1000					Offset=		0x0000_0134							
									0x0000_0154							
									0x0000_0174							
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-												RTC_DIV[19:16]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_DIV[15:0]															

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...0	RTC_DIV[19:0]		Значение счетчика предварительного делителя часов реального времени

7.3.2.9 RTC_PRL_TMRx

Base ADDR=	0x4000_1000	Offset=	0x0000_0138 0x0000_0158 0x0000_0178												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												RTC_PRL[19:16]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_PRL[15:0]															

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...0	RTC_PRL[19:0]		Значение основания для счетчика предварительного делителя часов реального времени

7.3.2.10 RTC_ALRM_TMRx

Base ADDR=	0x4000_1000	Offset=	0x0000_013C 0x0000_015C 0x0000_017C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC_ALRM[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_ALRM[15:0]															

Бит	Имя	Значение	Описание
31...0	RTC_ALRM[31:0]		Значение, при превышении которого основным счетчиком часов реального времени будет выработан сигнал ALRF

7.3.2.11 RTC_CS_TMRx

Base ADDR=	0x4000_1000	Offset=	0x0000_0140 0x0000_0160 0x0000_0180												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERR_CMx	ERR_DIVx	ERR_CNTx	ERR_CC	ERR_CM	ERR_DIV	ERR_CNT	CLR_ERR	WEC	ALRF_IE	SECF_IE	OWF_IE	ALRF	SECF	OWF

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
15	ERR_IE		Флаг разрешения прерывания RTC_IF по событиям сбоя 0 – Прерывание запрещено 1 – Прерывание разрешено
14	ERR_CMx		Сбой в данном канале тактовой частоты RTC 0 – нет сбоя 1 – есть сбой
13	ERR_RTCx		Сбой в данном канале DIV часов RTC 0 – нет сбоя 1 – есть сбой
12	ERR_CNTx		Сбой в данном канале CNT часов RTC 0 – нет сбоя 1 – есть сбой
11	ERR_CC		Отказ источника тактовой частоты RTC 0 – нет сбоя 1 – есть сбой При отказе источника тактовой частоты, RTC автоматически переключается на LSI генератор.
10	ERR_CM		Сбой в одном из каналов тактовой частоты RTC превышения максимальной частоты 0 – нет сбоя 1 – есть сбой
9	ERR_DIV		Сбой в одном из каналов DIV часов RTC 0 – нет сбоя 1 – есть сбой
8	ERR_CNT		Сбой в одном из каналов CNT часов RTC И конфигурации RTC 0 – нет сбоя 1 – есть сбой
7	CLR_ERRx		Бит сброса флагов ошибки в данном канале 0 – нет сброса. 1 – сброс флагов ERR_RTCx, ERR_CNTx, ERR_CMx и сброса ошибок конфигурации RTC (кроме PRL и ALRM). Флаги не сбрасываются, если в момент их сброса событие сбоя продолжается. Для сброса флага ERR_CC необходима запись единиц во все CLR_ERRx.

Бит	Имя	Значение	Описание
6	WEC		Запись завершена: 0 – можно записывать в регистры часов реального времени 1 – запись в регистры запрещена
5	ALRF_IE		Флаг разрешения прерывания RTC_IF по событию ALRF 0 – не генерировать прерывание по событию ALRF 1 – генерировать прерывание по событию ALRF
4	SECF_IE		Флаг разрешения прерывания RTC_IF по событию SECF 0 – не генерировать прерывание по событию 1 – генерировать прерывание по событию
3	OWF_IE		Флаг разрешения прерывания RTC_IF по событию OWF 0 – не генерировать прерывание по событию 1 – генерировать прерывание по событию
2	ALRF		Флаг возникновения события превышения основным счетчиком RTC_CNT значения в регистре RTC_ALARM 0 – нет события 1 – есть событие Сбрасывается записью 1.
1	SECF		Флаг возникновения события равенства предварительного счетчика RTC_DIV и значения в регистре RTC_PRL 0 – нет события 1 – есть событие Сбрасывается записью 1.
0	OWF		Флаг возникновения события переполнения основного счетчика RTC_CNT 0 – нет события 1 – есть событие Сбрасывается записью 1.

7.4 Контроллер детектора напряжения питания (PWRCNTR)

Блок детектора напряжений питания позволяет оценить напряжение внешних источников питания U_{сс} (питание IO, приемопередатчиков и т.п.), U_{сс1} (питание LDO) и BU_{сс} (питание батарейного домена), а также определить ток потребления доменов питания DU_{сс}. Определение значения источников напряжения осуществляется путем их сравнения с источником опорного напряжения в блоке PVD и самих LDO. При этом обеспечиваемая точность измерения значения напряжения источников питания составляет ±100 мВ. Точность измерения токов потребления составляет ±5 мА для LDO0-LDO3 и ±0,5 мА для LDO батарейного домена и блока умножителей частоты PLL.

В блоке PWR_CNTR могут быть настроены условия для выработки событий превышения или снижения ниже других значений, при возникновении которых формируются прерывания, либо события для блока управления сбоями, ошибками и отказами (FT_CNTR), приводящими к сбросу микроконтроллера.

7.4.1 Описание регистров

Таблица 26 – Описание регистров блока детектора напряжений питания

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_2000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	CNTR0		Регистр управления PWR_CNTR0
	0x0000_0008	CNTR1		Регистр управления PWR_CNTR1
	0x0000_000C	CNTR2		Регистр управления PWR_CNTR2
	0x0000_0010	STAT		Регистр статуса PWR_CNTR
	0x0000_0014	ULIMIT		Регистр настройки ограничения напряжений PWR_CNTR
	0x0000_0018	CLIMIT		Регистр настройки ограничения токов PWR_CNTR

7.4.1.1 KEY

Base ADDR=	0x4000_2000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока PWR_CNTR

7.4.1.2 CNTR0

Base ADDR=	0x4000_2000					Offset=	0x0000_0004									
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PDID_EN[7:0]	PBVD_EN	PVD1_EN	PVD_EN	

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...3	PDID_EN[7:0]		Бит разрешения контроля тока потребления через LDO регуляторы PDID_EN[0] - регулятор LDO0 PDID_EN[1] - регулятор LDO1 PDID_EN[2] - регулятор LDO2 PDID_EN[3] - регулятор LDO3 PDID_EN[4] - регулятор LDO батарейного домена PDID_EN[5] - регулятор LDO умножителей частоты Остальные не используются 0 – отключен контроль 1 – включен контроль
2	PBVD_EN		Бит разрешения контроля уровня напряжения ВUсс 0 – отключен 1 – включен Внимание: При PBVD_EN = 1 идет разрядка батарейного источника питания, независимо от наличия основного питания.
1	PVD1_EN		Бит разрешения контроля уровня напряжения Uсс1 0 – отключен 1 – включен
0	PVD_EN		Бит разрешения контроля уровня напряжения Uсс 0 – отключен 1 – включен

7.4.1.3 CNTR1

Base ADDR=		0x4000_2000				Offset=		0x0000_0008									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDID_IE[7:0]													PBVD_IE	PVD1_IE	PVD_IE

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...3	PDID_IE[7:0]		Бит разрешения прерывания по событию тока потребления LDO регуляторов PDID_IE[0] - регулятор LDO0 PDID_IE[1] - регулятор LDO1 PDID_IE[2] - регулятор LDO2 PDID_IE[3] - регулятор LDO3 PDID_IE[4] - регулятор LDO батарейного домена PDID_IE[5] - регулятор LDO умножителей частоты Остальные не используются 0 – запрещено прерывание 1 – разрешено прерывание
2	PBVDP_IE		Бит разрешения прерывания по событию уровня напряжения ВUcc 0 – запрещено прерывание 1 – разрешено прерывание
1	PVD1P_IE		Бит разрешения прерывания по событию уровня напряжения Ucc1 0 – запрещено прерывание 1 – разрешено прерывание
0	PVDP_IE		Бит разрешения прерывания по событию уровня напряжения Ucc 0 – запрещено прерывание 1 – разрешено прерывание

7.4.1.4 CNTR2

Base ADDR=	0x4000_2000					Offset=	0x0000_000C									
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIDN[7:0]											PBVDN	PVD1N	PVDN		

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...3	PDIDN[7:0]		Бит инвертирования сигнала от датчика тока LDO для формирования события PDIDN[0] - регулятор LDO0 PDIDN[1] - регулятор LDO1 PDIDN[2] - регулятор LDO2 PDIDN[3] - регулятор LDO3 PDIDN[4] - регулятор LDO батарейного домена PDIDN[5] - регулятор LDO умножителей частоты Остальные не используются 0 – не инвертировать (флаг по превышению тока) 1 – инвертировать (флаг по снижению тока)
2	PBVDN		Бит инвертирования сигнала от датчика уровня напряжения ВUсс для формирования события 0 – не инвертировать (флаг по превышению) 1 – инвертировать (флаг по снижению)
1	PVD1N		Бит инвертирования сигнала от датчика уровня напряжения Uсс1 для формирования события 0 – не инвертировать (флаг по превышению) 1 – инвертировать (флаг по снижению)
0	PVDN		Бит инвертирования сигнала от датчика уровня напряжения Uсс для формирования события 0 – не инвертировать (флаг по превышению) 1 – инвертировать (флаг по снижению)

7.4.1.5 STAT

Base ADDR=		0x4000_2000					Offset=		0x0000_0010						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDID[7:0]													PBVDP	PVDP1	PVDP

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...8	PDID[7:0]		Флаги события по току LDO PDID[0] - регулятор LDO0 PDID[1] - регулятор LDO1 PDID[2] - регулятор LDO2 PDID[3] - регулятор LDO3 PDID[4] - регулятор LDO батарейного домена PDID[5] - регулятор LDO умножителей частоты Остальные не используются 0 – нет события 1 – есть событие, ток больше допустимого CURDUCC при PDIDN=0, или ток ниже допустимого CURDUCC при PDIDN=1 Сбрасывается записью 1.
7...3	-		Зарезервировано
2	PBVDP		Флаг события по уровню напряжения ВUcc 0 – нет события 1 – есть событие, напряжение больше допустимого BUCCMAX при PVBDN=0, или напряжение ниже допустимого BUCCMAX при PVBDN=1 Сбрасывается записью 1.
1	PVDP1		Флаг события по уровню напряжения Ucc1 0 – нет события 1 – есть событие, напряжение больше допустимого UCC1MAX при PVD1N=0, или напряжение ниже допустимого UCC1MAX при PVD1N=1 Сбрасывается записью 1.
0	PVDP		Флаг события по уровню напряжения Ucc 0 – нет события 1 – есть событие, напряжение больше допустимого UCCMAX при PVDN=0, или напряжение ниже допустимого UCCMAX при PVDN=1 Сбрасывается записью 1.

7.4.1.6 ULIMIT

Base ADDR=		0x4000_2000				Offset=		0x0000_0014								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-											BUCCMAX[4:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			UCC1MAX[4:0]				-			UCCMAX[4:0]					

Бит	Имя	Значение	Описание
31...21	-		Зарезервировано
20...16	BUCCMAX[4:0]		Значение максимально допустимого напряжения BUCC, при превышении которого происходит выработка флага PBVDP и EVEN_PVBD 5'b 00000 – более 1,2В (ref) 5'b 00001 – более 1,4В 5'b 00010 – более 1,6В 5'b 00011 – более 1,8В ... 5'b 11111 – более 7,4В
15...13	-		Зарезервировано
12...8	UCC1MAX[4:0]		Значение максимально допустимого напряжения UCC1, при превышении или снижении которого происходит выработка флага PVDP1 и EVENT_PVD1 5'b 00000 – более 1,2В (ref) 5'b 00001 – более 1,4В 5'b 00010 – более 1,6В 5'b 00011 – более 1,8В ... 5'b 11111 – более 7,4В
7...5	-		Зарезервировано
4...0	UCCMAX[4:0]		Значение максимально допустимого напряжения UCC, при превышении которого происходит выработка флага PVDP и EVENT_PVD 5'b 00000 – более 1,2В (ref) 5'b 00001 – более 1,4В 5'b 00010 – более 1,6В 5'b 00011 – более 1,8В ... 5'b 11111 – более 7,4В

7.4.1.7 CLIMIT

Base ADDR=		0x4000_2000				Offset=		0x0000_0018											
REG Name:																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
								CURDUCC7[2:0]				CURDUCC6[2:0]				CURDUCC5[2:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
CURDUCC5[0]				CURDUCC4[2:0]			CURDUCC3[2:0]			CURDUCC2[2:0]			CURDUCC1[2:0]			CURDUCC0[2:0]		

Бит	Имя	Значение	Описание
31...18	-		Зарезервировано
17...15	CURDUCC5[2:0]		Уровень тока для выработки события по току DUCC0 и формирования флага PDID[5] и события EVENT_CUR[5] Аналогично CURDUCC4[2:0]
14...12	CURDUCC4[2:0]		Уровень тока для выработки события по току LDO батарейного домена и формирования флага PDID[4] и события EVENT_CUR[4] 3'b 000 – ток более 0,5 мА 3'b 001 – ток более 1,5 мА 3'b 010 – ток более 2,5 мА 3'b 011 – ток более 3,5 мА 3'b 100 – ток более 4,5 мА 3'b 101 – ток более 5,5 мА 3'b 110 – ток более 6,5 мА 3'b 111 – ток более 7,5 мА
11...9	CURDUCC3[2:0]		Уровень тока для выработки события по току LDO3 и формирования флага PDID[3] и события EVENT_CUR[3] Аналогично CURDUCC0[2:0]
8...6	CURDUCC2[2:0]		Уровень тока для выработки события по току LDO2 и формирования флага PDID[2] и события EVENT_CUR[2] Аналогично CURDUCC0[2:0]
5...3	CURDUCC1[2:0]		Уровень тока для выработки события по току LDO1 и формирования флага PDID[1] и события EVENT_CUR[1] Аналогично CURDUCC0[2:0]
2...0	CURDUCC0[2:0]		Уровень тока для выработки события по току LDO0 и формирования флага PDID[0] и события EVENT_CUR[0] 3'b 000 – ток более 5 мА 3'b 001 – ток более 15 мА 3'b 010 – ток более 25 мА 3'b 011 – ток более 35 мА 3'b 100 – ток более 45 мА 3'b 101 – ток более 55 мА 3'b 110 – ток более 65 мА 3'b 111 – ток более 75 мА

7.5 Контроллер обработки событий отказов, сбоев и ошибок (FT_CNTR)

Все события, критические для функционирования и безопасности выполнения пользовательской программы, обрабатываются в контроллере обработки сбоев FT_CNTR. В зависимости от настроек блока FT_CNTR возникновение события может вызвать сигнал прерывания или сигналы предупреждения FT_WRNn или сигнал запроса сброса FT_RSTn микроконтроллера.

Для каждого события в блоке FT_CNTR возможна настройка реакции:

- игнорирование;
- прерывание (для группы с высокой важностью FT_IF0, средней FT_IF1 и низкой FT_IF2);
- формирование отложенного сброса микроконтроллера FT_WRNn и FT_RSTn.

При возникновении события, вызывающего запрос сброса микроконтроллера, в контроллере FT_CNTR автоматически запускается таймер. Период времени таймера задается программно и формирует временной интервал для выполнения действий по парированию возникшей ошибки. В момент возникновения ошибки формируется сигнал FT_WRNn, а по завершению счета таймера формируется сигнал FT_RSTn. Таким образом период таймера дает время, за которое можно успеть исправить программными средствами ситуацию, вызвавшую сбой. Если к моменту окончания счета таймера сбойная ситуация разрешилась, то сброс таймера приведет к снятию сигнала FT_WRNn.

Сигналы FT_WRNn и FT_RSTn являются выходами с открытым стоком, и имеют встроенную подтяжку к питанию с номиналом 50 кОм.

7.5.1 Описание регистров

Таблица 27 – Описание регистров контроллера FT_CNTR

Базовый адрес	Название	Описание
0x4000_3000	FT_CNTR	Блок обработки сбоев, ошибок и отказов
Смещение		
0x0000_0000	KEY	Регистр ключа
0x0000_0004	CONTROL	Регистр управления блоком
0x0000_0008	STATUS	Регистр статуса блока
0x0000_000C	TIMEOUT	Регистр времени отложенного сброса
0x0000_0010	TICKCNT	Общий счетчик HSI генератора
0x0000_0014	FIRSTEVENT	Значение TICKCOUNT первого зафиксированного события
0x0000_0018	LASTEVENT	Значения TICKCOUNT последнего зафиксированного события
0x0000_001C	TIMEOUTCNT	Текущее значение счетчика отложенного сброса
0x0000_0020	EVENT0	Флаги событий сбоев
0x0000_0024	EVENT1	Флаги событий сбоев
0x0000_0028	EVENT2	Флаги событий сбоев
0x0000_002C	EVENT3	Флаги событий сбоев
0x0000_0030	EVENT4	Флаги событий сбоев
0x0000_0034	EVENT5	Флаги событий сбоев
0x0000_0038	EVENT6	Флаги событий сбоев
0x0000_003C	EVENT7	Флаги событий сбоев
0x0000_0040	EVENT8	Флаги событий сбоев
0x0000_0044	EVENT9	Флаги событий сбоев
0x0000_0048	EVENT10	Флаги событий сбоев
0x0000_004C	EVENT11	Флаги событий сбоев
0x0000_0050	EVENT12	Флаги событий сбоев
0x0000_0054	RESET_EVENT0	Биты разрешения отложенного сброса от событий сбоев
0x0000_0058	RESET_EVENT1	Биты разрешения отложенного сброса от событий сбоев
0x0000_005C	RESET_EVENT2	
0x0000_0060	RESET_EVENT3	Биты разрешения отложенного сброса от событий сбоев
0x0000_0064	RESET_EVENT4	Биты разрешения отложенного сброса от событий сбоев

0x0000_0068	IE_EVENT5	Биты разрешения прерывания от событий сбоев
0x0000_006C	IE_EVENT6	Биты разрешения прерывания от событий сбоев
0x0000_0070	IE_EVENT7	Биты разрешения прерывания от событий сбоев
0x0000_0074	IE_EVENT8	Биты разрешения прерывания от событий сбоев
0x0000_0078	IE_EVENT9	Биты разрешения прерывания от событий сбоев
0x0000_007C	IE_EVENT10	Биты разрешения прерывания от событий сбоев
0x0000_0080	IE_EVENT11	Биты разрешения прерывания от событий сбоев
0x0000_0084	IE_EVENT12	Биты разрешения прерывания от событий сбоев

7.5.1.1 KEY

Base ADDR=	0x4000_3000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока FT_CNTR

7.5.1.2 CONTROL

Base ADDR=	0x4000_3000	Offset=	0x0000_0004												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													RESET TIMEOUT CNT	TICK OVER_IE	

Бит	Имя	Значение	Описание
31...3	-		Зарезервировано
2	RESET_TIMEOUTCNT	0	Бит сброса регистра TIMEOUTCNT в состояние ожидания Запись 0 – ничего Запись 1 – сброс, при условии отсутствия каких-либо флагов события, для которых разрешен вызов отложенного сброса
1	TICKOVER_IE	0	Бит разрешения формирования сигнала предупреждения FT_WRNn при возникновении переполнения счетчика TICKCNT. Счетчик TICKCNT переполняется раз в ~9 минут
0	-	1	Зарезервировано

7.5.1.3 STATUS

Base ADDR=		0x4000_3000				Offset=		0x0000_0008							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									FT_IF3	FT_IF2	FT_IF1	FT_IF0	LAST EVENT	FIRST EVENT	TICKOVER

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7			
6	FT_IF3	0	Флаг запроса прерывания от возникновения разрешенных событий с низкой важностью (EVENT9-EVENT12) 0 – нет запроса прерывания 1 – есть запрос прерывания
5	FT_IF2	0	Флаг запроса прерывания от возникновения разрешенных событий со средней важностью (EVENT5-EVENT8) 0 – нет запроса прерывания 1 – есть запрос прерывания
4	FT_IF1	0	Флаг запроса прерывания от возникновения любых событий с высокой важностью (EVENT0-EVENT4) 0 – нет запроса прерывания 1 – есть запрос прерывания
3	FT_IF0	0	Флаг запроса прерывания от возникновения событий с высокой важностью, для которых разрешен отложенный сброс (EVENT0-EVENT4) 0 – нет запроса прерывания при запуске отложенного сброса 1 – есть запрос прерывания при запуске отложенного сброса
2	LAST_EVENT	0	Флаг возникновения последнего события сбоя 0 – нет события или был сброшен 1 – есть событие Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	FIRST_EVENT	0	Флаг возникновения первого события сбоя 0 – нет события 1 – есть событие Данный флаг сбрасывается только сигналом сброса
0	TICKOVER	0	Флаг переполнения счетчика TICKCNT 0 – нет переполнения 1 – было переполнение Флаг сбрасывается записью 1, если в момент сброса происходит очередное переполнение, флаг не сбрасывается.

7.5.1.4 TIMEOUT

Base ADDR=		0x4000_3000				Offset=		0x0000_000C							
REG Name:		TIMEOUT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	TIMEOUT[15:0]	16'HFFFF	Число тактов генератора HSI от момента возникновения сбоя до генерации сигнала сброса. При возникновении события сброса данное значение перезаписывается в регистр TIMEOUTCNT, который после этого начинает отсчет данного времени

7.5.1.5 TICKCNT

BaseADDR=		0x4000_3000				Offset=		0x0000_0010							
REGName:		TICKCNT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TICKCNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TICKCNT[15:0]															

Бит	Имя	Значение	Описание
31...0	TICKCNT[31:0]		Счетчик тактов HSI генератора с момента последнего сброса. При переполнении данного счетчика взводится бит TICKOVER.

7.5.1.6 FIRSTEVENT

BaseADDR=		0x4000_3000				Offset=		0x0000_0014								
REGName:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FIRSTEVENT[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRSTEVENT[15:0]															

Бит	Имя	Значение	Описание
31...0	FIRSTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения первого события сбоя. Сбрасывается только сигналом сброса

7.5.1.7 LASTEVENT

Base ADDR=		0x4000_3000				Offset=		0x0000_0018								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LASTEVENT[31:0]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LASTEVENT[15:0]															

Бит	Имя	Значение	Описание
31...0	LASTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения последнего события сбоя. Сбрасывается только сигналом сброса

7.5.1.8 TIMEOUTCNT

Base ADDR=		0x4000_3000				Offset=		0x0000_001C									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
RESERVED															TIMEOUTCNT[16]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUTCNT[15:0]															

Бит	Имя	Значение	Описание
31...17	-		Зарезервировано
16...0	TIMEOUTCNT[16:0]	17`H1FFFF	Значение счетчика времени отложенного сброса.

В момент возникновения события, для которого разрешено формирование отложенного сброса, в счетчик TIMEOUTCNT переписывается значение регистра TIMEOUT. Счетчик начинает считать вниз до нуля и при достижении нулевого значения формируется сигнал сброса FT_RSTn.

Счетчик TIMEOUTCNT имеет 17 разрядов, при сбросе устанавливается в состояние всех единиц. При возникновении события устанавливается в состояние 16-ти битного регистра TIMEOUT, при этом 17-й разряд сбрасывается в 0. Это условие начала счета для данного счетчика.

Счетчик выполнен по троированной логике.

Для прекращения счета отложенного сброса необходимо:

- снять все флаги событий, разрешающих формирование сигнала запроса сброса FT_RSTn;
- через запись 1 в бит RESET_TIMEOUTCNT установить счетчик в исходное состояние.

Если в момент сброса счетчика TIMEOUTCNT сохраняются условия для его счета, то сброс счетчика не произойдет.

7.5.1.9 EVENT0 (HIGH)

BaseAD	0x4000_3000					Offset=	0x0000_0020										
DR=																	
REGName:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
		EXTBUS_COMP_REG_ERR	AHB_EXTBUS_COMP_ERR			RAMD_COMP_REG_ERR	AHB_RAMD_COMP_ERR	OTP_COMP_REG_ERR	AHB_OTP_COMP_ERR	AHB_C_COMP_ERR	AHB_M_COMP_ERR	AHB_S_COMP_ERR	AHB_D_COMP_ERR	AHB_I_COMP_ERR	COMP_REG_ERR		

Бит	Имя	Значение	Описание
31...14	-		Зарезервировано
13	EXTBUS_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера EXTBUS 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
12	AHB_EXTBUS_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера EXTBUS 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
11..10	-	2'b00	Зарезервировано
9	RAMD_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера RAMD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
8	AHB_RAMD_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера RAMD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
7	OTP_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера OTP 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
6	AHB_OTP_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера OTP 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
5	-		Зарезервировано
4	AHB_M_COMP_ERR	0	Флаг ошибки сравнения сигналов системной шины DMA процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	AHB_S_COMP_ERR	0	Флаг ошибки сравнения сигналов системной шины S процессорного ядра CPU 0 и процессорного ядра CPU1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	AHB_D_COMP_ERR	0	Флаг ошибки сравнения сигналов системной шины D процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	AHB_I_COMP_ERR	0	Флаг ошибки сравнения сигналов системной шины I процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	COMP_REG_ERR	0	Ошибка сравнения вспомогательных сигналов процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.10 EVENT1 (HIGH)

Base ADDR=		0x4000_3000				Offset=		0x0000_0024							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												ECCERR_	ECCERR_	ECCERR_	ECCERR_
												EXTBUS1	EXTBUS0	RAMC1	RAMC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCERR_	ECCERR_	ECCERR_	ECCERR_	-	ECCERR_	ECCERRC_	ECCERRM_	ECCERRS_	ECCERRD_	ECCERRI_	ECCERRC_	ECCERRM_	ECCERRS_	ECCERRD_	ECCERRI_
RAMD1	RAMD0	OTP1	OTP0		ROM	1	1	1	1	1	0	0	0	0	0

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19	ECCERR_EXTBUS1*	0	Ошибка ECC на шине АНВ на EXT_BUS1 0 – нет ошибки 1 – есть ошибка
18	ECCERR_EXTBUS0*	0	Ошибка ECC на шине АНВ на EXT_BUS0 0 – нет ошибки 1 – есть ошибка
17..16	-	2'b00	Зарезервировано
15	ECCERR_RAMD1*	0	Ошибка ECC на шине АНВ на RAMD1 0 – нет ошибки 1 – есть ошибка
14	ECCERR_RAMD0*	0	Ошибка ECC на шине АНВ на RAMD0 0 – нет ошибки 1 – есть ошибка
13	ECCERR_OTP1*	0	Ошибка ECC на шине АНВ на OTP1 0 – нет ошибки 1 – есть ошибка
12	ECCERR_OTP0*	0	Ошибка ECC на шине АНВ на OTP0 0 – нет ошибки 1 – есть ошибка.
11	-	-	Зарезервировано
10	ECCERR_ROM*	0	Ошибка ECC на шине АНВ на ROM 0 – нет ошибки 1 – есть ошибка
9	-		Зарезервировано
8	ECCERRM_1*	0	Ошибка ECC на шине М на DMA 1 0 – нет ошибки 1 – есть ошибка
7	ECCERRS_1*	0	Ошибка ECC на шине S на CPU 1 0 – нет ошибки 1 – есть ошибка
6	ECCERRD_1*	0	Ошибка ECC на шине D на CPU 1 0 – нет ошибки 1 – есть ошибка

* Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
5	ECCERRI_1*	0	Ошибка ЕСС на шине I на CPU 1 0 – нет ошибки 1 – есть ошибка
4	-		Зарезервировано
3	ECCERRM_0*	0	Ошибка ЕСС на шине M на DMA 0 0 – нет ошибки 1 – есть ошибка
2	ECCERRS_0*	0	Ошибка ЕСС на шине S на CPU 0 0 – нет ошибки 1 – есть ошибка
1	ECCERRD_0*	0	Ошибка ЕСС на шине D на CPU 0 0 – нет ошибки 1 – есть ошибка
0	ECCERRI_0*	0	Ошибка ЕСС на шине I на CPU 0 0 – нет ошибки 1 – есть ошибка

7.5.1.11 EVENT2 (HIGH)

Base ADDR=	0x4000_3000				Offset=	0x0000_0028									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ECC_CACHE_RAMD_DE	ECC_CACHE_D_1_DE	ECC_CACHE_D_0_DE	ECC_CACHE_I_1_DE	ECC_CACHE_I_0_DE	ECC_EXTBUS_DE		ECC_RAMD_DE	ECC_OTP_DE	ECC_ROM_DE

Бит	Имя	Значение	Описание
31...10	-		Зарезервировано
9	ECC_CACHE_RAMD_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE в RAMD 0 – нет ошибки 1 – есть ошибка
8	ECC_CACHE_D_1_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_D в CPU1 0 – нет ошибки 1 – есть ошибка
7	ECC_CACHE_D_0_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_D в CPU0 0 – нет ошибки 1 – есть ошибка
6	ECC_CACHE_I_1_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_I в CPU1 0 – нет ошибки 1 – есть ошибка
5	ECC_CACHE_I_0_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_I в CPU0 0 – нет ошибки 1 – есть ошибка
4	ECC_EXTBUS_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти EXT_BUS 0 – нет ошибки 1 – есть ошибка
3	-	0	Зарезервировано
2	ECC_RAMD_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти RAMD 0 – нет ошибки 1 – есть ошибка
1	ECC_OTP_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти OTP 0 – нет ошибки 1 – есть ошибка
0	ECC_ROM_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти ROM 0 – нет ошибки 1 – есть ошибка

7.5.1.12 EVENT3 (HIGH)

Base ADDR=		0x4000_3000				Offset=		0x0000_002C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CLK_MONITOR7	CLK_MONITOR6	CLK_MONITOR5		CLK_MONITOR3	CLK_MONITOR2	CLK_MONITOR1	CLK_MONITOR0

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	CLK_MONITOR7	0	Резерв
6	CLK_MONITOR6	0	Резерв
5	CLK_MONITOR5	0	Резерв
4	-	0	Зарезервировано
3	CLK_MONITOR3*	0	Ошибка превышения максимальной тактовой частоты RAMD 0 – нет ошибки 1 – есть ошибка
2	CLK_MONITOR2*	0	Ошибка превышения максимальной тактовой частоты ОТР 0 – нет ошибки 1 – есть ошибка
1	CLK_MONITOR1*	0	Ошибка превышения максимальной тактовой частоты CPU1 0 – нет ошибки 1 – есть ошибка
0	CLK_MONITOR0*	0	Ошибка превышения максимальной тактовой частоты CPU0 0 – нет ошибки 1 – есть ошибка

7.5.1.13 EVENT4 (HIGH)

Base ADDR=		0x4000_3000				Offset=		0x0000_0030								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							ROM_REG_DE		RAMD_REG_DE	OTP_REG_DE	FT_REG_DE	FTCNTR_ERR	BKP_MEM_DE	BKP_ERR	RTC_ERR

Бит	Имя	Значение	Описание
31...9			
8	ROM_REG_DE*		Двойная неисправимая ошибка в регистрах управления ROM_CNTR 0 – нет ошибки 1 – есть ошибка
7	-		Зарезервировано
6	RAMD_REG_DE*		Двойная неисправимая ошибка в регистрах управления RAMD_CNTR 0 – нет ошибки 1 – есть ошибка
5	OTP_REG_DE*	0	Двойная неисправимая ошибка в регистрах управления OTP_CNTR 0 – нет ошибки 1 – есть ошибка
4	FT_REG_DE*	0	Двойная неисправимая ошибка в регистрах управления FT_CNTR 0 – нет ошибки 1 – есть ошибка
3	FTCNTR_ERR*	0	Ошибка в счетчике TIMEOUTCNT отложенного сброса FT_CNTR 0 – нет ошибки 1 – есть ошибка
2	BKP_MEM_DE*	0	Двойная ошибка в массиве памяти ВКР (00-59) 0 – нет ошибки 1 – есть ошибка
1	BKP_ERR*	0	Ошибка в конфигурационных регистрах ВКР (60-63) 0 – нет ошибки 1 – есть ошибка
0	RTC_ERR*	0	Ошибка в часах реального времени RTC 0 – нет ошибки 1 – есть ошибка

* Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.14 EVENT5 (MEDIUM)

Base ADDR=		0x4000_3000				Offset=		0x0000_0034							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			RTC_CLK_DE	DAC_CLK_DE	ADC_CLK_DE	SSP_CLK_DE	UART_CLK_DE		SPW_CLK_DE	EPHY_CLK_DE			MIL_CLK_DE	TIM_CLK_DE	CAN_CLK_DE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PLL2_CLK_DE	PLL1_CLK_DE	PLL0_CLK_DE	HSE1_CLK_DE	HSE0_CLK_DE	LSE_CLK_DE	LSI_CLK_DE	PER_CLK_DE	CPU_CLK_DE	MAX_CLK_DE	KEY_DE

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28	RTC_CLK_DE		Двойная неисправимая ошибка в регистре RTC_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
27	DAC_CLK_DE		Двойная неисправимая ошибка в регистре DAC_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
26	ADC_CLK_DE		Двойная неисправимая ошибка в регистре ADC_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
25	SSP_CLK_DE		Двойная неисправимая ошибка в регистре SSP_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
24	UART_CLK_DE		Двойная неисправимая ошибка в регистре UART_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
23			Зарезервировано

Бит	Имя	Значение	Описание
22	SPW_CLK_DE		Двойная неисправимая ошибка в регистре SPW_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
21	EPHY_CLK_DE		Двойная неисправимая ошибка в регистре EPHY_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
20, 19	-		Зарезервировано
18	MIL_CLK_DE		Двойная неисправимая ошибка в регистре MIL_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
17	TIM_CLK_DE		Двойная неисправимая ошибка в регистре TIM_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
16	CAN_CLK_DE		Двойная неисправимая ошибка в регистре CAN_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
15...11			Зарезервировано
10	PLL2_CLK_DE		Двойная неисправимая ошибка в регистре PLL2_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
9	PLL1_CLK_DE		Двойная неисправимая ошибка в регистре PLL1_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
8	PLL0_CLK_DE		Двойная неисправимая ошибка в регистре PLL0_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
7	HSE1_CLK_DE		Двойная неисправимая ошибка в регистре HSE1_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
6	HSE0_CLK_DE		Двойная неисправимая ошибка в регистре HSE0_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5	LSE_CLK_DE		Двойная неисправимая ошибка в регистре LSE_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	LSI_CLK_DE		Двойная неисправимая ошибка в регистре LSI_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	PER_CLK_DE		Двойная неисправимая ошибка в регистре PER_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	CPU_CLK_DE		Двойная неисправимая ошибка в регистре CPU_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	MAX_CLK_DE		Двойная неисправимая ошибка в регистре MAX_CLK блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	KEY_DE		Двойная неисправимая ошибка в регистре KEY блока CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.15 EVENT6 (MEDIUM)

Base ADDR=		0x4000_3000				Offset=		0x0000_0038							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									PLL2_CLOCK_ERROR	PLL1_CLOCK_ERROR	PLL0_CLOCK_ERROR	LSE_CLOCK_ERROR	LSI_CLOCK_ERROR	HSE1_CLOCK_ERROR	HSE0_CLOCK_ERROR
															CPU_CLOCK_ERROR

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	PLL2_CLOCK_ERROR		Ошибка при формировании частоты PLL2_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
6	PLL1_CLOCK_ERROR		Ошибка при формировании частоты PLL1_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5	PLL0_CLOCK_ERROR		Ошибка при формировании частоты PLL0_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	LSE_CLOCK_ERROR		Ошибка при формировании частоты LSE_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасываются записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	LSI_CLOCK_ERROR		Ошибка при формировании частоты LSI_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	HSE1_CLOCK_ERROR		Ошибка при формировании частоты HSE1_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
1	HSE0_CLOCK_ERROR		Ошибка при формировании частоты HSE0_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	CPU_CLOCK_ERROR		Ошибка при формировании частоты CPU_CLK в блоке CLK_CHECKER в блоке CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.16 EVENT7 (MEDIUM)

Base ADDR=	0x4000_3000	Offset=	0x0000_003C													
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RAMD_CACHE_SE	ROM_REG_SE	.	RAMD_REG_SE	OTP_REG_SE	FT_REG_SE	BKP_MEM_SE	.	RAMD_SE	ECC_CACHE_D_1_SE	ECC_CACHE_D_0_SE	ECC_CACHE_I_1_SE	ECC_CACHE_I_0_SE	ROM_SE	OTP_SE

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14	RAMD_CACHE_SE		Одиночная ошибка в регистрах блока кэш-памяти RAMD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
13	ROM_REG_SE		Одиночная ошибка в регистрах управления блока ROMCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
12	-		Зарезервировано
11	RAMD_REG_SE		Одиночная ошибка в регистрах управления блока RAMD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
10	OTP_REG_SE	0	Одиночная ошибка в регистрах управления блока OTPCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
9	FT_REG_SE	0	Одиночная ошибка в регистрах управления блока FTCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
8	ВКР_MEM_SE	0	Одиночная ошибка в массиве памяти ВКР (00-59) 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
7	-	0	Зарезервировано
6	RAMD_SE	0	Одиночная ошибка в массиве памяти RAMD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5	ECC_CACHE_D_1_SE	0	Одиночная ошибка в массиве памяти CACHEDCPU1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	ECC_CACHE_D_0_SE	0	Одиночная ошибка в массиве памяти CACHEDCPU0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	ECC_CACHE_I_1_SE	0	Одиночная ошибка в массиве памяти CACHEICPU1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	ECC_CACHE_I_0_SE	0	Одиночная ошибка в массиве памяти CACHEICPU0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	ROM_SE	0	Одиночная ошибка в массиве памяти ROM 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	ОТР_SE	0	Одиночная ошибка в массиве памяти ОTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.17 EVENT8 (MEDIUM)

Base ADDR=		0x4000_3000				Offset=		0x0000_0040								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					LDO7_CUR_EVENT	LDO6_CUR_EVENT	LDO5_CUR_EVENT	LDO4_CUR_EVENT	LDO3_CUR_EVENT	LDO2_CUR_EVENT	LDO1_CUR_EVENT	LDO0_CUR_EVENT	PVD_BUCC_EVENT	PVD_UCC1_EVENT	PVD_UCC_0_EVENT

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10	LDO7_CUR_EVENT	0	Ошибка по току потребления LDO7 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
9	LDO6_CUR_EVENT	0	Ошибка по току потребления LDO6 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
8	LDO5_CUR_EVENT	0	Ошибка по току потребления LDO5 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
7	LDO4_CUR_EVENT	0	Ошибка по току потребления LDO4 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
6	LDO3_CUR_EVENT	0	Ошибка по току потребления LDO3 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5	LDO2_CUR_EVENT	0	Ошибка по току потребления LDO2 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	LDO1_CUR_EVENT	0	Ошибка по току потребления LDO1 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
3	LDO0_CUR_EVENT	0	Ошибка по току потребления LDO0 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	PVD_BUCC_EVENT	0	Ошибка по питанию BUcc от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	PVD_UCC1_EVENT	0	Ошибка по питанию Ucc1 от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	PVD_UCC_EVENT	0	Ошибка по питанию Ucc от блока PVD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.18 EVENT9 (LOW)

Base ADDR=		0x4000_3000				Offset=		0x0000_0044								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
			RTC_CLK_SE	DAC_CLK_SE	ADC_CLK_SE	SSP_CLK_SE	UART_CLK_SE		SPW_CLK_SE	EPHY_CLK_SE	EMAC_CLK_SE	ARC_CLK_SE		TIM_CLK_SE	CAN_CLK_SE	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PLL2_CLK_SE	PLL1_CLK_SE	PLL0_CLK_SE	HSE1_CLK_SE	HSE0_CLK_SE	LSE_CLK_DE	LSI_CLK_DE	PER_CLK_SE	CPU_CLK_SE	MAX_CLK_SE	KEY_SE

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28	RTC_CLK_SE		Одиночная исправимая ошибка в регистре RTC_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
27	DAC_CLK_SE		Одиночная исправимая ошибка в регистре DAC_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
26	ADC_CLK_SE		Одиночная исправимая ошибка в регистре ADC_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
25	SSP_CLK_SE		Одиночная исправимая ошибка в регистре SSP_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
24	UART_CLK_SE		Одиночная исправимая ошибка в регистре UART_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
23	-		Зарезервировано
22	SPW_CLK_SE		Одиночная исправимая ошибка в регистре SPW_CLK блока CLKNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
21	EPHY_CLK_SE		Одиночная исправимая ошибка в регистре EPHY_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
20	EMAC_CLK_SE		Одиночная исправимая ошибка в регистре EMAC_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
19	ARC_CLK_SE		Одиночная исправимая ошибка в регистре ARC_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
18	-		Зарезервировано
17	TIM_CLK_SE		Одиночная исправимая ошибка в регистре TIM_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
16	CAN_CLK_SE		Одиночная исправимая ошибка в регистре CAN_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
15...11	-		Зарезервировано
10	PLL2_CLK_SE		Одиночная исправимая ошибка в регистре PLL2_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
9	PLL1_CLK_SE		Одиночная исправимая ошибка в регистре PLL1_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
8	PLL0_CLK_SE		Одиночная исправимая ошибка в регистре PLL0_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
7	HSE1_CLK_SE		Одиночная исправимая ошибка в регистре HSE1_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
6	HSE0_CLK_SE		Одиночная исправимая ошибка в регистре HSE0_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5	LSE_CLK_SE		Одиночная исправимая ошибка в регистре LSE_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	LSI_CLK_SE		Одиночная исправимая ошибка в регистре LSI_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	PER_CLK_SE		Одиночная исправимая ошибка в регистре PER_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	CPU_CLK_SE		Одиночная исправимая ошибка в регистре CPU_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	MAX_CLK_SE		Одиночная исправимая ошибка в регистре MAX_CLK блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	KEY_SE		Одиночная исправимая ошибка в регистре KEY блока CLKCNTNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.19 EVENT10 (LOW)

Base ADDR=	0x4000_3000				Offset=	0x0000_0048									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PORTF_CURLIM	PORTE_CURLIM	PORTD_CURLIM	PORTC_CURLIM	PORTB_CURLIM	PORTA_CURLIM

Бит	Имя	Значение	Описание
31...6	-		Зарезервировано
5	PORTF_CURLIM		Ошибка превышения выходного тока в PORTF 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
4	PORTE_CURLIM		Ошибка превышения выходного тока в PORTE 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
3	PORTD_CURLIM		Ошибка превышения выходного тока в PORTD 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
2	PORTC_CURLIM		Ошибка превышения выходного тока в PORTC 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	PORTB_CURLIM		Ошибка превышения выходного тока в PORTB 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
0	PORTA_CURLIM		Ошибка превышения выходного тока в PORTA 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.20 EVENT11 (LOW)

Base ADDR=		0x4000_3000					Offset=		0x0000_004C						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMAC0_DE			ARC1_DE	ARC0_DE			MIL1_DE	MIL0_DE						CAN0_DE

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14	EMAC0_DE	0	Двойная неисправимая ошибка в массивах памяти ETHERNETMAC0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
13, 12	-		Зарезервировано
11	ARC1_DE	0	Двойная неисправимая ошибка в массивах памяти ARC1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
10	ARC0_DE	0	Двойная неисправимая ошибка в массивах памяти ARC0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
9, 8	-		Зарезервировано
7	MIL1_DE	0	Двойная неисправимая ошибка в массивах памяти MIL1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
6	MIL0_DE	0	Двойная неисправимая ошибка в массивах памяти MIL0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5...1	-		Зарезервировано
0	CAN0_DE	0	Двойная неисправимая ошибка в массивах памяти CAN0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.21 EVENT12 (LOW)

Base ADDR=		0x4000_3000					Offset=		0x0000_0050						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMAC0_DE			ARC1_DE	ARC0_DE			MIL1_DE	MIL0_DE						CAN0_DE

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14	EMAC0_SE	0	Одиночная неисправимая ошибка в массивах памяти ETHERNETMAC0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
13, 12	-		Зарезервировано
11	ARC1_SE	0	Одиночная неисправимая ошибка в массивах памяти ARC1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
10	ARC0_SE	0	Одиночная неисправимая ошибка в массивах памяти ARC0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
9, 8	-		Зарезервировано
7	MIL1_SE	0	Одиночная неисправимая ошибка в массивах памяти MIL1 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
6	MIL0_SE	0	Одиночная неисправимая ошибка в массивах памяти MIL0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
5...1	-		Зарезервировано
0	CAN0_SE	0	Одиночная неисправимая ошибка в массивах памяти CAN0 0 – нет ошибки 1 – есть ошибка Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.

7.5.1.22 RESET_EVENT0

Base ADDR=		0x4000_3000				Offset=		0x0000_0054								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		RESET_EN[13:0]													

Бит	Имя	Значение	Описание
31...14	-		Зарезервировано
13...0	RESET_EN[13:0]	14'h3FFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT0 0 – сброс не формируется 1 – сброс формируется

7.5.1.23 RESET_EVENT1

Base ADDR=		0x4000_3000				Offset=		0x0000_0058								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
												RESET_EN[19:16]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_EN[15:0]															

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...0	RESET_EN[19:0]	20'h1FFFFFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT1 0 – сброс не формируется 1 – сброс формируется

7.5.1.24 RESET_EVENT2

Base ADDR=		0x4000_3000				Offset=		0x0000_005C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_EN[9:0]															

Бит	Имя	Значение	Описание
31...10	-		Зарезервировано
9...0	RESET_EN[9:0]	10'h3FF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT2 0 – сброс не формируется 1 – сброс формируется

7.5.1.25 RESET_EVENT3

Base ADDR=		0x4000_3000				Offset=		0x0000_0060							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_EN[3:0]															

Бит	Имя	Значение	Описание
31...4	-		Зарезервировано
3...0	RESET_EN[3:0]	4'hF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT3 0 – сброс не формируется 1 – сброс формируется

7.5.1.26 RESET_EVENT4

Base ADDR=		0x4000_3000				Offset=		0x0000_0064							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
							RESET_EN[8:0]									

Бит	Имя	Значение	Описание
31...9	-		Зарезервировано
8...0	RESET_EN[8:0]	9'h1FF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT4 0 – сброс не формируется 1 – сброс формируется

7.5.1.27 IE_EVENT5

Base ADDR=		0x4000_3000				Offset=		0x0000_0068							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			INT_EN[28]	INT_EN[27]	INT_EN[26]	INT_EN[25]	INT_EN[24]	-	INT_EN[22]	INT_EN[21]	-	-	-	INT_EN[17]	INT_EN[16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					INT_EN[10]	INT_EN[9]	INT_EN[8]	INT_EN[7]	INT_EN[6]	INT_EN[5]	INT_EN[4]	INT_EN[3]	INT_EN[2]	INT_EN[1]	INT_EN[0]

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...0	INT_EN[28:0]	29'h0	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT5 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.28 IE_EVENT6

Base ADDR=	0x4000_3000					Offset=	0x0000_006C										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
								INT_EN[7:0]										

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	INT_EN[7:0]	8'h00	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT6 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.29 IE_EVENT7

Base ADDR=	0x4000_3000					Offset=	0x0000_0070										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	INT_EN[14:0]															

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14...0	INT_EN[14:0]	15'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT7 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.30 IE_EVENT8

Base ADDR=	0x4000_3000					Offset=	0x0000_0074										
REG Name:																	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN[10:0]															

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...0	INT_EN[10:0]	11'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT8 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.31 IE_EVENT9

Base ADDR=	0x4000_3000	Offset=	0x0000_0078												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_EN[28:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN[15:0]															

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...0	INT_EN[28:0]	29'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT9 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.32 IE_EVENT10

Base ADDR=		0x4000_3000				Offset=		0x0000_007C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										INT_EN[5:0]					

Бит	Имя	Значение	Описание
31...6	-		Зарезервировано
5...0	INT_EN[5:0]	6'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT10 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.33 IE_EVENT11

Base ADDR=		0x4000_3000				Offset=		0x0000_0080								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT_EN[14:0]														

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14...0	INT_EN[14:0]	15'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT11 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.5.1.34 IE_EVENT12

Base ADDR=		0x4000_3000				Offset=		0x0000_0084								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT_EN[14:0]														

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14...0	INT_EN[14:0]	15'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT12 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

7.6 Контроллер сторожевых таймеров (WDTCNTR)

В микроконтроллере реализован сторожевой таймер, при использовании которого, микроконтроллер может быть сброшен при несоблюдении временных интервалов обязательного периода сброса.

Стороживой таймер предназначен для аварийного перезапуска микроконтроллера, когда за период счета сторожевого таймера не будет выполнена хотя бы одна операция по его перезапуску. Стороживой таймер выполнен в виде сбоеустойчивой логики, т.е. при возникновении одиночного сбоя, таймер продолжит выполнять поставленную задачу. Для последующего устранения последствий сбоя требуется программная переинициализация таймера. Конфигурация блока защищена с помощью ECC кодирования, а сам счетчик построен по троированной схеме.

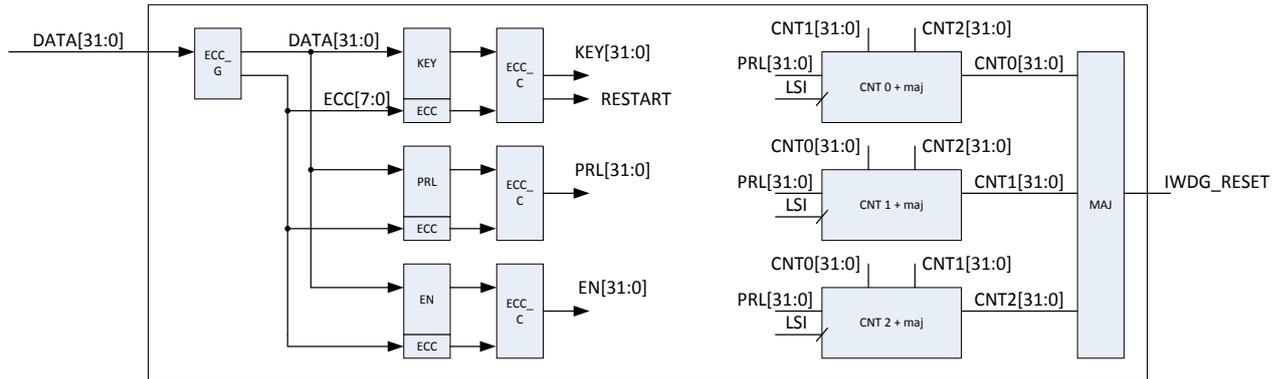


Рисунок 37 – Структурная схема контроллера троированного сторожевого таймера

Для инициализации сторожевого таймера необходимо:

- 1 В регистр KEY записать 0x05555;
- 2 Считать регистр и убедиться, что записано верное значение;
- 3 В регистр PRL записать требуемое основание счета;
- 4 Считать регистр и убедиться, что записано верное значение;
- 5 Записать в регистр KEY значение 0x0AAAA.

Через время не менее 200 мкс значение PRL должно быть автоматически переписано в регистр CNT.

- 1 Считать регистр CNT и убедиться, что в него записано верное значение из регистра PRL;
- 2 В регистр KEY записать 0x0CCCC;
- 3 Считать регистр и убедиться, что записано верное значение;
- 4 В регистр EN записать значение 0x03333. После чего сторожевой таймер начнет работу;
- 5 При возникновении сбоев провести операции по переинициализации сторожевого таймера;
- 6 С периодом меньшим заданного через регистр PRL проводить перезапуск сторожевого таймера;
- 7 Контролировать, что генератор LSI работает.

Для перезапуска сторожевого таймера необходимо:

- 1 При необходимости считать значение CNT и проверить, что оно находится в ожидаемом диапазоне;
- 2 Записать в регистр KEY значение 0x0AAAA;

Для переинициализации сторожевого таймера необходимо:

- 1 В регистр KEY записать 0x0CCCC;
- 2 Считать регистр и убедиться, что записано верное значение;
- 3 В регистр EN записать значение отличное от 0x033333. После чего будет остановлен счет;
- 4 Выполнить операции по инициализации сторожевого таймера.

7.6.1 Описание регистров

Рисунок 38 – Описание регистров контроллера сторожевых таймеров

Базовый Адрес	Название	Описание
0x4000_4000	MDR_WDT	Сторожевой таймер WDTCNTR
Смещение		
0x0000_0000	KEY	Регистр Ключей
0x0000_0004	PRL	Регистр периода счета сторожевого таймера
0x0000_0008	EN	Регистр разрешения работы
0x0000_000C	CNT	Значение счетчика сторожевого таймера

7.6.1.1 KEY

Base ADDR=	0x4000_4000	Offset=	0x0000_0000												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0	Регистр управляющих ключей блока сторожевого таймера. 0x05555 – значение, разрешающее запись в регистр периода PRL счета сторожевого таймера. 0x0CCCC – значение, разрешающее запись в регистр разрешения работы EN сторожевого таймера. Запись 0x0AAAA – перезапускает счет сторожевого таймера. Значения, отличные от указанных выше, не имеют значения.

7.6.1.2 PRL

Base ADDR=		0x4000_4000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
PRL[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PRL[15:0]																

Бит	Имя	Значение	Описание
31...0	PRL[31:0]	0	Регистр периода PRL счета сторожевого таймера Определяет период работы сторожевого таймера. Счетчик сторожевого таймера CNT считает от значения PRL до 0. При достижении нулевого значения в счетчике CNT формируется сигнал сброса IWDG_RESET Счетчик CNT считает на частоте LSI генератора. При значении PRL = 0xFF должен перезапускаться не реже чем раз в 2,5 мс При значении PRL = 0xFFFF должен перезапускаться не реже чем раз в 655 мс При значении PRL = 0xFFFFFFFF должен перезапускаться не реже чем раз в 10 часов.

7.6.1.3 EN

Base ADDR=		0x4000_4000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
EN[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
EN[15:0]																

Бит	Имя	Значение	Описание
31...0	EN[31:0]	0	Значение разрешения работы счетчика 0x03333 – счетчик работает Любые отличные значения останавливают счетчик

7.6.1.4 CNT

Base ADDR=		0x4000_4000				Offset=		0x0000_000C									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
CNT[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Бит	Имя	Значение	Описание
31...0	CNT[31:0]		Текущее значение счетчика CNT При чтении надо учитывать, что счетчик CNT работает на частоте LSI (~40 кГц), и при высокой частоте работы ядра может потребоваться дополнительное время для обновления счетчика.

7.7 Контроллер внешней шины (EXTBUSCNTR)

С помощью контроллера внешней шины EXTBUS можно организовать до 8-ми регионов для подключения внешних устройств с отображением в адресное пространство микроконтроллера по приведенным ниже диапазонам (Таблица 28).

Таблица 28 – Базовые адреса внешней системной шины

Номер	Начальный адрес	Конечный адрес	Шина доступа
REGION0	0x1000_0000	0x17FF_FFFF	I, D, M
REGION1	0x1800_0000	0x1FFF_FFFF	I, D, M
REGION2	0x5000_0000	0x57FF_FFFF	S, M
REGION3	0x5800_0000	0x5FFF_FFFF	S, M
REGION4	0x6000_0000	0x67FF_FFFF	S, M
REGION5	0x6800_0000	0x6FFF_FFFF	S, M
REGION6	0x7000_0000	0x7FFF_FFFF	S, M
REGION7	0x8000_0000	0xDFFF_FFFF	S, M

Для каждого региона может быть задан свой режим работы функционирования. При отображении выводов контроллера внешней шины на пользовательские выводы этим выводам необходимо задать функцию внешней шины. Если выводу не задана функция внешней шины, то с него считывается ноль.

7.7.1 Организация доступа к внешней шине EXTBUS

Таблица 29 – Интерфейс контроллера EXT_BUS

Имя входа	Направление	Описание
ADDR[31:0]	Выход	Шина адреса
DATA[31:0] DATA[63:32]	Вход/Выход	Шина данных
DATA[71:64] DATA[79:72]	Вход/Выход	Шина данных ECC
BE _n [9:0]	Выход	Выход ByteEnable 0 – разрешение байта 1 – запрет байта BE _n [0] – DATA _o [7:0] BE _n [1] – DATA _o [15:8] ... BE _n [7] – DATA _o [63:54] BE _n [8] – DATA _o [71:64] BE _n [9] – DATA _o [79:72]
CS _n [7:0]	Выход	Выход разрешения региона 0 – регион выбран 1 – регион не выбран
CS[7:0]	Выход	Выход разрешения региона 0 – регион не выбран 1 – регион выбран
WE _n [1:0]	Выход	Выход сигнала записи 0 – запись 1 – нет записи WE _n [0] – DATA _o [31:0] + DATA _o [71:64] WE _n [1] – DATA _o [63:32] + DATA _o [79:71]
OE _n [1:0]	Выход	Сигнал чтения данных с внешних выводов 0 – есть чтение 1 – нет чтения OE _n [0] - DATA _i [31:0] + DATA _i [71:64] OE _n [1] - DATA _i [63:32] + DATA _i [79:72]

Имя входа	Направление	Описание
BWEn[9:0]	Выход	Выход сигнала записи с сигналом ByteEnable BWEn[0] = WEn[0] or BEn[0] BWEn[1] = WEn[0] or BEn[1] ... BWEn[7] = WEn[1] or BEn[7] BWEn[8] = WEn[0] or BEn[8] BWEn[9] = WEn[1] or BEn[9]
READY[7:0]	Вход	Вход готовности от блоков памяти с сигналом готовности 0 – не готов 1 – готов
CLOCK	Выход	Выход сигнала синхронизации внутри транзакции
OCLK	Выход	Выход тактового сигнала, формируемого контроллером внешней системной шины

Внешняя шина может иметь 8-ти, 16-ти, 32-х и 64-х битную организацию шины данных (без учета проверочных бит ECC при параллельной организации). При этом процессорное ядро может оперировать только 8-ти, 16-ти и 32-х битными данными на внешней шине. Реализация 64-х битной шины возможна и имеет смысл только для выполнения программного кода из внешней памяти. В этом случае из внешней памяти извлекается сразу два последовательных 32-х разрядных слова (запрошенное четное и следующее за ним или запрошенное нечетное и предыдущее перед ним). Это позволяет повысить производительность при исполнении кода из внешней памяти при последовательной выборке, так как в момент обращения к следующему слову оно уже считано и хранится в контроллере, и чтение на внешней шине не выполняется. Реализация 64-х битной шины для работы с какой-либо внешней периферией недопустима, так как упреждающее чтение может повлиять на информацию критичную к чтению (буфера FIFO, регистры данных различных контроллеров интерфейсов и т.п.) Внешняя шина имеет 8 программно управляемых диапазонов. Для каждого диапазона настраивается собственный режим работы, включая временные характеристики транзакции на шине, разрядность шины, способ обращения с ожиданием или без сигнала готовности, и способ организации ECC.

Диаграммы работы контроллера при записи и чтении представлены на рисунках и соответственно.

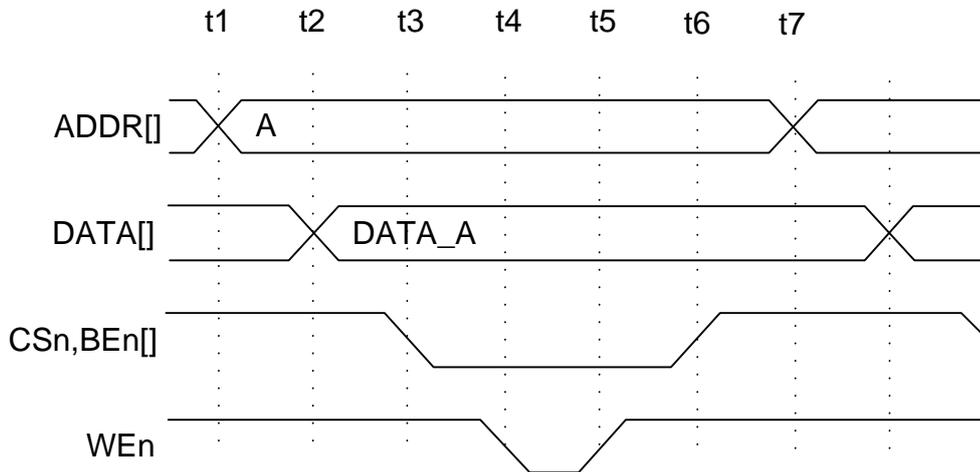


Рисунок 39 – Диаграмма записи на внешней шине

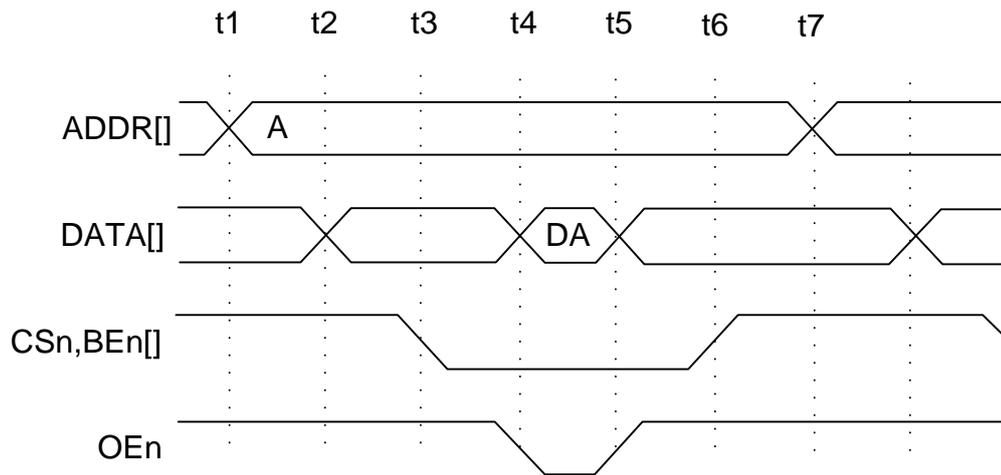


Рисунок 40 – Диаграмма чтения на внешней шине

В момент времени t_1 начинается транзакция.

$$t_2 - t_1 = t_{HCLK};$$

$$t_3 - t_2 = t_{HCLK};$$

$$t_4 - t_3 = t_{HCLK} * (WS_SETUP + 1);$$

$$t_5 - t_4 = t_{HCLK} * (WS_ACTIVE + 1);$$

$$t_6 - t_5 = t_{HCLK} * (WS_HOLD + 1);$$

$$t_7 - t_6 = t_{HCLK};$$

В момент времени t_7 может быть начата следующая транзакция.

7.7.1.1 Организация ECC для внешней системной шины

Для каждого из диапазонов внешней системной шины может быть задан режим работы с проверкой ECC или без проверки. Если проводится проверка ECC, то она организована по коду Хемминга (72,64). При этом проверочные биты ECC могут располагаться как в дополнительных разрядах шины данных (32 данные + 8 ECC – параллельная организация), так и непосредственно в самой 32-х разрядной памяти, начиная с программно задаваемого базового адреса расположения ECC для каждого региона.

7.7.1.2 Параллельная организация ECC

Параллельная организация ECC для внешней системной шины возможна только при организации 64-х и 32-х разрядной шины данных. При этом дополнительно к каждому 32 разрядам данных добавляется 8 разрядов ECC, т.е. при 64-х битной шине данных общая разрядность составляет 80 бит, а при 32-х разрядной – 40 бит. При настройке 16-ти или 8-ми битной шины с параллельной организацией ECC обращение к внешней шине будет приводить к исключению BUSERROR. При 64-х битной организации процессор может за одно обращение считать максимум только 32-х битное число, при этом второе 32-х битное число может быть «закешировано» (только если CACHEBLE = 1) внутри контроллера внешней системной шины. При последующем обращении по адресу слова, оно будет считано не из внешней памяти, а из внутреннего регистра. Это позволяет увеличить скорость чтения. Кэширование в контроллере внешней шины возможно только при 64-ти битной организации, при установке бита CACHEBLE = 1 данного региона и при условии, что кэшируемое слово считано без ошибок (нет ни двойных, не одинарных ошибок). Если при чтении «кэшируемого» слова была обнаружена одинарная или двойная ошибка, это слово не будет «закешировано». В регистр счетчика ошибок, регистр адреса и данных ошибочного слова не заносится информация об ошибках, возникших при чтении кэшируемого слова. В регистры счетчика ошибок, регистр адреса и данных ошибочного слова заносится информация только того слова, которое было запрошено. Кеширование данных в контроллере внешней шины при 64 битной организации никак не связано с кешированием в блоках кэш памяти DCACHE и ICACHE. Это два различных механизма.

При записи 8-ми и 16-ти разрядных слов во внешнюю системную шину обращение выполняется в два этапа:

- 1 выполняется считывание из внешней системной шины по адресу обращения данных с ECC, проверка ECC считанных данных;
- 2 модификация 8-ми или 16-ти разрядного слова в считанном и проверенном по ECC 32-х разрядном слове, генерация ECC для модифицированного слова и запись в память.

При записи 32-х разрядного слова за один этап проводится генерация ECC и запись в память.

Запись 64-х разрядного слова не возможна, либо выполняется двумя записями 32-х разрядных слов.

При чтении 8-ми или 16-ти разрядных слов из памяти всегда извлекается 32-х разрядное слово с ECC (либо два 32-х разрядных слова при 64-х битной организации). В контроллере внешней шины проверяется ECC, генерируется новое ECC и возвращается процессору, на входе которого ECC проверяется еще раз.

Таблица 30 – Число транзакций на шине EXT_BUS в зависимости от разрядности операции

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	32,16,8	64	1 чтение 2-х 32-х разрядных слов
Запись	32	64	1 запись 32-х разрядного слова
Запись	16,8	64	1 чтение одного 32-х разрядного слова + 1 запись одного 32-х разрядного слова
Чтение	32,16,8	32	1 чтение
Запись	32	32	1 запись
Запись	16,8	32	1 чтение + 1 запись

7.7.1.3 Последовательная организация ECC

Последовательная организация ECC для внешней системной шины возможна для всех разрядностей шины данных, но при этом контроллер внешней системной шины все равно манипулирует 32-х разрядными словами. При 64-х битной организации шины, второе 32-х разрядное слово выбирается на случай повторного последовательного обращения (если это разрешено в контроллере CACHEBLE = 1). Таким образом, независимо от реализации памяти на внешней системной шине для процессора отображается 32-х разрядная память. Контрольные ECC биты для памяти располагаются в старших адресах каждого диапазона, начиная с программно задаваемого базового адреса ECCBASE (младшие 4 бита адреса ECCBASE игнорируются и считаются равными нулю). В зависимости от организации шины их расположение различно.

Для 8-ми, 16-ти и 32-х битных шин данных принцип расположения данных представлен на рисунке .

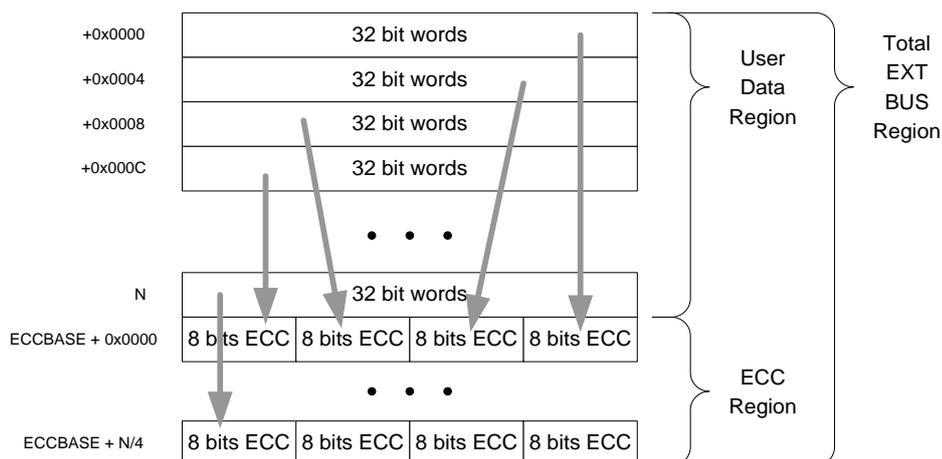


Рисунок 41 – Расположение данных в памяти на шине EXT_BUS для 32-ти,16-ти,8-ми битных шин

Для 64-х битной организации принцип расположения представлен на рисунке .

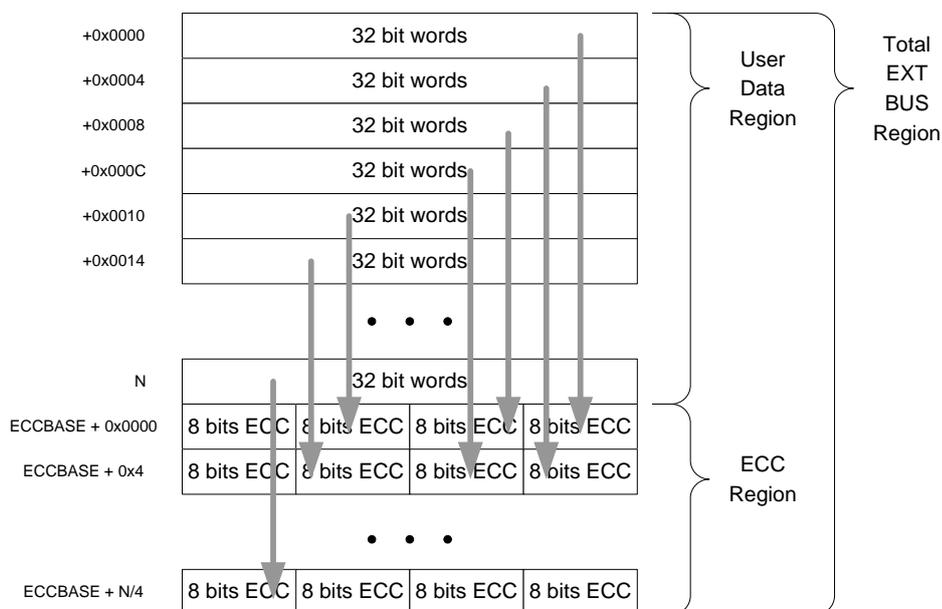


Рисунок 42 – Расположение данных в памяти на шине EXT_BUS при 64-х битной шине

Внимание! ECC-биты при последовательной организации располагаются последовательно, начиная с адреса ECCBASE. ECC-биты для четных слов расположены последовательно, но в четных ячейках, начиная с ECCBASE, для нечетных слов – последовательно, но в нечетных ячейках начиная с ECCBASE+0x4. Это необходимо для минимизации повторных обращений при чтении ECC в случае, если проверочные биты ECC от считанных слов попадут в разные ячейки.

Такая организация памяти требует:

- два обращения для считывания четырех 32-х разрядных слов при 64-х битной организации;
- два обращения для считывания 32-х данных с ECC при 32-х разрядной шине данных;
- три обращения при 16-ти разрядной шине данных;
- пять обращений при 8-ми разрядной шине данных.

Число обращений для выполнения операций чтения и записи данных различной разрядности при различной организации шины данных представлено в таблице ниже (Таблица 31)

Таблица 31 – Число транзакций на шине EXT_BUS при последовательном ECC

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	32,16,8	64	2 чтения
Чтение	32,16,8	32	2 чтения
Чтение	32,16,8	16	3 чтения
Чтение	32,16,8	8	5 чтений
Запись	32	64	2 записи
Запись	32	32	2 записи
Запись	32	16	3 записи
Запись	32	8	5 записей
Запись	16,8	64	2 чтения + 2 записи
Запись	16,8	32	2 чтения + 2 записи
Запись	16,8	16	3 чтения + 2 записи
Запись	16	8	5 чтений + 3 записи
Запись	8	8	5 чтений + 2 записи

Область выше ECCBASE, предназначенная для хранения ECC может быть считана/записана как обычные данные. Для обращения в эту область необходимо предварительно перейти в режим работы без контроля ECC. Во избежание порчи данных в этой области, программно может быть установлена защита от записи по адресам выше ECCBASE.

7.7.1.4 Вычисление адреса расположения ECC-бит

При параллельной организации ECC-биты располагаются в расширении шины данных и считываются одновременно со словом. При последовательной организации ECC-биты хранятся в том же массиве памяти, что и проверяемые слова, но начиная с адреса ECCBASE. И для их считывания выполняются дополнительные циклы чтения или записи. При последовательной организации адрес ECC-бит вычисляется согласно формулам:

$$ADDR_{ECC}[31:0] = ECCBASE[31:0] + \{7'b0, ADDR[26:2]\};$$

при 8-ми, 16-ти, 32-х битной шине данных

$$ADDR_{ECC}[31:0] = ECCBASE[31:0] + \{7'b0, ADDR[26:5], ADDR[2], ADDR[4:3]\};$$

при 64-х битной шине данных

7.7.1.5 Организация шин без контроля ECC

При отсутствии возможности или необходимости организации контроля ECC для диапазона внешней памяти может быть установлен режим работы без контроля ECC для определенного диапазона. В этом случае данные считываются из внешней памяти без контроля ECC. При этом для возможности кэширования данных для них в контроллере внешней шины будет сгенерирован ECC при возвращении данных процессору. В остальном данный режим работает аналогично режиму с параллельной организацией ECC. При необходимости выполняются дополнительные циклы чтения для формирования 32-х разрядных данных, что необходимо учитывать при работе с внешними периферийными блоками, критичными к чтению (FIFO, и т. п.).

Таблица 32 – Число транзакций на шине EXT_BUS в режиме работы без контроля ECC

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	32,16,8	64	1 чтение
Чтение	32,16,8	32	1 чтения
Чтение	32	16	2 чтения
Чтение	32	8	4 чтения

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	16,8	16	1 чтение
Чтение	16	8	2 чтения
Чтение	8	8	1 чтение
Запись	32	64	1 запись
Запись	32	32	1 запись
Запись	32	16	2 записи
Запись	32	8	4 записи
Запись	16,8	64	1 чтение + 1 запись
Запись	16,8	32	1 чтение + 1 запись
Запись	16	16	1 запись
Запись	16	8	2 записи
Запись	8	16	1 чтение +1 запись
Запись	8	8	1 запись

7.7.1.6 Организация 64-х, 32-х, 16-ти и 8-ми разрядной внешней системной шины

Для внешней системной шины может быть организована шина данных различной разрядности. При параллельной организации ЕСС допускается только 64-х и 32-х разрядная организация шины данных (64 разряда данных + 16 разрядов ЕСС и 32 разряда данных + 8 разрядов ЕСС). При последовательной организации ЕСС может быть задана разрядность шины данных 64, 32, 16 и 8 разрядов. При организации без контроля ЕСС в контроллере внешней системной шины может быть задана разрядность шины данных 64, 32, 16 и 8 разрядов, но при назначении функций портов микроконтроллера она может быть сокращена вплоть до 0 бит. В этом случае разряды шины, для которых не задана функция выводов шины данных, внешней системной шины всегда читаются как НОЛЬ.

7.7.1.7 Примеры организаций 64-х, 32-х, 16-ти и 8-ми разрядной внешней системной шины

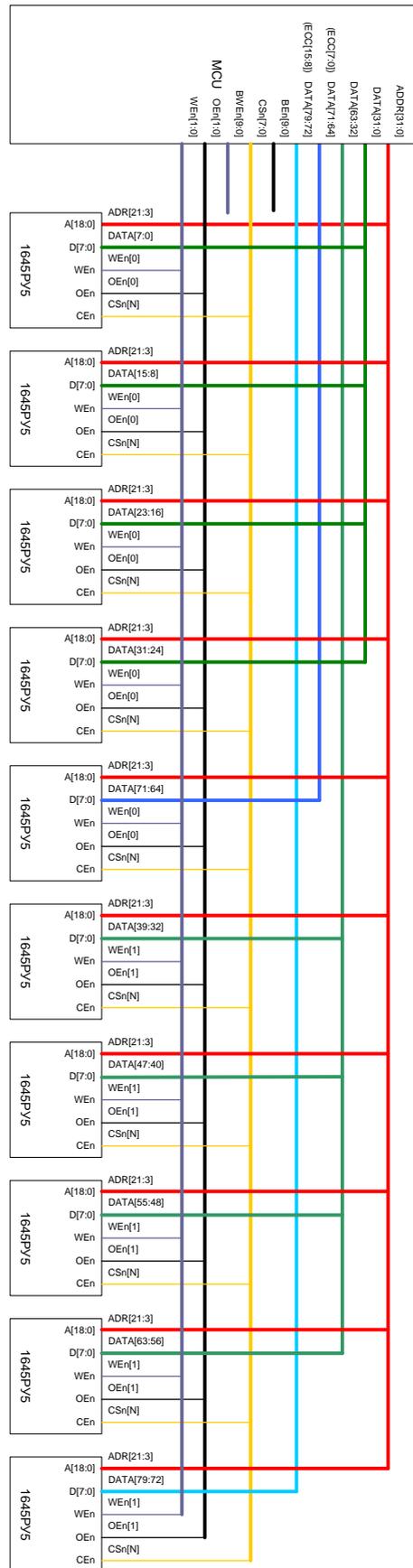


Рисунок 43 – Организация 64-х битной шины с параллельной ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 4 МБайт)

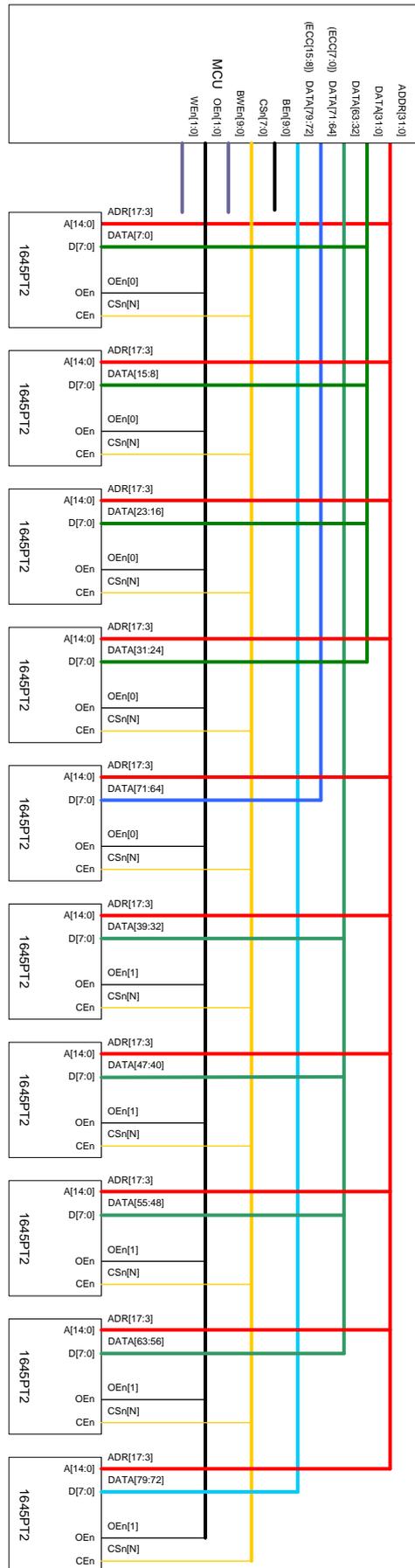


Рисунок 44 – Организация 64-х битной шины с параллельной ECC на базе ПЗУ 1645PT2 (объем пользовательских данных до 256 Кбайт)

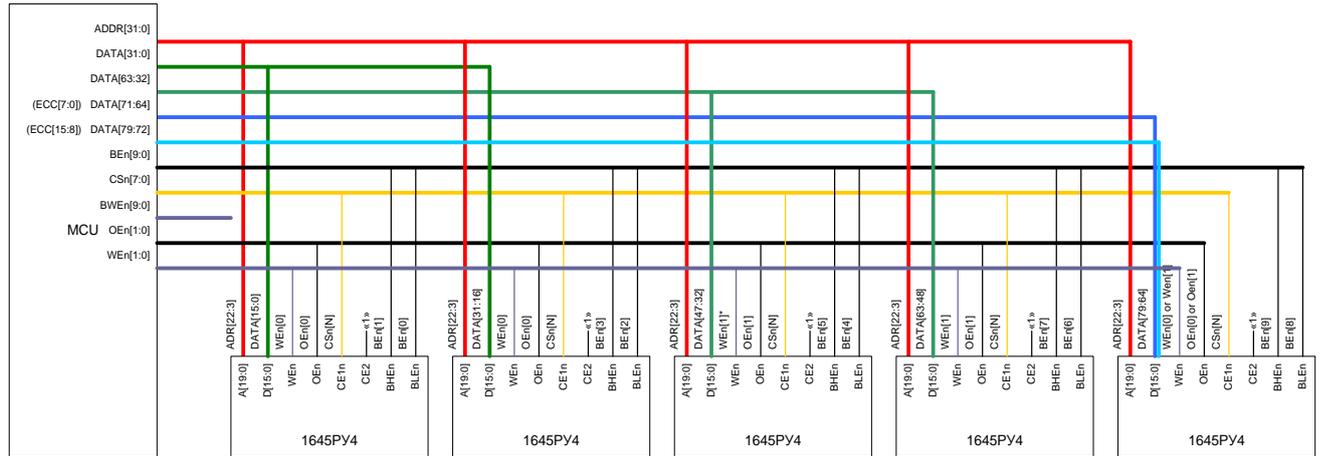


Рисунок 45 – Организация 64-х битной шины с параллельной ECC на базе ОЗУ 1645PY4 (объем пользовательских данных до 8 Мбайт)

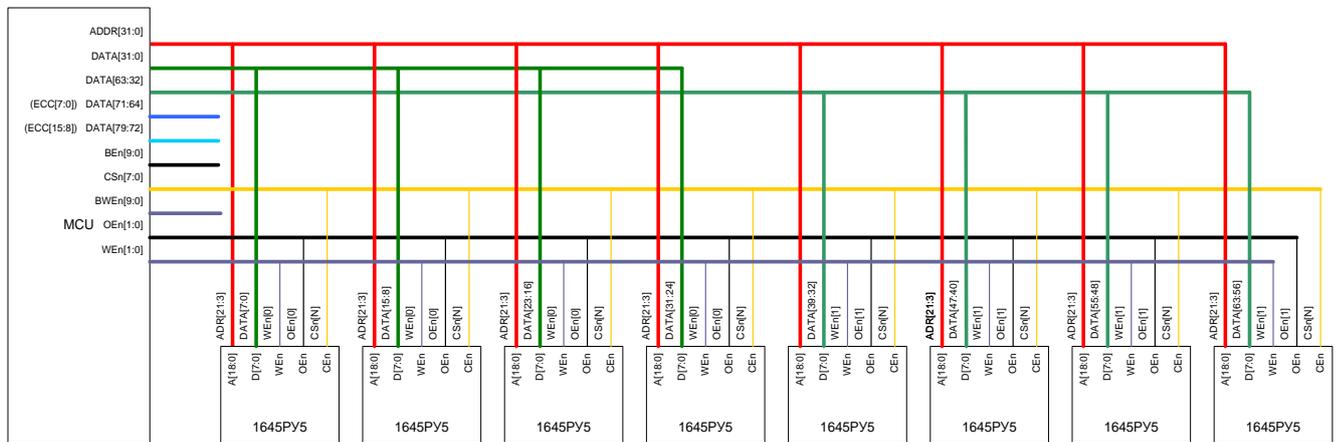


Рисунок 46 – Организация 64-х битной шины с последовательной ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 3,2 Мбайт и 0,8 Мбайт ECC)

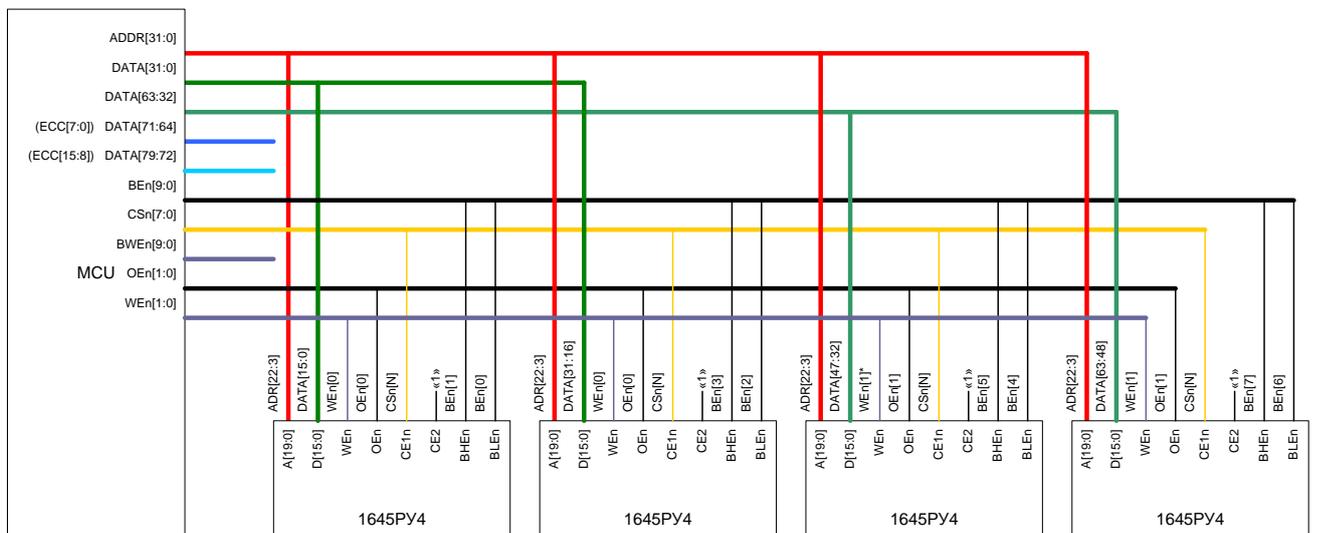


Рисунок 47 – Организация 64-х битной шины с последовательной ECC на базе ОЗУ 1645PY4 (объем пользовательских данных до 6,4 Мбайт и 1,6 Мбайт ECC)

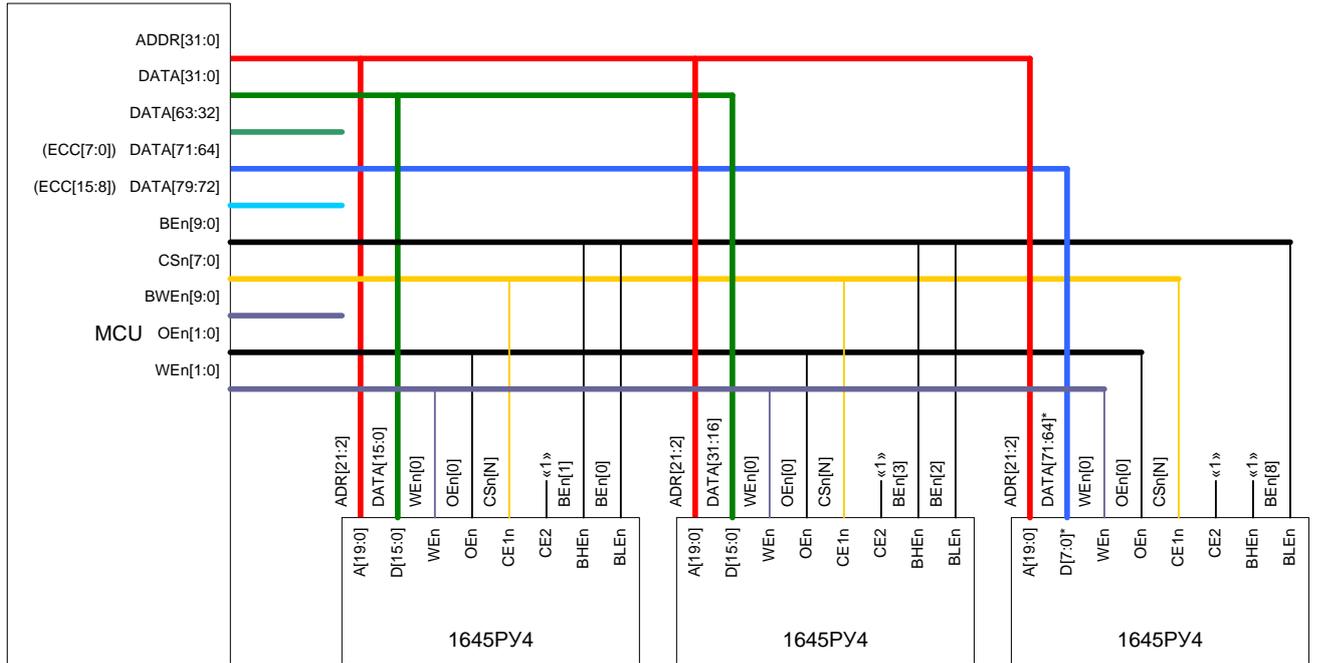


Рисунок 48 – Организация 32-х битной шины с параллельной ECC на базе ОЗУ 1645PY4 (объем пользовательских данных до 4 Мбайт)

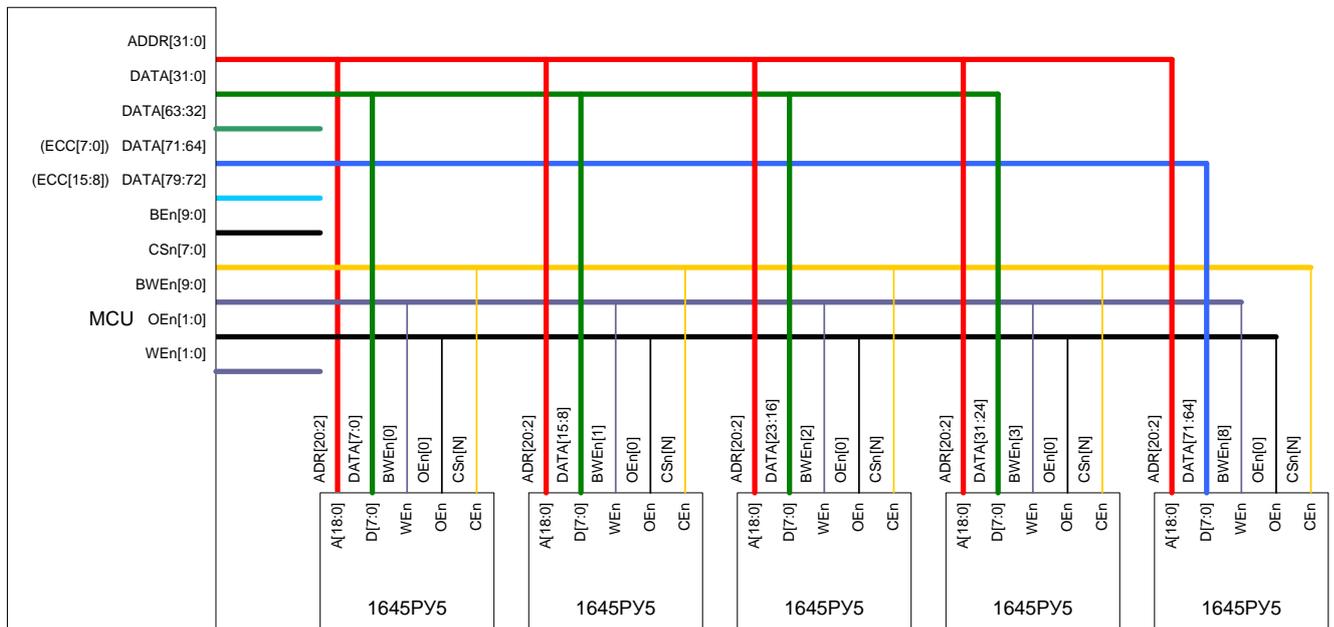


Рисунок 49 – Организация 32-х битной шины с параллельной ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 2 Мбайт)

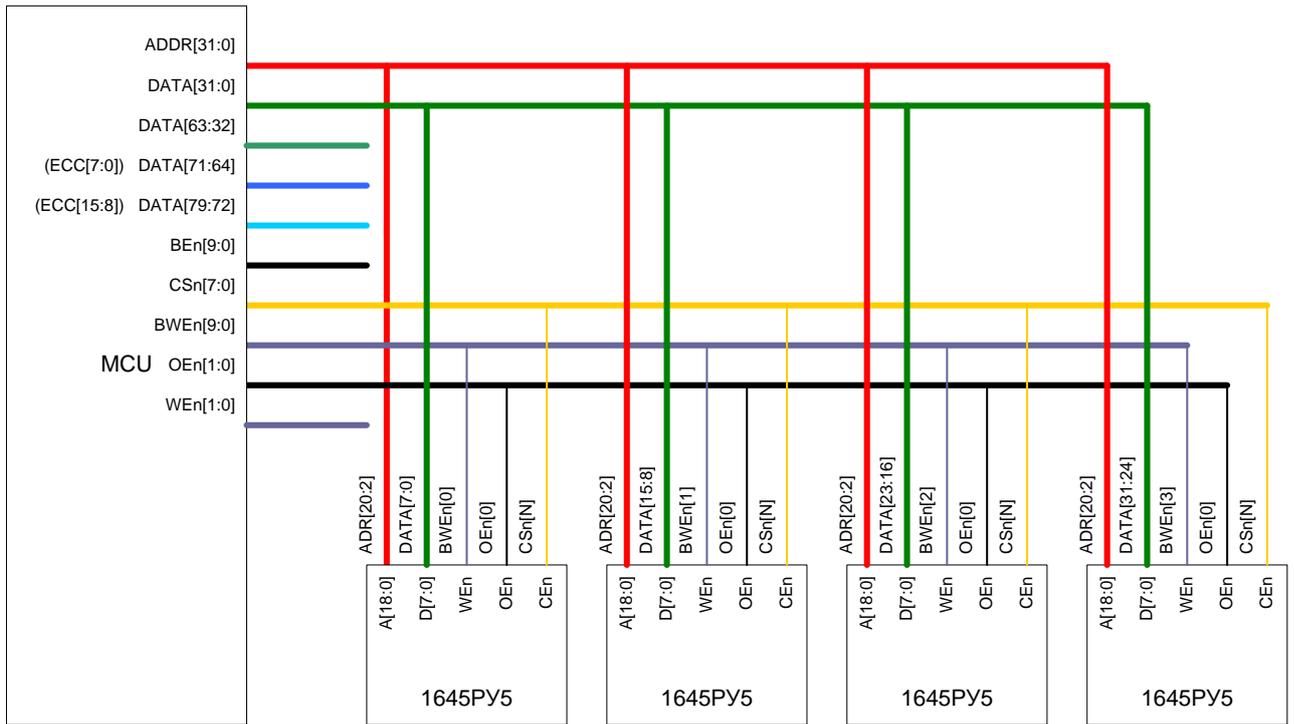


Рисунок 50 – Организация 32-х битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 1,6 Мбайт с последовательной ECC и до 2 Мбайт без ECC)

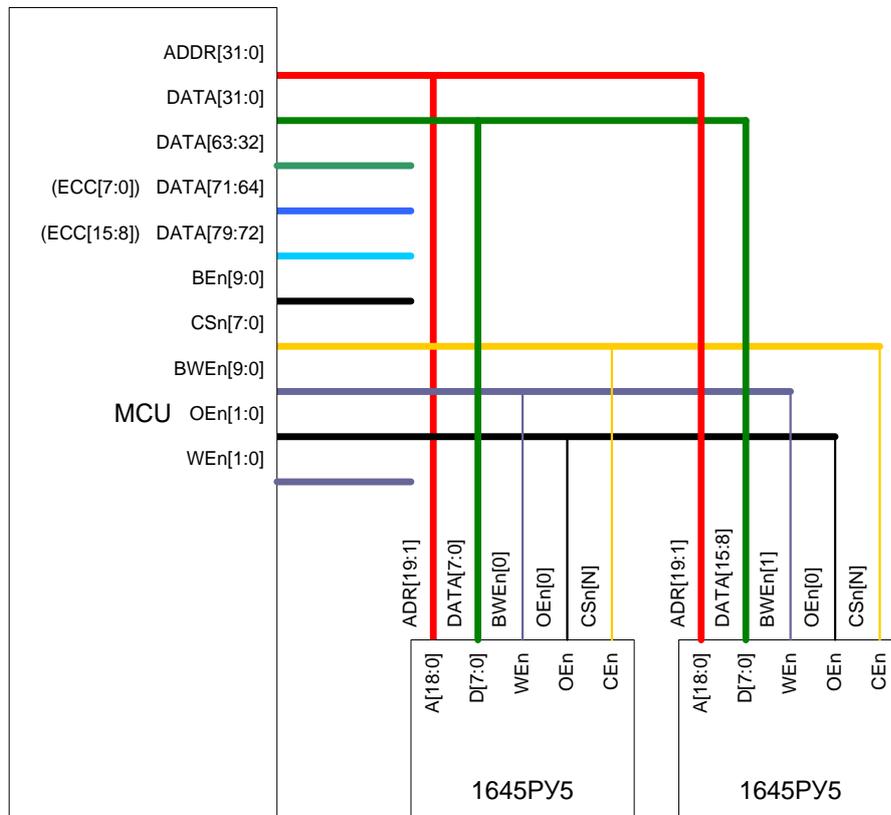


Рисунок 51 – Организация 16-ти битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 0,8 Мбайт с последовательной ECC и до 1 Мбайт без ECC)

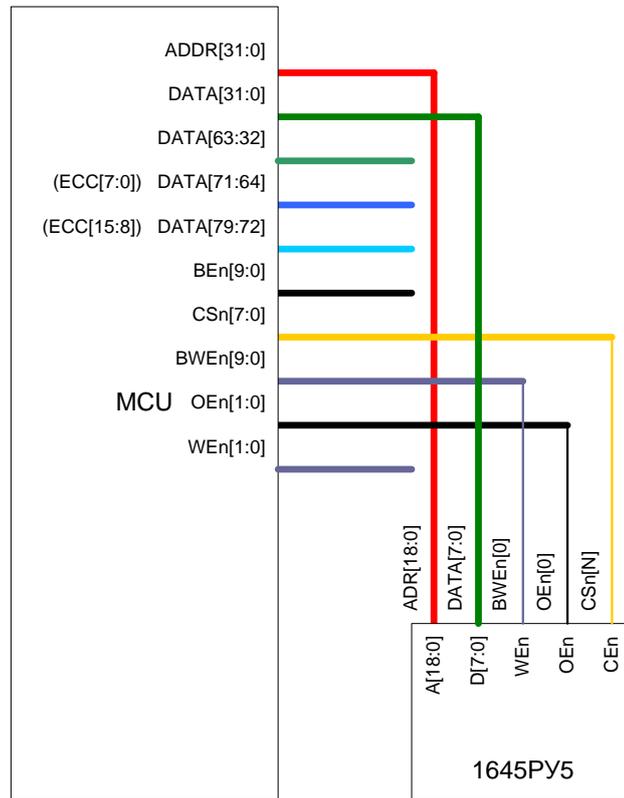


Рисунок 52 – Организация 8-ми битной шины с последовательной организацией ЕС или без ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 0,4 Мбайт с последовательной ECC и до 0,5 Мбайт без ECC)

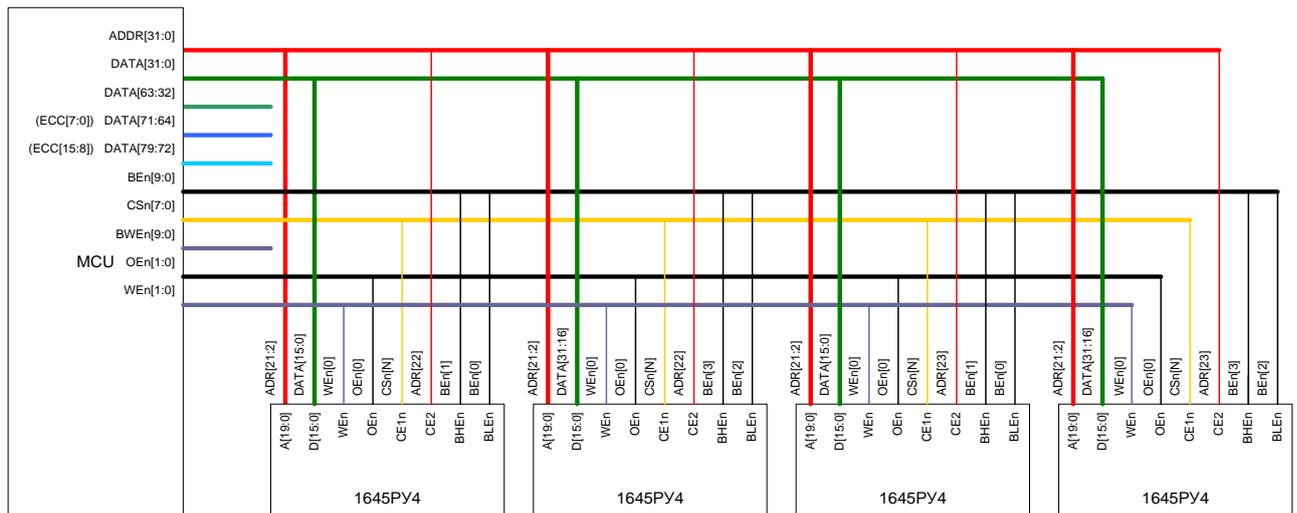


Рисунок 53 – Организация 32-х битной шины с последовательной ECC или без ECC на базе ОЗУ 1645PY4 (объем пользовательских данных до 6,4 Мбайт с ECC или до 8 Мбайт без ECC)

7.7.1.8 Организация внешней системной шины с ожиданием сигнала готовности

Для каждого диапазона внешней системной шины может быть настроен режим работы с ожиданием сигнала готовности READY. В этом случае транзакция на внешней системной шине может быть начата только при пассивном (шина готова) состоянии сигнала READY. При этом будут выполнены фазы предустановки сигналов выборки, адреса/данных перед сигналами

записи/чтения и вход в фазу записи/чтения. Переход из фазы записи/чтения в фазу удержания адреса/данных будет выполнен только при наличии пассивного состояния сигнала READY. При наличии активного состояния сигнала READY (шина не готова) контроллер внешней системной шины будет находиться в фазе записи/чтения. В режиме работы с ожиданием сигнала готовности в любом случае для всех фаз выполняются временные настройки их длительности, сигналом готовности может быть только увеличена фаза записи/чтения. Для каждого диапазона, работающего с ожиданием внешнего сигнала готовности, может быть программно задана максимальная длительность времени ожидания сигнала готовности (перед началом транзакции и нахождения в фазе записи/чтения). Если транзакция успешно не выполняется (без учета корректности ECC) за максимальное время ожидания, то транзакция со стороны процессора завершается с флагом неготовности (исключение BusFault). При работе внешней системной шины в режиме без ожидания сигнала готовности сигнал READY всегда считается находящимся в пассивном состоянии.

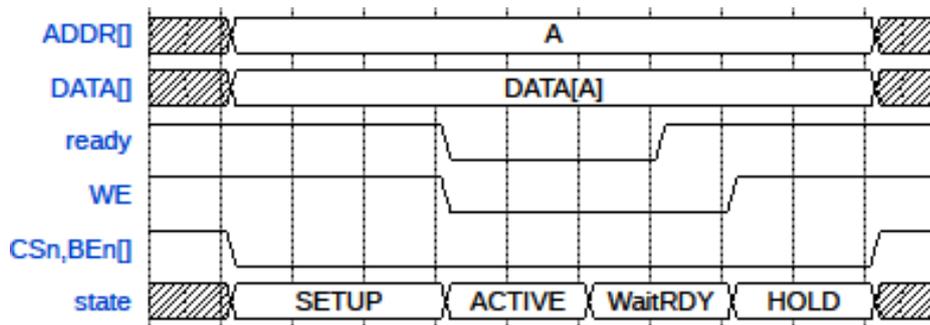


Рисунок 54 – Диаграмма записи

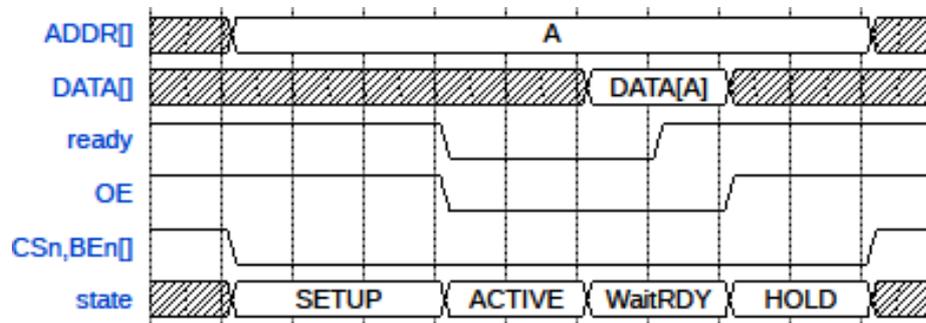


Рисунок 55 – Диаграмма чтения

7.7.2 Описание регистров

Таблица 33 – Описание регистров контроллера внешней шины

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_5000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	REGION[0].CNTRL		Настройки региона 0
	0x0000_0024	REGION[0].ECCBASE		Базовый адрес таблицы ECC региона 0
	0x0000_0044	REGION[0].ECCCS		Регистр статуса ошибок ECC региона 0
	0x0000_0008	REGION[1].CNTRL		Настройки региона 1
	0x0000_0028	REGION[1].ECCBASE		Базовый адрес таблицы ECC региона 1
	0x0000_0048	REGION[1].ECCCS		Регистр статуса ошибок ECC региона 1
	0x0000_000C	REGION[2].CNTRL		Настройки региона 2
	0x0000_002C	REGION[2].ECCBASE		Базовый адрес таблицы ECC региона 2
	0x0000_004C	REGION[2].ECCCS		Регистр статуса ошибок ECC региона 2
	0x0000_0010	REGION[3].CNTRL		Настройки региона 3
	0x0000_0030	REGION[3].ECCBASE		Базовый адрес таблицы ECC региона 3
	0x0000_0050	REGION[3].ECCCS		Регистр статуса ошибок ECC региона 3
	0x0000_0014	REGION[4].CNTRL		Настройки региона 4
	0x0000_0034	REGION[4].ECCBASE		Базовый адрес таблицы ECC региона 4
	0x0000_0054	REGION[4].ECCCS		Регистр статуса ошибок ECC региона 4
	0x0000_0018	REGION[5].CNTRL		Настройки региона 5
	0x0000_0038	REGION[5].ECCBASE		Базовый адрес таблицы ECC региона 5
	0x0000_0058	REGION[5].ECCCS		Регистр статуса ошибок ECC региона 5
	0x0000_001C	REGION[6].CNTRL		Настройки региона 6
	0x0000_003C	REGION[6].ECCBASE		Базовый адрес таблицы ECC региона 6
	0x0000_005C	REGION[6].ECCCS		Регистр статуса ошибок ECC региона 6
	0x0000_0020	REGION[7].CNTRL		Настройки региона 7
	0x0000_0040	REGION[7].ECCBASE		Базовый адрес таблицы ECC региона 7
	0x0000_0060	REGION[7].ECCCS		Регистр статуса ошибок ECC региона 7
	0x0000_0094	ECCADR		Регистр адреса последней ошибки ECC
	0x0000_0098	ECCDATA		Регистр данных последней ошибки ECC
	0x0000_009C	ECCECC		Регистр ECC последней ошибки ECC

7.7.2.1 KEY

Base ADDR=		0x4000_5000				Offset=		0x0000_0000									
REG Name:		KEY															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
KEY[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
KEY[15:0]																

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока EXT_BUS_CNTR

7.7.2.2 REGION[n].CNTRL

Base ADDR=	0x4000_5000	Offset=	0x0000_0004 0x0000_0008 0x0000_000C 0x0000_0010 0x0000_0014 0x0000_0018 0x0000_001C 0x0000_0020												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIVOCLK[3:0]				ENOCLK		CACHEBLE		WS_HOLD[2:0]			WS_SETUP[2:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WS_ACTIVE[7:0]								CPOL	ROM	MODE[1:0]		RDYWAIT	ECCMODE	ECCEN	EN

Бит	Имя	Значение	Описание
31...28	DIVOCLK[3:0]		Биты задания коэффициент деления частоты OCLK FOCLK = HCLK / (2 ^{DIVOCLK+1}) <i>Примечание</i> – Бит реализован только в регистре REGION0
27	ENOCLK		Бит разрешения выдачи синхросигнала OCLK 0 – OCLK не формируется 1 – OCLK формируется Прим: Бит реализован только в регистре REGION0
26	READ32		Разрешение считывания 32-х разрядных слов при 16-ти и 8-ми битной шине MODE = 01 или 10 и в режиме работы без ECC 0 – при чтении 8-ми или 16-ти битных слов читается только 8-ми или 16-ти битные слова. И при возврате процессору ECC не формируется. В этом режиме кэширование не допустимо 1 – при чтении 8-ми или 16-ти битных слов всегда читается только 32-х битное слов. И при возврате процессору ECC формируется корректно.
25	CACHEBLE		Разрешение кэширования внутри контроллера внешней шины при 64-х битной организации MODE=11 0 – кэширование запрещено 1 – кэширование разрешено (при считывании 64-х бит данных, если следующее обращение попадает в выбранное ранее, то транзакции на шине не происходит, и процессору с минимальной задержкой возвращаются запрошенные данные)
24	-		Зарезервировано. Записывать всегда 0.
23...20	WS_HOLD[3:0]		Длительность фазы удержания число тактов HCLK*(WS_HOLD+1)
19...16	WS_SETUP[3:0]		Длительность фазы предустановки число тактов HCLK*(WS_SETUP+1)

Бит	Имя	Значение	Описание
15...8	WS_ACTIVE[7:0]		Длительность активной фазы число тактов $HCLK*(WS_ACTIVE+1)$
7	CPOL		Бит полярности сигнала CLOCK в фазе ACTIVE 0 – передний фронт 1 – задний фронт
6	ROM		Режим работы региона 0 – RAM 1 – ROM При записи в регион при ROM=1 транзакция завершается с флагом ошибка на шине.
5...4	MODE[1:0]		Режим организации шины данных 00 – 32-х разрядная шина данных (+8 разрядов при ECCMODE=0 и ECCEN=1) 01 – 16-ти разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1) 10 – 8-ми разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1) 11 –64-х разрядная шина данных (+16 разрядов при ECCMODE=0 и ECCEN=1)
3	RDYWAIT		Режим работы с ожиданием флага готовности READY 0 – режим работы без ожидания готовности, длительность активной фазы определяется битами WS_ACTIVE 1 – режим работы с ожиданием сигнала готовности, длительность активной фазы не меньше чем задана битами WS_ACTIVE и завершается при наличии высокого уровня на линии READY. Если сигнал READY не сформирован за 256 тактов, транзакция завершается с флагом ошибка на шине.
2	ECCMODE		Режим ECC 0 – параллельная организация ECC (проверочные биты расположены в разрядах шины данных DATA[39:32]) 1 – последовательная организация ECC (проверочные биты расположены в верхних адресах начиная с адреса ECCBASE)
1	ECCEN		Бит разрешения контроля ECC 0 – контроль ECC не выполняется 1 – контроль ECC выполняется
0	EN		Бит разрешения работы региона 0 – регион не работает 1 – регион работает При обращении в адресное пространство не работающего региона транзакция завершается с флагом ошибка на шине.

7.7.2.3 REGION[n].ECCBASE

Base ADDR=	0x4000_5000	Offset=	0x0000_0024												
			0x0000_0028												
			0x0000_002C												
			0x0000_0030												
			0x0000_0034												
			0x0000_0038												
			0x0000_003C												
			0x0000_0040												

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCBASE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCBASE[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCBASE[31:0]		Базовый адрес расположения таблицы ECC в регионе, при ECCMODE=1 и ECCEN=1, иначе не имеет значения

7.7.2.4 REGION[n].ECCCS

Base ADDR=	0x4000_5000	Offset=	0x0000_0044 0x0000_0048 0x0000_004C 0x0000_0050 0x0000_0054 0x0000_0058 0x0000_005C 0x0000_0060												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных и двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика двойных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке 0 – запрещено 1 – разрешено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 0 – запрещено 1 – разрешено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE		Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

Бит	Имя	Значение	Описание
0	SECC		Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.7.2.5 ECCADR

Base ADDR=	0x4000_5000					Offset=	0x0000_0094										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ECCADR[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ECCADR[15:0]																

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.7.2.6 ECCDATA

Base ADDR=		0x4000_5000				Offset=		0x0000_0098									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ECCDATA[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC.

7.7.2.7 ECCECC

Base ADDR=		0x4000_5000				Offset=		0x0000_009C									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
-																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC.

7.8 Контроллер кэш-памяти (CACHENTR)

В микроконтроллере для процессора реализованы кэш инструкций (I-CACHE) размером 1 Кбайт и кэш данных (D-CACHE) размером 128 байт. Кэш-память включается программно. При выключении питания кэш-память обнуляется. С помощью обращения к регистрам кэш-контроллера, кэш-память может быть сброшена. Кэш-память заполняется автоматически по ходу выполнения программы, алгоритм обновления данных в кэш-памяти построен по принципу FIFO в I-Cache и PLRU в D-Cache. Тактирование кэш-памяти осуществляется всегда с частотой ядра HCLK. В кэш-памяти кэшируются данные из OTP и диапазона внешней шины EXT_BUS, обращения к которым осуществляется через шины IBus и DBus. При этом не обеспечивается аппаратная поддержка когерентности закэшированных данных, которые могут быть изменены в ходе работы. Обеспечение когерентности этого региона реализуется пользовательской программой путем перезапуска кэш-памяти. Кэш-память состоит из записей тега адреса и 4-х слов данных с ECC кодами.

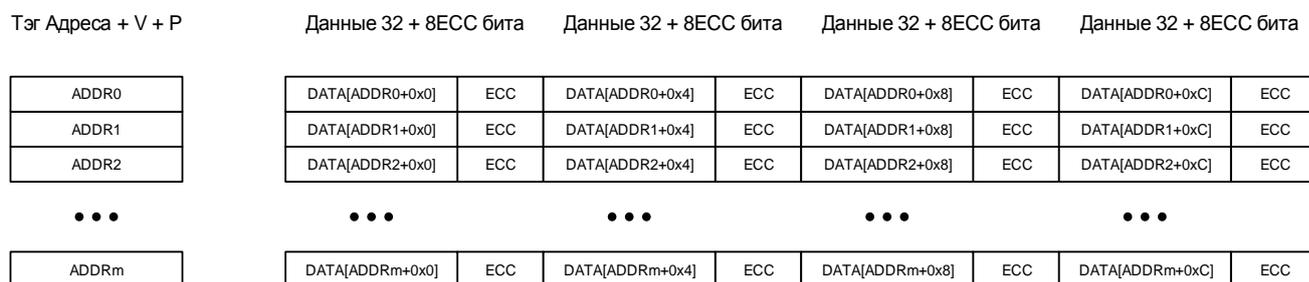


Рисунок 56 – Структура записей в кэш памяти

Для I-CACHE $m = 64$, для D-CACHE $m = 8$.

Значение ECC, сохраняемое в кэш-памяти, позволяет в поле данных исправить одинарную ошибку и обнаружить двойную, а также в поле адреса обнаружить одинарную и двойную ошибку.

При исполнении кода программы из OTP в кэш-память записываются строки целиком, т.е. из OTP-памяти извлекается 160 битное слово (4 x 32 бита данных и 4 x 8 бита ECC), которые по отдельному каналу записываются в кэш-память. Данные будут записаны в I- или D-кэш в зависимости от того, по какой шине (I или D) происходит обращение.

При исполнении кода или чтении данных из внешней шины EXT_BUS через шины IBus и DBus, строки кэш-памяти заполняются последовательно через сами шины IBus и DBus. При этом у необновленных слов в строках кэш-памяти не установлен флаг готовности. И в случае обращения по адресу, совпадающему с активным тегом в кэш, но без установленного флага готовности у необходимого слова, обращение будет пропущено в память OTP или внешнюю шину. Далее при возврате данных они автоматически запишутся в нужную строку кэш-памяти и установят бит готовности. Таким образом, при последующем обращении по этому адресу ответ будет получен уже от кэш-памяти. Управление кэшированием осуществляется через регистр CNTR соответствующего кэш-контроллера. Бит EN_CACHE разрешает кэширование для кэш-контроллера глобально. Биты EN_CACHE_OTP и EN_CACHE_BUS разрешают кэширование соответствующего диапазона памяти. Настройки MPU игнорируются.

7.8.1 Описание регистров

Таблица 34 – Описание регистров контроллера Кэш-памяти

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0xe004_4000				Базовый адрес контроллера кэша шины I-Bus
0xe004_5000				Базовый адрес контроллера кэша шины D-Bus
0x4000_b000				Базовый адрес контроллера кэша шины S-Bus
	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	CNTR		Регистр управления контроллером кэш-памяти
	0x0000_0008	HIT_CNT		Регистр счетчика числа попаданий Кэш
	0x0000_000C	MISS_CNT		Регистр счетчика числа промахов Кэш
	0x0000_0010 ... 0x0000_001F	-		Зарезервировано
	0x0000_0020	ECCCS		Регистр статуса ошибок ECCCACHE
	0x0000_0024	ECCADR		Регистр адреса последней ошибки
	0x0000_0028	ECCDATA		Регистр данных последней ошибки
	0x0000_002C	ECCECC		Регистр контрольной суммы последней ошибки

7.8.1.1 KEY

Base ADDR=	0xe004_4000 0xe004_5000 0x4000_b000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока CACHE_CNTR

7.8.1.2 CNTR

Base ADDR=		0xe004_4000 0xe004_5000 0x4000_b000				Offset=		0x0000_0004							
REG Name:		s													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											CLR_MIS	CLR_HIT	EN_CACHE_B US	EN_CACHE_O TP	CLR_CACHE	EN_CACHE

Бит	Имя	Значение	Описание
31...6			Зарезервировано
5	CLR_MIS		Бит сброса регистра счетчика промахов Сброс выполняется записью 1.
4	CLR_HIT		Бит сброса регистра счетчика попаданий Сброс выполняется записью 1.
3	EN_CACHE_BUS		Бит разрешения кэширования данных от EXT_BUS Имеет значение только при EN_CACHE=1 0 – данные не кэшируются 1 – данные кэшируются Кэширование осуществляется пословно. Для кэша шины S-Bus должен быть 0
2	EN_CACHE_OTP		Бит разрешения кэширования данных от OTP Имеет значение только при EN_CACHE=1 0 – данные не кэшируются 1 – данные кэшируются Кэширование осуществляется строками целиком Для кэша шины S-Bus должен быть 0
1	CLR_CACHE		Бит сброса кэша. Сброс выполняется записью 1.
0	EN_CACHE		Бит общего разрешения работы кэша 0 – выключен 1 – включен

7.8.1.3 HIT_CNT

Base ADDR=		0xe004_4000 0xe004_5000 0x4000_b000				Offset=		0x0000_0008							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Бит	Имя	Значение	Описание
31...0	CNT[31:0]		Счетчик числа попаданий в кэш

7.8.1.4 MISS_CNT

Base ADDR=		0xe004_4000 0xe004_5000 0x4000_b000				Offset=		0x0000_000C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Бит	Имя	Значение	Описание
31...0	CNT[31:0]		Счетчик числа промахов в кэш

7.8.1.5 ECCCS

Base ADDR=	0xe004_4000 0xe004_5000 0x4000_b000	Offset=	0x0000_0020												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке 0 – разрешено 1 – запрещено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 0 – разрешено 1 – запрещено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE		Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.
0	SECC		Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.8.1.6 ECCADR

Base ADDR=		0xe004_4000 0xe004_5000 0x4000_b000				Offset=		0x0000_0024							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.8.1.7 ECCDATA

Base ADDR=		0xe004_4000 0xe004_5000 0x4000_b000				Offset=		0x0000_0028							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC.

7.8.1.8 ECCECC

Base ADDR=		0xe004_4000				Offset=		0x0000_002C								
		0xe004_5000														
		0x4000_b000														
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							
-															

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC.

7.9 Контроллер OTP памяти программ для микросхем 1986VE8Т (OTPCNTR)

Встроенная OTP имеет параллельную организацию ECC кодирования и имеет размерность 4 x 8192 x (32+8ECC) бит. В микроконтроллере реализовано два контроллера OTP всегда работающих в режиме LockStep. Доступ к шинам и памяти имеет только один контроллер, второй используется только для проверки корректности работы первого. В режиме работы LockStep контроль сигналов выполняется на границе между контроллером OTP и непосредственно самим OTP. Конфликты между процессорным ядром и DMA контроллера одного номера разрешаются на уровне арбитража коммутатора АНВ. Данные OTP могут быть кэшированы в кэш-памяти процессора. OTP-память имеет доступ только на чтение. Попытка записи в OTP вызывает исключение busfaulterror. Из OTP извлекаются 4 слова с адресами ADDR+0x0, ADDR+0x4, ADDR+0x8 и ADDR+0xC. Процессору возвращается считанные по адресу ADDR данные с ECC. Контроллер OTP различает обращения по шине I и D и в зависимости от того, по какой шине произошло обращение записывает 4 считанных слова соответственно в I-кэш или D-кэш. После завершения обращения от процессора и отсутствия нового, контроллер начинает упреждающее считывание из адреса ADDR+0x10, ADDR+0x14, ADDR+0x18, ADDR+0x1C. Если в момент считывания приходит обращение в данные адреса, то чтение продолжается до полного выполнения. Если за время упреждающего чтения обращений со стороны процессора не было, то считанные данные фиксируются во внутреннем регистре контроллера OTP, и память OTP переводится в режим пониженного потребления. Если при наличии упреждающе считанных данных приходит обращение по этим адресам, то данные без чтения из OTP выставляются процессору, и начинается новое упреждающее чтение. В зависимости от частоты процессора и времени чтения из OTP-памяти, обращение может занимать от 1 до 8 тактов.

В старших словах OTP-памяти записана уникальная идентификационная информация микросхемы. Данная информация записывается на этапе тестирования и предназначена для идентификации микросхем.

Память OTP имеет топологически распределенную структуру, т.е. биты одного слова (32 данных + 8 ECC) максимально разнесены по площади блока. Это позволяет при возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а максимум получить единичную ошибку, но во многих словах, вместо множественной ошибки в одном слове.

7.9.1 Методика программирования OTP

Однократнопрограммируемая память программ микросхемы 1986VE8Т может быть запрограммирована в двух режимах:

- режим тестирования OTP;
- режим регистрового доступа со стороны процессорного ядра.

В обоих случаях алгоритм программирования одинаков, отличаются только способы формирования сигналов управления при программировании. В режиме тестирования OTP сигналы управления формируются напрямую на выводах микросхемы и отображаются во внутренние сигналы управления блоками памяти OTP. В режиме регистрового доступа внутренние сигналы управления OTP памяти формируются путем записи соответствующих значений в регистры управления блока OPCNTR.

Программирование однократнопрограммируемой памяти выполняется в три этапа, общая блок-схема процесса программирования представлена на рисунке .

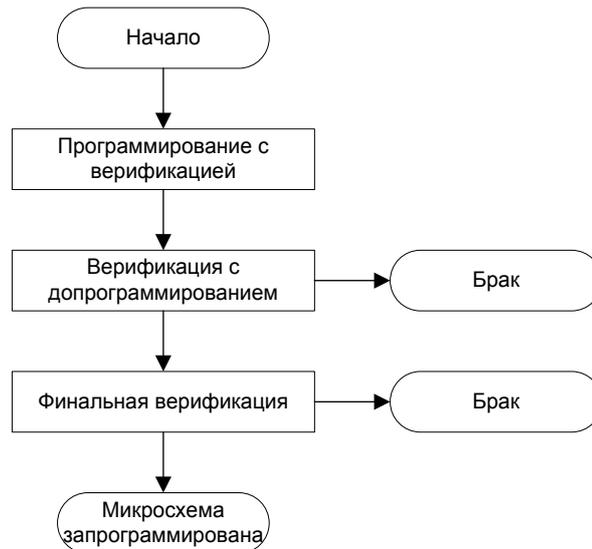


Рисунок 57 – Общая блок-схема процесса программирования

7.9.1.1 Общие требования к процессу программирования OTP

В процессе программирования должны быть выполнены требования, изложенные ниже.

При программировании памяти программ в режиме регистрового доступа невозможно исполнение программного кода из OTP памяти. Исполняемый код в этом случае можно расположить, например, в ОЗУ.

Программирование OTP памяти должно проводиться в лабораторных условиях при температуре от 15 до 45 °С.

Для микросхем 1986VE8Т гарантируется программируемость с коэффициентом программируемости N_{PR} , что необходимо учитывать при выборе метода программирования и монтажа микросхем.

Минимальной программируемой ячейкой памяти является одно 40-битное слово (32 информационных бита и 8 бит корректирующего кода ECC).

При программировании информационные биты и корректирующие биты ECC программируются одинаковым методом. Расположение информационных битов и битов ECC в шине данных блока памяти при программировании в тестовом режиме OTP следующее:

- Биты ECC[7:0] соответствуют битам шины данных для записи d[7:0] и q[7:0] шины данных для чтения.
- Информационные биты DATA[31:0] соответствуют битам шины данных для записи d[39:8] и q[39:8] шины данных для чтения.
- В режиме регистрового доступа биты ECC[7:0] при записи необходимо устанавливать в поле WECC регистра CNTR блока OTPCNTR.
- В режиме регистрового доступа информационные биты DATA[31:0] при записи необходимо устанавливать в поле DATA регистра WDATA блока OTPCNTR.

Значение 8-ми бит корректирующего кода вычисляется на основании значения информационных бит и адреса их расположения в соответствии с подразделом 6.6.5 «Функция вычисления ECC для записи в память».

При программировании памяти программ возможно последующее добавление записи в ранее не запрограммированную область. При этом финальная верификация должна выполняться для всех областей, в том числе для ранее запрограммированных.

При программировании OTP памяти программ должны быть запрограммированы также не менее 8-ми последующих 40-битных слов памяти за последним словом программируемого массива. Это необходимо для исключения возможного возникновения ошибок при упреждающем чтении памяти при выборке инструкций в последнем слове программируемого массива.

7.9.1.2 Этап «Программирование с верификацией»

Этап «Программирование с верификацией» предназначен для программирования основного массива памяти. Алгоритм этапа представлен на рисунке .

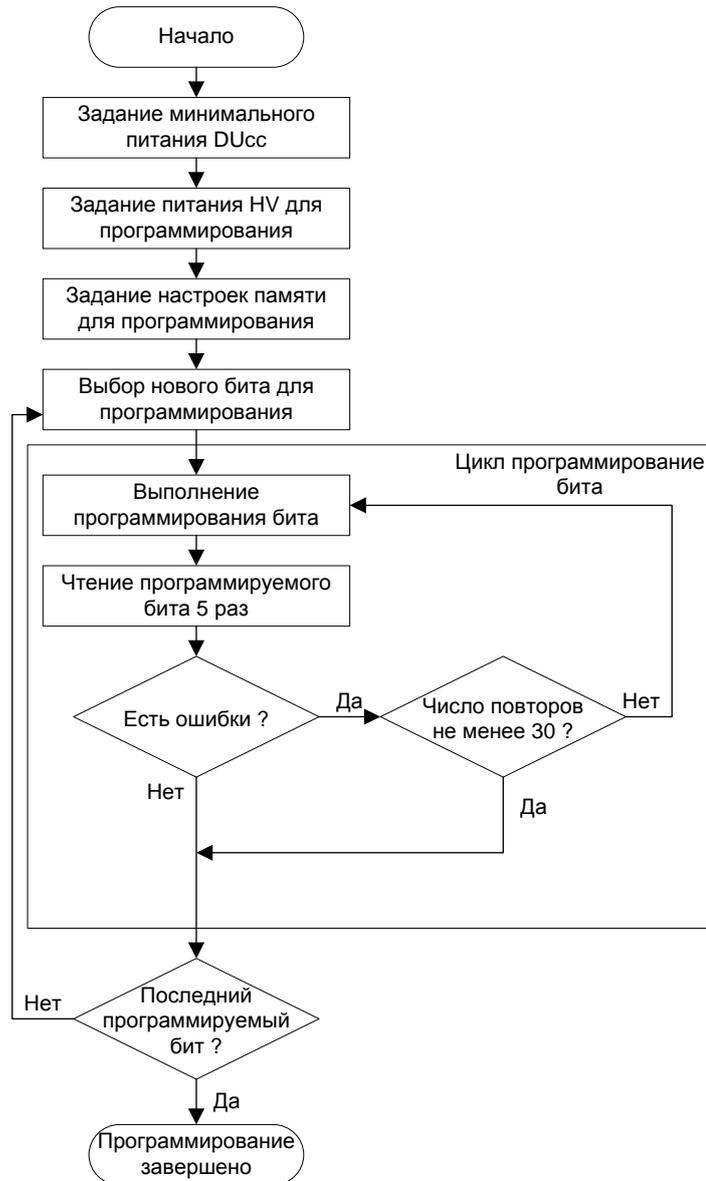


Рисунок 58 – Алгоритм «Программирование с верификацией»

Подэтап «Задание минимального питания DUcc» предназначен для формирования пониженного уровня напряжения DUcc (типовое значение 1,8 В). Снижение напряжения питания может быть осуществлено одним из двух способов:

- перестройкой встроенных регуляторов;
- заданием внешнего питания при отключении встроенных регуляторов.

Перестройка встроенных регуляторов напряжения осуществляется записью значения 3'b0111 в поля TRIM0, TRIM1, TRIM2 и TRIM3 регистров REG_61_TMR0,1,2 батарейного домена ВКРСNTR.

Задание внешнего питания при отключении встроенных регуляторов осуществляется путем подачи на выводы микросхемы DUcc0,1,2 и DUcc_V и DUcc_PLL от внешнего источника напряжения питания 1,62 В с последующей подачей на вывод SHDN высокого уровня.

Подэтап «Задание питания HV для программирования» предназначен для формирования высокого уровня напряжения на выводе HV для программирования. Высокое напряжения Uprog на выводе HV должно быть сформировано внешним источником питания со

значением от 7,0 до 7,4 В. Типовое значение 7,2 В. Схема подачи высокого напряжения программирования на выводе HV должна быть реализована таким образом, чтобы при отсутствии высокого напряжения на выводе формировался уровень напряжения от 3,0 до 5,5 В, но не более Uсс.

Подэтап «Задание настроек памяти для программирования» предназначен для перевода памяти в режим, при котором «усложнено» чтение запрограммированных бит, что позволяет более качественно проверить успешность их программирования. Для задания настроек необходимо сформировать на входах repl_tune[3:0] и time_tune[2:0] блоков памяти OTP значения в соответствии с таблицей .

Таблица 35 – Значения на входах repl_tune[3:0] и time_tune[2:0]

Настройка	При программировании	При допрограммировании	При финальной верификации	При чтении (типичное значение)
repl_tune[3:0]	4'b 1111	4'b 1111	4'b 1000	4'b 0010
time_tune[2:0]	3'b 0010	3'b 0010	3'b 0010	3'b 0011

В тестовом режиме OTP данные значения (см. таблицу) задаются на выходы PC[31:28] и PA[31:29], в режиме регистрового доступа данные задаются в регистре TEST_TUNING блока OTPCNTR.

Подэтап «Выбор нового бита программирования» предназначен для задания очередного программируемого из 0 в 1 бита. Незапрограммированные биты памяти читаются как 0. Программируются только биты, для которых необходимо установить значение 1. Программирование осуществляется побитно. То есть в каждый момент времени может программироваться только один бит. Для ранее запрограммированных битов необходимо устанавливать нулевое значение при программировании других битов в одном слове.

Подэтап «Выполнение программирования бита» предназначен для формирования диаграммы сигналов управления для программирования одного бита. Диаграмма программирования представлена на рисунке .

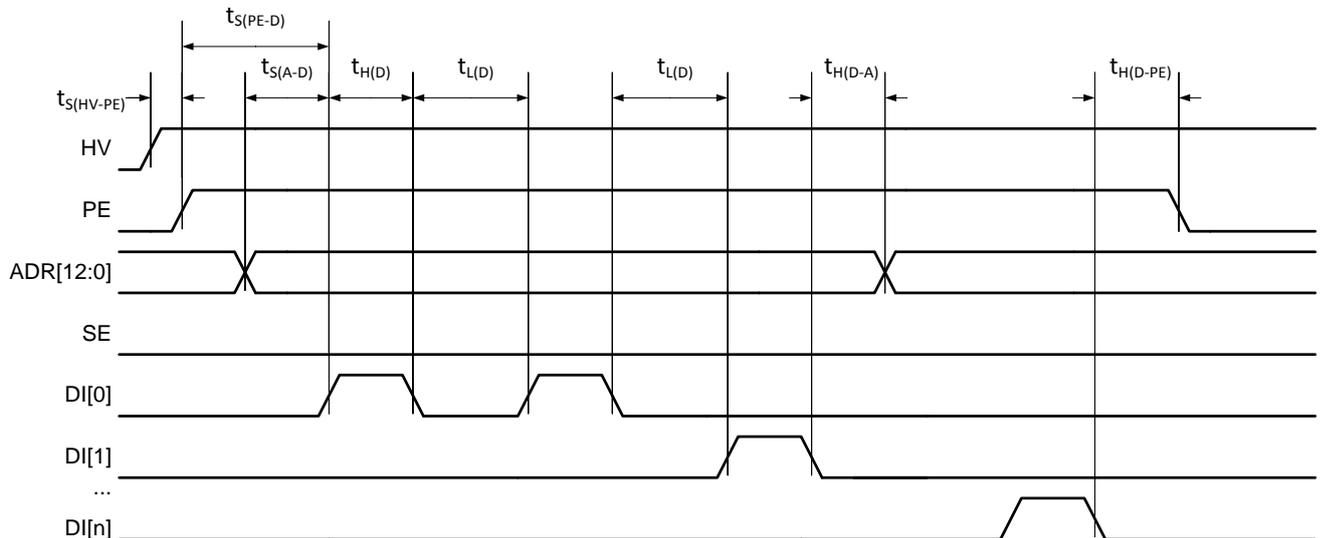


Рисунок 59 – Временная диаграмма процесса программирования OTP памяти

Подэтап «Чтение программируемого бита 5 раз» предназначен для определения результата выполнения программирования бита. Диаграмма чтения представлена на рисунке .

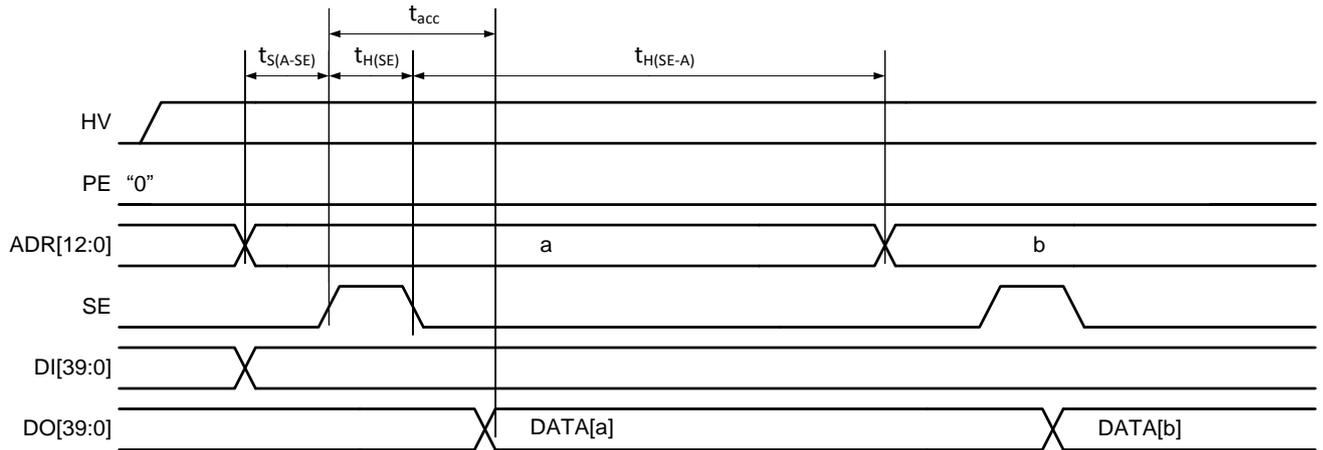


Рисунок 60 – Диаграмма чтения

Чтение должно быть выполнено не менее пяти раз. Если при этом хотя бы один раз запрограммированный бит считался с ошибкой (контролироваться должен только программируемый бит), бит считается не запрограммированным. Необходимо провести повторное программирование бита. При превышении 30 раз числа повторов программирования одного бита можно перейти к программированию следующего бита, не смотря на результат программирования данного бита.

Этап считается выполненным успешно после выполнения всех подэтапов.

7.9.1.3 Этап «Верификация и допрограммирование»

Этап «Верификация и допрограммирование» предназначен для проверки запрограммированной информации и допрограммирования «проблемных» битов. При проверке контролируются все биты, в том числе нулевые. Алгоритм процесса представлен на рисунке .

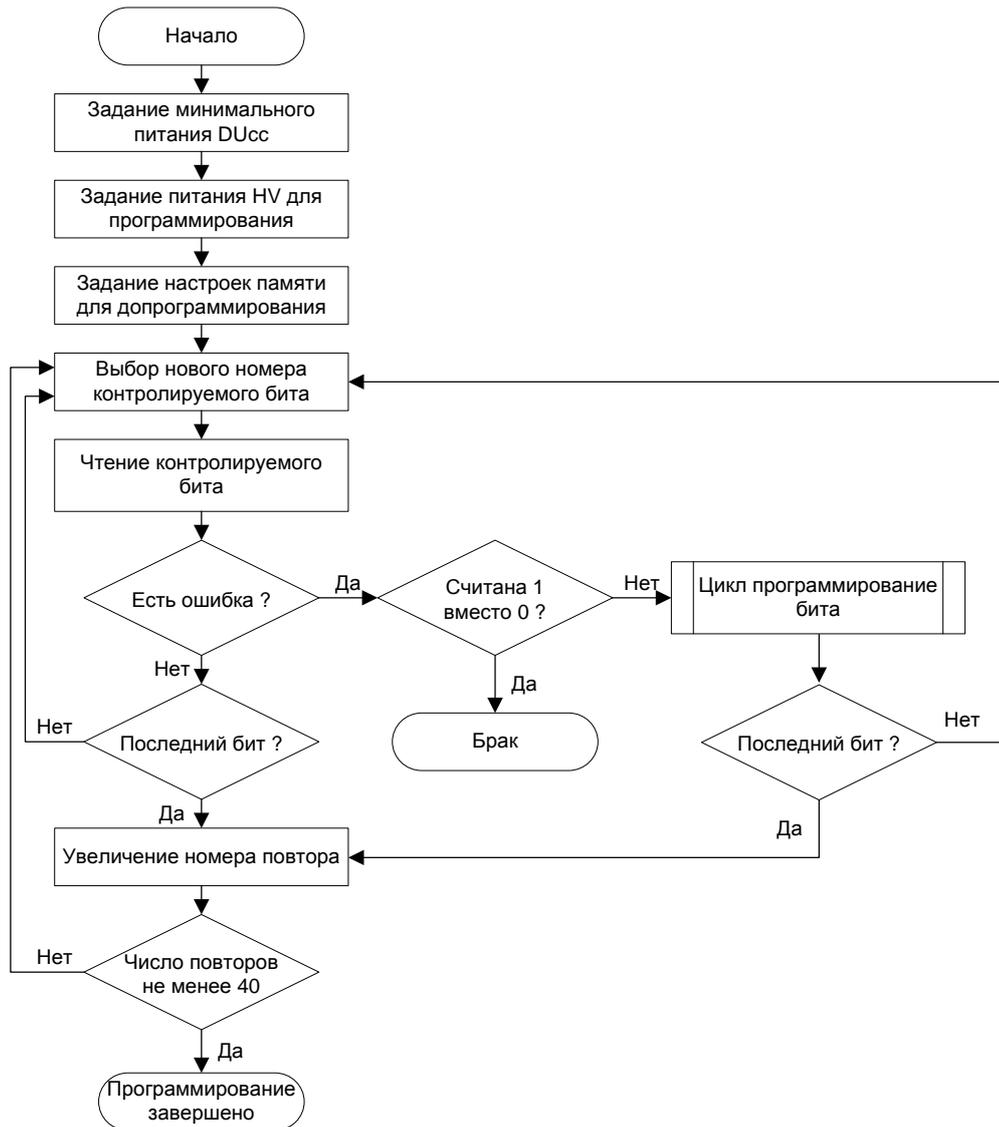


Рисунок 61 – Алгоритм этапа «Верификация и допрограммирование»

Подэтапы «Задание минимального напряжения питания DUcc» и «Задание питания HV для программирования» аналогичны соответствующим подэтапам, рассмотренным в подразделе 7.9.1.2 «Этап «Программирование с верификацией»».

Подэтап «Задание настроек памяти для допрограммирования» предназначен для перевода памяти в режим, при котором «усложнено» чтение запрограммированных бит, что позволяет более качественно проверить успешность их программирования. Для задания настроек необходимо сформировать на входах `per1_tune[3:0]` и `time_tune[2:0]` блоков памяти OTP значения в соответствии с таблицей.

Подэтап «Выбор нового номера контролируемого бита» предназначен для выбора очередного контролируемого бита ранее запрограммированной информации.

Подэтап «Чтение контролируемого бита» предназначен для чтения ранее запрограммированной информации, содержащей выбранный контролируемый бит. Для контроля битов одной ячейки памяти допускается выполнение однократного чтения ячейки. Если в результате чтения возникла ошибка, и вместо ожидаемого «0» считана «1», процесс программирования бракуется. Если вместо ожидаемой «1» считан «0», для данного бита должен быть выполнен цикл программирования бита, аналогичный циклу, рассмотренному в подразделе 7.9.1.2 «Этап «Программирование с верификацией»».

Подэтап «Увеличение номера повтора» предназначен для увеличения номера повтора цикла контроля всей запрограммированной информации.

Этап считается выполненным успешно, если число повторов превысило значение 40.

7.9.1.4 Этап «Финальная верификация»

Этап «Финальная верификация» предназначен для проверки запрограммированной информации с учетом времени выборки из памяти. При чтении данных из ОТР памяти кроме самой информации должно контролироваться время выборки данных.

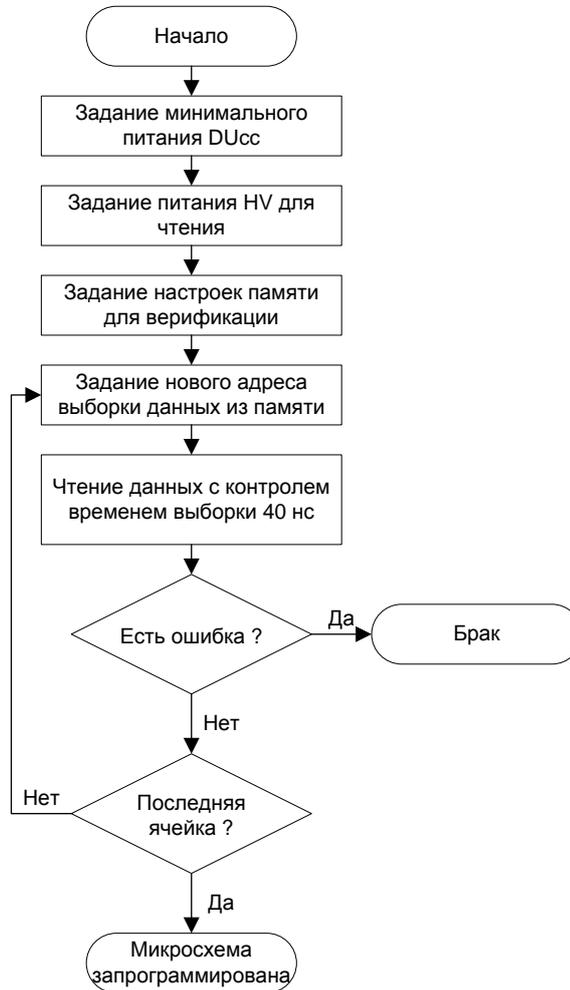


Рисунок 62 – Алгоритм этапа «Финальная верификация»

Подэтап «Задание минимального питания DUcc» предназначен для задания пониженного уровня питания DUcc. В случае использования встроенных регуляторов напряжения должна быть выполнена перестройка встроенных регуляторов напряжения, которая осуществляется записью значения 3'b0111 в поля TRIM0, TRIM1, TRIM2 и TRIM3 регистров REG_61_TMR0,1,2 батарейного домена WKPCNTR.

Подэтап «Задание питания HV для чтения» предназначен для задания на выводе HV микросхемы напряжения в диапазоне от 2,4 до 4,9 В, но не более Ucc.

Подэтап «Задание настроек памяти для верификации» предназначен для перевода памяти в режим, при котором обеспечивается «усложненное» чтение запрограммированных бит, по сравнению с чтением при реальной работе. Для задания настроек необходимо сформировать на входах `per_tune[3:0]` и `time_tune[2:0]` блоков памяти ОТР значения в соответствии с таблицей.

Подэтап «Задание нового адреса выборки данных из памяти» предназначен для выбора очередного контролируемого слова в памяти.

Подэтап «Чтение данных с контролем времени выборки 40 нс» предназначен для чтения данных из памяти с контролем времени выборки. Данный режим чтения обеспечивается в тестовом режиме ОТР путем задания момента стробирования считанных данных с шины данных в 55 нс от момента выдачи высокого уровня SE (55 нс включают время выборки 40 нс из ОТР и задержку на выводах микросхемы). Либо путем чтения запрограммированной памяти процессорным ядром при 25 МГц (40 нс) с последующим контролем информации. При этом

число тактов ожидания WAITCYCLE должно быть установлено в 1. В тестовом режиме OTP памяти возможно контролировать все биты информации, включая ECC биты. В режиме чтения памяти со стороны процессорного ядра будет выполняться автоматическое исправления одиночных и обнаружение двойных ошибок. Таким образом, считанная информация считается верной, если информационные биты совпадают с требуемым значением, и при чтении контроллером OTP не было обнаружено одиночных и двойных ошибок в результате контроля данных по ECC.

Микросхема считается успешно запрограммированной, если вся ранее запрограммированная информация считана без ошибок.

При исполнении программного кода из OTP памяти программ должны быть выставлены типовые настройки уровня питания DUcc и repl_tune[3:0] и time_tune[2:0].

Таблица 36 – Параметры при программировании OTP

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Значение параметра	
		не менее	не более
Напряжение при программировании HV, В	U _{HVROG}	7,0	7,4
Напряжение при хранении и чтении HV, В	U _{HVREAD}	2,4	4,9 и U _{cc}
Время предустановки HV перед PE, мс	t _{S(HV-PE)}	10	-
Время предустановки PE перед данными, мкс	t _{S(PE-D)}	300	-
Время предустановки адреса перед данными, мкс	t _{S(A-D)}	300	-
Время удержания адреса после данных, мкс	t _{H(D-A)}	5	-
Длительность программирования бита данных, мс	t _{H(D)}	3	7
Время между программируемыми битами данных, мкс	t _{L(D)}	5	-
Время удержания PE после данных, мкс	t _{H(D-PE)}	0	-
Время предустановки адреса перед SE, нс	t _{S(A-SE)}	5	-
Время удержания адреса после SE, нс	t _{H(SE-A)}	0	-
Длительность сигнала SE, нс	t _{H(SE)}	10	-

7.9.2 Параметры контроллера OTP

Контроллер OTP может работать в двух режимах:

- нормальный режим;
- режим регистрового доступа.

Выбор режима осуществляется с помощью бита REG_ACCESS регистра CNTR.

Для корректной работы контроллера в нормальном режиме необходимо установить параметр WAITCYCL регистра CNTR в значение, соответствующее частоте работы CPU_CLK. Изменения данного параметра необходимо производить только в отсутствие обращений к OTP со стороны контроллера. Т.е. если код выполняется из OTP, процедура установки WAITCYCL должна быть скопирована в ОЗУ и выполнена оттуда. Диаграмма работы контроллера в нормальном режиме представлена на рисунке .

Перевод контроллера в регистровый режим позволяет управлять входными сигналами OTP по отдельности и считывать состояние выходных сигналов блоков OTP.

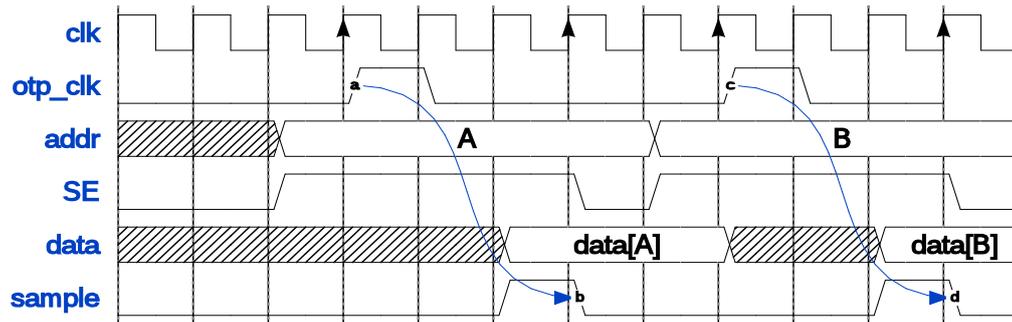


Рисунок 63 – Диаграмма работы контроллера OTP с WAITCYCL=3

7.9.3 Описание регистров

Таблица 37 – Описание регистров контроллера OTP-памяти

Базовый адрес	Смещение	Название	Описание
0x4000_6000	0x0000_0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	CNTR	Регистр управления OTP памяти
	0x0000_0008	ADR	Регистр адреса при программировании
	0x0000_000C	WDATA	Регистр данных при программировании
	0x0000_0010	RDATA	Регистр данных при регистровом считывании
	0x0000_0014	ECCCS	Регистр статуса ошибок ECCOTP
	0x0000_0018	ECCADR	Регистр адреса последней ошибки
	0x0000_001C	ECCDATA	Регистр данных последней ошибки
	0x0000_0020	ECCECC	Регистр данных ECC последней ошибки
	0x0000_0024	TEST_TUNING	Регистр подстройки параметров блока

7.9.3.1 KEY

Base ADDR=	0x4000_6000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока OTPCNTR

7.9.3.3 ADR

Base ADDR=		0x4000_6000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ADR[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ADR[31:0]		Адрес записываемого слова

7.9.3.4 WDATA

Base ADDR=		0x4000_6000				Offset=		0x0000_000C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Бит	Имя	Значение	Описание
31...0	DATA[31:0]		Данные для записи. При записи в память к ним добавляется поле WECC из регистра CNTR.

7.9.3.5 RDATA

Base ADDR=		0x4000_6000				Offset=		0x0000_0010								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Бит	Имя	Значение	Описание
31...0	DATA[31:0]		Данные при регистровом чтении. Проверка ЕСС для данных не выполняется, поле ЕСС считанных данных отображено в битах RECC регистра CNTR.

7.9.3.6 ECCCS

BaseADDR=	0x4000_6000	Offset=	0x0000_0014												
REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных последней ошибки при двойной ошибке 0 – запрещено 1 – разрешено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных последней ошибки при одинарной ошибке 0 – запрещено 1 – разрешено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE		Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.
0	SECC		Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.9.3.7 ECCADR

Base ADDR=		0x4000_6000				Offset=		0x0000_0018								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCADR[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.9.3.8 ECCDATA

Base ADDR=		0x4000_6000				Offset=		0x0000_001C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCDATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC.

7.9.3.9 ECCECC

Base ADDR=		0x4000_6000				Offset=		0x0000_0020								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC.

7.9.3.10 TEST_TUNING

Base ADDR=		0x4000_6000				Offset=		0x0000_0024								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
												REPL_TUNING[3:0]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TIME_TUNING[2:0]							TE[1:0]		TUNE[2:0]	

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19:16	REPL_TUNING[3:0]	4'b0010	Регистр подстройки ОТП 0000 – Большой запас по чтению “1” и меньший по чтению “0” 0001 0010 – Типовое значение ... 1100 1101 1110 1111 – Меньший запас по чтению “1” и больший по чтению “0”
15...11	-		Зарезервировано
10:8	TIME_TUNING[2:0]	3'b011	Регистр подстройки ОТП 000 – Меньше время выборки, меньше общий запас по чтению и “0” и “1” 001 010 011 – Типовое значение 100 101 110 111 – Больше время выборки, больше общий запас по чтению и “0” и “1”
7...5	-		Зарезервировано
4...3	TE[1:0]	2'b00	Биты выбора тестового режима при отбраковочном тестировании
2...0	-		Зарезервировано

7.10 Контроллер ОЗУ памяти программ для микросхем 1986BE81Т (ОТРСNTR)

В микросхеме 1986BE81Т память программ реализована в виде ОЗУ данных. При этом оперирование с ней максимально приближено к работе ОTR памяти программ микросхемы 1986BE8Т. При чтении данных из ОЗУ памяти программ со стороны процессора обеспечиваются те же временные параметры, что и при чтении ОTR памяти. При записи в ОЗУ реализуется аналогичный метод регистрового доступа, но отличаются управляющие сигналы. Запись данных в память осуществляется одновременно для всех 40-ка бит.

Временная диаграмма чтения памяти программ типа RAM приведена на рисунке .

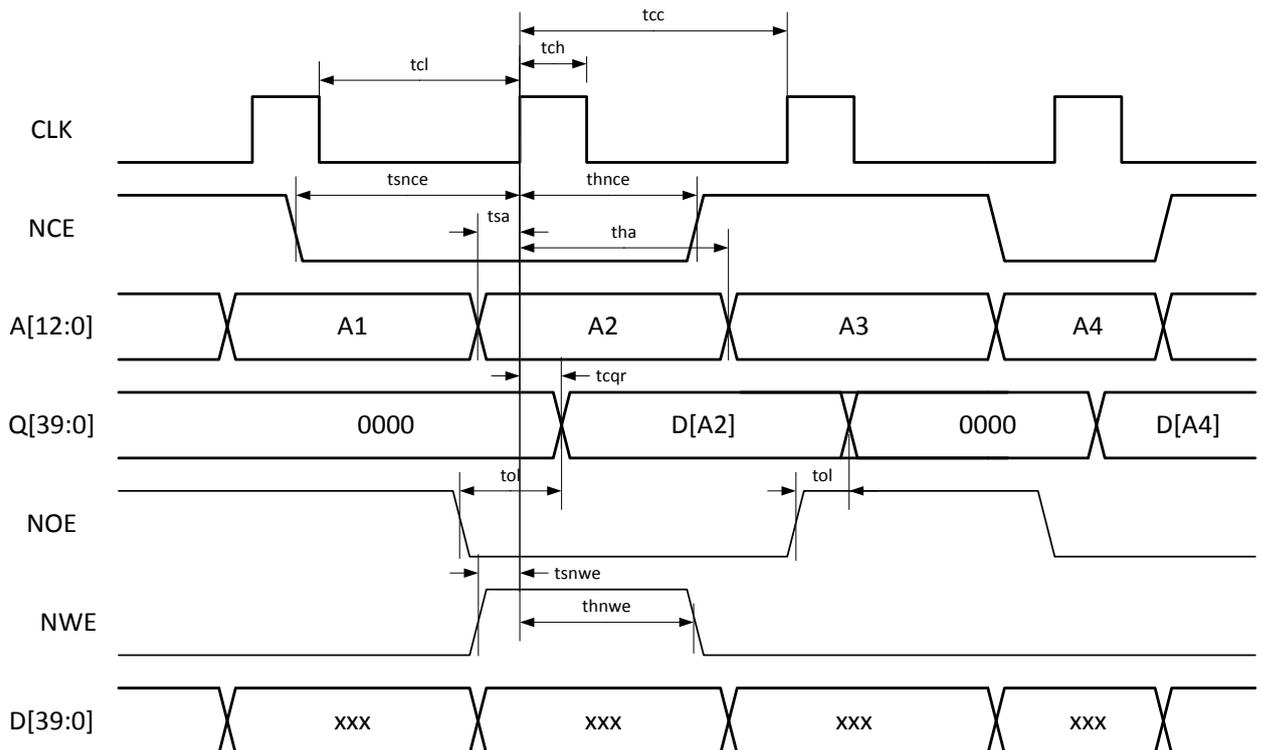


Рисунок 64 – Временная диаграмма чтения памяти программ типа RAM

Временная диаграмма записи памяти программ типа RAM приведена на рисунке .

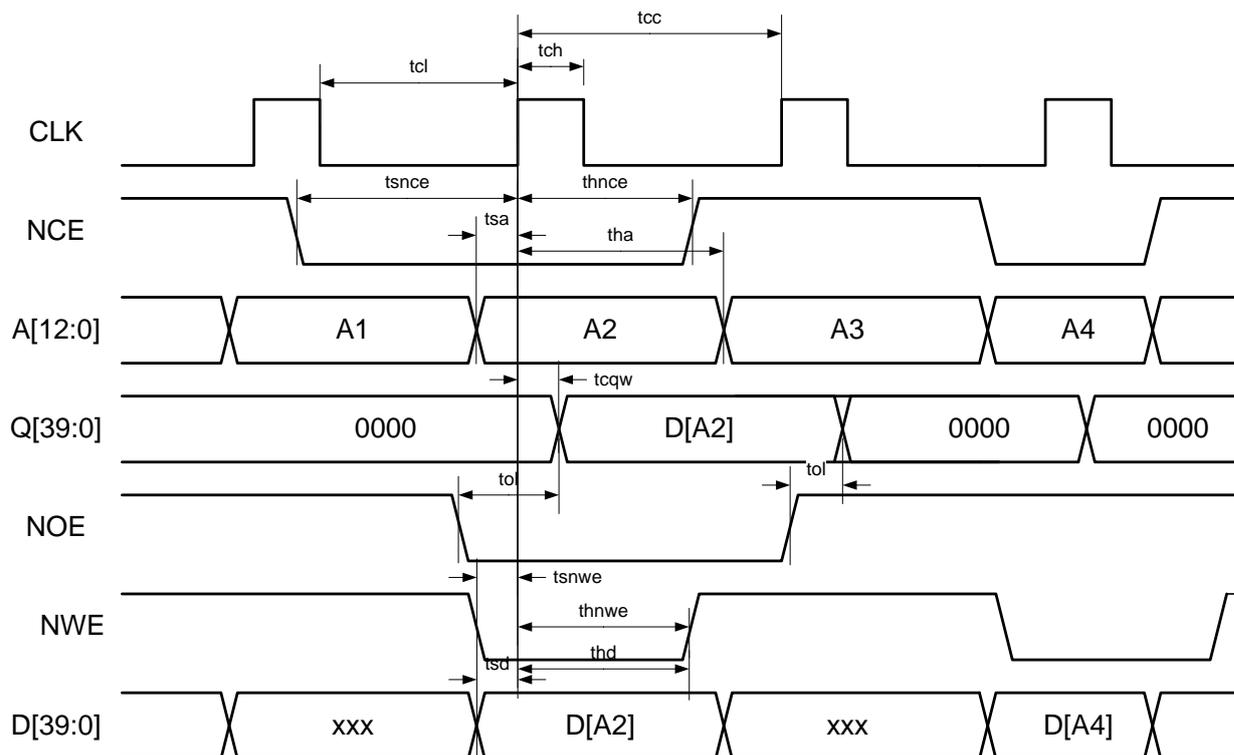


Рисунок 65 – Временная диаграмма записи памяти программ типа RAM

Таблица 38 – Параметры временной диаграммы записи памяти программ типа RAM

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Значение параметра	
		не менее	не более
Период тактового сигнала clk, нс	t_{cc}	10	-
Длительность высокого уровня импульса clk, нс	t_{ch}	5	-
Длительность низкого уровня импульса clk, нс	t_{cl}	5	-
Время установления сигнала адреса относительно переднего фронта сигнала clk, нс	t_{sa}	5	-
Время удержания сигнала адреса относительно переднего фронта сигнала clk, нс	t_{ha}	5	-
Время установления сигнала pce относительно переднего фронта сигнала clk, нс	t_{snce}	5	-
Время удержания сигнала pce относительно переднего фронта сигнала clk, нс	t_{hnce}	5	-
Время установления сигнала разрешения записи pwe относительно переднего фронта сигнала clk, нс	t_{snwe}	5	-
Время удержания сигнала разрешения записи pwe относительно переднего фронта сигнала clk, нс	t_{hnwe}	5	-
Время установления сигнала данных d относительно переднего фронта сигнала clk, нс	t_{sd}	5	-
Время удержания сигнала данных d относительно переднего фронта сигнала clk, нс	t_{hd}	5	-
Время установки и удержания сигнала NOE, нс	t_{ol}	5	-

7.10.1 Описание регистров

Таблица 39 – Описание регистров контроллера OTP-памяти

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_6000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	CNTR		Регистр управления RAM памяти
	0x0000_0008	ADR		Регистр адреса при программировании
	0x0000_000C	WDATA		Регистр данных при программировании
	0x0000_0010	RDATA		Регистр данных при регистровом считывании
	0x0000_0014	ECCCS		Регистр статуса ошибок ECC
	0x0000_0018	ECCADR		Регистр адреса последней ошибки
	0x0000_001C	ECCDATA		Регистр данных последней ошибки
	0x0000_0020	ECCECC		Регистр данных ECC последней ошибки
	0x0000_0024	TEST_TUNING		Регистр подстройки параметров блока

7.10.1.1 KEY

Base ADDR=	0x4000_6000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока OTPCNTR

7.10.1.2 CNTR

Base ADDR=		0x4000_6000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RECC[7:0]								WECC[7:0]								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	CEN	WE[4:0]							OE		CLK	REG_ACCESS	WAITCYCL[3:0]			

Бит	Имя	Значение	Описание
31...24	RECC[7:0]	0x00	Биты ECC при регистровом чтении памяти
23...16	WECC[7:0]	0x00	Биты ECC при программировании WDATA
15	-		Зарезервировано
14	CEN	0	Сигнал CEN для RAM 0 – память выбрана 1 – память не выбрана
13...9	WE[4:0]	0x1F	Сигнал WEN[4:0] для RAM WE[0] для первого блока ADDR[3:0] = 0x0 WE[1] для первого блока ADDR[3:0] = 0x4 WE[2] для первого блока ADDR[3:0] = 0x8 WE[3] для первого блока ADDR[3:0] = 0xC 0 – запись 1 – хранение
8	-		Зарезервировано
7	OE	0	Сигнал OE для RAM 0 – чтение 1 – на выходной шине данных нули
6	-		Зарезервировано
5	CLK	0	Сигнал CLK Переход из 0 в 1 для выполнения операции записи или чтения
4	REG_ACCESS	0	Бит перевода RAM память в регистровый доступ для программирования 0 – доступ осуществляется через I и D шины (выполнение кода) 1 – доступ через регистры периферии по шине S (программирование) При переводе RAM в регистровый доступ исполнение программного кода из RAM невозможно.
3...0	WAITCYCL[3:0]	0x3	Регистр задания числа дополнительных циклов ожидания при работе с RAM памятью Если программа планируется для переноса на 1986BE8Т, то значение WAITCYCL должно быть установлено в соответствии с тактовой частотой по аналогии с данным полем для 1986BE8Т, если не планируется для переноса, то WAITCYCL может быть установлен в 1 для всех частот.

7.10.1.3 ADR

Base ADDR=	0x4000_6000					Offset=	0x0000_0008										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ADR[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ADR[31:0]	0x00000000	Адрес записываемого слова

7.10.1.4 WDATA

Base ADDR=	0x4000_6000					Offset=	0x0000_000C										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
DATA[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Бит	Имя	Значение	Описание
31...0	DATA[31:0]	0x00000000	Данные для записи. При записи в память к ним добавляется поле WECC из регистра CNTR

7.10.1.5 RDATA

Base ADDR=	0x4000_6000					Offset=	0x0000_0010									
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Бит	Имя	Значение	Описание
31...0	DATA[31:0]	0x00000000	Данные при регистровом чтении. Проверка ЕСС для данных не выполняется, поле ЕСС считанных данных отображено в битах RECC регистра CNTR.

7.10.1.6 ECCCS

BaseADDR=	0x4000_6000	Offset=	0x0000_0014												
REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]	0x0000	Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]	0x00	Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT	0	Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT	0	Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC	0	Бит разрешения фиксации в регистрах адреса и данных последней ошибки при двойной ошибке 0 – запрещено 1 – разрешено
4	FIX_SECC	0	Бит разрешения фиксации в регистрах адреса и данных последней ошибки при одинарной ошибке 0 – запрещено 1 – разрешено
3	DECC_IE	0	Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE	0	Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC	0	Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.
0	SECC	0	Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.10.1.7 ECCADR

Base ADDR=	0x4000_6000					Offset=	0x0000_0018										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ECCADR[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ECCADR[15:0]																

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]	0x00000000	Адрес последней двойной или одинарной ошибки

7.10.1.8 ECCDATA

Base ADDR=	0x4000_6000					Offset=	0x0000_001C										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ECCDATA[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ECCDATA[15:0]																

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]	0x00000000	Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC.

7.10.1.9 ECCECC

Base ADDR=		0x4000_6000				Offset=		0x0000_0020								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]	0x00	Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC

7.10.1.10 TEST_TUNING

Base ADDR=	0x4000_6000					Offset=	0x0000_0024										
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
												REPL_TUNING[3:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
					TIME_TUNING[2:0]						TE[1:0]			TUNE[2:0]		

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...16	REPL_TUNING[3:0]	4'b0010	Регистр подстройки OTP 0000 – Большой запас по чтению “1” и меньший по чтению “0” 0001 0010 – Типовое значение ... 1100 1101 1110 1111 – Меньший запас по чтению “1” и больший по чтению “0” Если программа планируется для переноса на 1986BE8Т, то значение REPL_TUNING должно быть установлено в значение по аналогии с данным полем для 1986BE8Т, если не планируется для переноса, то REPL_TUNING может быть установлен в типовое значение.
15...11	-		Зарезервировано
10:8	TIME_TUNING[2:0]	3'b011	Регистр подстройки OTP 000 – Меньше время выборки, меньше общий запас по чтению и “0” и “1” 001 010 011 – Типовое значение 100 101 110 111 – Больше время выборки, больше общий запас по чтению и “0” и “1” Если программа планируется для переноса на 1986BE8Т, то значение TIME_TUNING должно быть установлено в значение по аналогии с данным полем для 1986BE8Т, если не планируется для переноса, то TIME_TUNING может быть установлен в типовое значение
7...6	-	2'b00	Зарезервировано
5...4	TE[1:0]	2'b00	Биты выбора тестового режима при отбраковочном тестировании
3	-	1'b0	Зарезервировано
2...0	TUNE[2:0]	3'b010	Регистр подстройки ОЗУ (при реализации основной памяти программ на ОЗУ – 1986BE81Т) 000 – Минимальное время считывания/записи. Минимальный запас по считыванию/по записи. 001 010 – Типовое значение 011 100 101 110 111 – Максимальное время считывания/записи. Максимальный запас по считыванию/по записи

7.11 Контроллер ПЗУ (ROMCNTR)

Контроллер ПЗУ не имеет каких-либо функций по управлению блоком ПЗУ. Данный блок используются всегда, и контроллер предоставляет центральному процессору только информацию об обнаруженных ошибках при работе с ПЗУ. В микроконтроллере реализовано два контроллера ПЗУ, работающих в режиме LockStep, для проверки корректности исполнения запросов.

Встроенная загрузочная ПЗУ имеет параллельную организацию ECC кодирования и имеет размерность 2048 x (32+8ECC). Память BOOTROM имеет топологически распределенную структуру, т.е. биты одного слова (32 данных + 8 ECC) максимально разнесены по площади блока. Это позволяет при возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а максимум получить единичную ошибку, но во многих словах, вместо множественной ошибки в одном слове.

7.11.1 Описание регистров

Таблица 40 – Описание регистров контроллера ПЗУ

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_9000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	ECCCS		Регистр статуса ошибок ECCROM
	0x0000_0008	ECCADR		Регистр адреса последней ошибки
	0x0000_000C	ECCDATA		Регистр данных последней ошибки
	0x0000_0010	ECCECC		Регистр данных ECC
	0x0000_0014	TEST_TUNING		Регистр подстройки параметров блока

7.11.1.1 KEY

Base ADDR=	0x4000_9000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока ROM_CNTR

7.11.1.2 ECCCS

Base ADDR=		0x4000_9000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SECC_CNT[15:0]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC	

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке 0 – разрешено 1 – запрещено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 0 – разрешено 1 – запрещено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE		Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.
0	SECC		Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.11.1.3 ECCADR

Base ADDR=		0x4000_9000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCADR[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.11.1.4 ECCDATA

Base ADDR=		0x4000_9000				Offset=		0x0000_000C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCDATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

7.11.1.5 ECCECC

Base ADDR=		0x4000_9000				Offset=		0x0000_0010								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-								ECC[7:0]								

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC.

7.11.1.6 TEST_TUNING

Base ADDR=		0x4000_9000				Offset=		0x0000_0014								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-														TUNE[2:0]		

Бит	Имя	Значение	Описание
31...3	-		Зарезервировано
2...0	TUNE	3'b010	Регистр подстройки ПЗУ 000 – Минимальное время считывания. Минимальный запас по считыванию. 001 010 – Типовое значение 011 100 101 110 111 – Максимальное время считывания. Максимальный запас по считыванию.

7.12 Контроллер ОЗУ (RAMCNTR)

Встроенная статическая ОЗУ RAM имеет размерность $32768 \times (32 + 8 \text{ ECC})$ и параллельную организацию ECC-кодирования. В микроконтроллере реализовано два контроллера RAM, работающих в режиме LockStep для проверки корректности исполнения запросов. При записи байта и полуслова в память RAM в контроллере на основании полученных адреса обращения, данных для записи и поля HW ECC[7:0] проверяется их корректность. Затем выполняется чтение $(32 + 8 \text{ ECC})$ разрядной ячейки, расположенной по адресу обращения, а также проверка ECC, считанных из памяти данных. В корректные считанные данные «вставляется» записываемый байт или полуслово, выполняется генерация ECC для модифицированного 32-х разрядного слова и выполняется его запись по адресу обращения. Таким образом, запись байт и полуслов выполняются в два этапа:

- 1 при обращении на запись 32-х разрядного слова в контроллере проверяется ECC записываемого слова;
- 2 генерируется новое ECC и записывается в память.

Запись 32-х разрядного слова выполняются в один этап с проверкой/генерацией ECC. При выполнении операции считывания из встроенного ОЗУ всегда извлекается 32-х разрядное слово с 8-ью битами ECC. В контроллере ОЗУ выполняется проверка считанных данных, которые возвращаются процессору, на входе которого осуществляется еще раз проверка ECC.

Память RAM имеет топологически распределенную структуру, т.е. биты одного слова $(32 \text{ данных} + 8 \text{ ECC})$ максимально разнесены по площади блока. Это позволяет при возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а получить единичную ошибку во многих словах, вместо множественной ошибки в одном слове.

7.12.1 Описание регистров

Таблица 41 – Описание регистров контроллера ОЗУ

Базовый адрес	Смещение	Название	Описание
0x4000_8000	0x0000_0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	ECCCS	Регистр статуса ошибок ECCROM
	0x0000_0008	ECCADR	Регистр статуса ошибок ECCDRAM
	0x0000_000C	ECCDATA	Регистр статуса ошибок ECCDRAM
	0x0000_0010	ECCECC	Регистр данных ECC последней ошибки
	0x0000_0014	TEST_TUNING	Регистр подстройки параметров блока

7.12.1.1 KEY

Base ADDR=	0x4000_8000	Offset=	0x0000_0000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока RAM_CNTR

7.12.1.2 ECCCS

Base ADDR=		0x4000_8000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SECC_CNT[15:0]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC	

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 или более ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке 0 – разрешено 1 – запрещено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 0 – разрешено 1 – запрещено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
2	SECC_IE		Бит разрешения прерывания при возникновении одиночной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.
0	SECC		Флаг возникновения одиночной ошибки 0 – ошибок не было 1 – ошибка была Сбрасывается записью 1.

7.12.1.3 ECCADR

Base ADDR=		0x4000_8000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCADR[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.12.1.4 ECCDATA

Base ADDR=		0x4000_8000				Offset=		0x0000_000C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ECCDATA[31:16]																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

7.12.1.5 ECCECC

Base ADDR=		0x4000_8000				Offset=		0x0000_0010									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
-																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
-								ECC[7:0]									

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC

7.12.1.6 TEST_TUNING

Base ADDR=		0x4000_8000				Offset=		0x0000_0014									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
														TUNE[2:0]			

Бит	Имя	Значение	Описание
31...3	-		Зарезервировано
2...0	TUNE	3'b010	Регистр подстройки ОЗУ 000 – Минимальное время считывания/записи. Минимальный запас по считыванию/по записи. 001 010 – Типовое значение 011 100 101 110 111 – Максимальное время считывания/записи. Максимальный запас по считыванию/по записи.

7.13 Контроллер DMA (DMACNTR)

7.13.1 Основные характеристики контроллера DMA

Основные характеристики и отличительные особенности контроллера DMA:

- 32 канала DMA;
- каждый канал DMA имеет свои сигналы управления передачей данных;
- каждый канал DMA имеет программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается, исходя из уровня приоритета, определяемого номером канала DMA;
- поддержка различного типа передачи данных:
 - память – память;
 - память – периферия;
 - периферия – память;
- поддержка различных типов DMA циклов;
- поддержка передачи данных различной разрядности;
- каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;
- все управляющие данные канала хранятся в системной памяти;
- разрядность данных приемника равна разрядности данных передатчика;
- количество передач в одном цикле DMA может программироваться от 1 до 1024;
- инкремент адреса передачи может быть больше чем разрядность данных.

7.13.2 Термины и определения

При описании контроллера DMA используются следующие термины:

Таблица 42 – Термины и определения

Термины/определения	Описание
Альтернативная	Альтернативная структура управляющих данных канала. Для изменения типа структуры данных можно установить соответствующий регистр (см. подраздел 7.13.5 «Структура управляющих данных канала»)
С	Идентификатор номера канала прямого доступа. Например: С=1 – канал DMA 1 С=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA
Управляющие данные канала	Структура данных находится в системной памяти. Можно программировать структуру данных так, чтобы контроллер выполнял передачу данных по каналу DMA в желаемом режиме. Контроллер должен иметь доступ к области системной памяти, где находится эта информация. <i>Примечание</i> – Любое упоминание в данном разделе структуры данных означает управляющие данные канала
Цикл DMA	Все передачи DMA, которые контроллер должен выполнить для передачи N пакетов данных
Передача DMA	Акция пересылки одного байта, полуслова или слова. Общее количество передач DMA, которые контроллер выполняет для канала
Пинг-понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую структуру данных. Контроллер сигнализирует об окончании каждого цикла

	DMA, позволяя главному процессору перенастраивать неактивную структуру данных. Контроллер продолжает переключаться от первичной к альтернативной структуре данных и обратно до тех пор, пока он не прочитает «неправильную» структуру данных, или пока он не завершит цикл без переключения к другой структуре
Первичная	Первичная структура управляющих данных канала. Контроллер использует эту структуру данных, если соответствующий разряд в регистре <code>chnl_pri_alt_set</code> установлен в 0.
R	Степень числа 2, устанавливающая число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 10
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После окончания цикла, в случае, если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. Затем контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг <code>dma_done</code> , если окончание подобного режима работы происходит после выполнения обычного цикла

7.13.3 Функциональное описание

Ниже показана упрощенная структурная схема контроллера DMA (Рисунок 66).

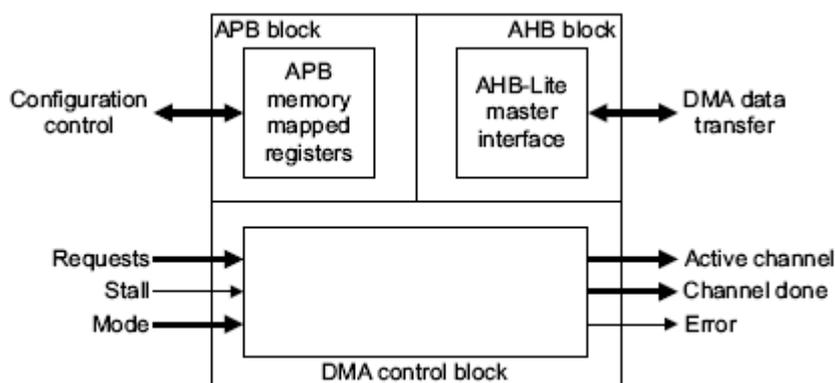


Рисунок 66 – Структурная схема контроллера DMA

Контроллер состоит из следующих основных функциональных блоков:

- блок, подключенный к шине APB;
- блок, подключенный к шине AHB;
- управляющий блок DMA.

7.13.3.1 Типы транзакций АНВ

Контроллер интерфейса не поддерживает пакетные (burst) транзакции на шине АНВ, выполняет только одиночные (non-sequence) транзакции. Отсутствие возможности осуществлять пакетные транзакции оказывает минимальное влияние на производительность системы, так как пакетные транзакции более эффективны в одноуровневых системах с шиной АНВ, где блоки должны «захватывать» шину или обращаться к внешней памяти. В микроконтроллере контроллер DMA подключен к многоуровневой системе с шиной АНВ, включающей встроенную память. Стоит обратить внимание, что количество DMA-передач до проведения арбитража, определяемое числом R, не связано с типом транзакций на шине АНВ. Как уже было сказано, контроллер DMA всегда использует одиночные транзакции АНВ.

7.13.3.2 Разрядность передач данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32-х разрядных данных. Таблица перечисляет значения комбинаций шины HSIZE.

Таблица 43 – Комбинации шины HSIZE

HSIZE[2] ^{*)}	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
	1	0	32
	1	1	**)

^{*)} – сигнал постоянно удерживается в состоянии логической ноль;

^{**)} – запрещенная комбинация.

Контроллер всегда использует передачи 32-х разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника, соответствующую разрядности данных приемника.

7.13.3.3 Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола АНВ-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

Таблица 44 перечисляет значения комбинаций шины HPROT.

Таблица 44 – Режимы защиты данных

HPROT[3] Кэширование	HPROT[2] буферизация	HPROT[1] Привилегированный	HPROT[0] Данные/команда	Описание
-	-	-	1 ^{*)}	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ
-	0	-	-	Без буферизации
-	1	-	-	Буферизированный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный

^{*)} – Контроллер удерживает HPROT[0] в состоянии логической единицы, чтобы обозначить доступ к данным.

Для каждого цикла DMA возможен выбор режимов защиты данных передач источника и приемника (подробнее см. в подразделе «Структура управляющих данных канала»).

Для каждого канала DMA также возможен выбор режима защиты данных (подробнее см. в подразделе 7.13.4 «Управление DMA»).

7.13.3.4 Инкремент адреса

Контроллер позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса зависит от разрядности передаваемых данных. В таблице перечислены возможные комбинации.

Таблица 45 – Инкремент адреса

Разрядность данных	Величина инкремента
8	Байт, полуслово, слово
16	Полуслово, слово
32	слово

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером, – одно слово. Более подробно о настройке инкремента адреса написано в подразделе «Структура управляющих данных канала». Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

Примечание – Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. подраздел «Структура управляющих данных канала»).

7.13.4 Управление DMA

7.13.4.1 Правила обмена данными

Контроллер использует правила обмена данными, перечисленные в таблице , при соблюдении следующих условий:

- канал DMA включен, что выполняется установкой в состояние логической единицы разрядов управления `chnl_enable_set[C]` и `master_enable`;
- флаги запроса `dma_req[C]` и `dma_sreq[C]` не замаскированы, что выполняется установкой в состояние логического нуля разряда управления `chnl_req_mask_set[C]`;
- контроллер находится не в тестовом режиме, что выполняется установкой в состояние логического нуля разряда управления `int_test_en bit[C]`.

Таблица 46 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
1	Если <code>dma_active[C]</code> установлен в 0, то установка в 1 <code>dma_req[C]</code> или <code>dma_sreq[C]</code> на один или более тактов сигнала <code>hclk</code> , следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда <code>dma_active[C]</code>
3	Контроллер устанавливает в 1 <code>dma_active[C]</code> в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или чем число передач, указанное в регистре <code>n_minus_1</code> . В периферийном режиме «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. Таким образом, контроллер выполняет 2^R передач, используя первичную структуру управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре <code>n_minus_1</code>), используя альтернативную структуру управляющих данных. По окончании последней передачи <code>dma_active[C]</code> сбрасывается в 0
5	Контроллер устанавливает <code>dma_active[C]</code> в 0 на, как минимум, один такт сигнала <code>hclk</code> перед тем, как снова установит <code>dma_active[C]</code> или <code>dma_active[]</code> в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного <code>dma_done[]</code>

Правило	Описание
7	Если dma_req[C] устанавливается в состояние 1 в момент, когда dma_active[C] или dma_stall также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды cycle_ctrl для канала установлены в состоянии 3'b100, 3'b101, 3'b110, 3'b111, то dma_done[C] никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды cycle_ctrl позволяют удержание dma_done[C], то по срезу сигнала dma_active[] произойдут события: - если dma_stall в состоянии 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт hclk; - если dma_stall в состоянии 1, работа контроллера приостановлена. После того, как dma_stall будет установлен в 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт hclk
10	Состояние dma_waitonreq[C] можно изменять только при выключенном канале
11	Если dma_waitonreq[C] в состоянии 1, то сигнал dma_active[C] не перейдет в состояние 0 до тех пор, пока: - контроллер завершит 2 ^R передач (или число передач, указанное в регистре n_minus_1); - dma_req[C] будет установлен в 0; - dma_sreq[C] будет установлен в 0
12	Если за один такт сигнала hclk перед установкой dma_active[C] в 0 dma_stall устанавливается в 1, то: контроллер установит dma_active[C] в 0 на следующем такте сигнала hclk; передача по каналу C не завершится, пока не будет сброшен в 0 dma_stall
13	Контроллер игнорирует dma_sreq[C], если dma_waitonreq[C] в состоянии 0
14	Контроллер игнорирует dma_sreq[C], если chnl_useburst_set[C] в состоянии 1 ^{*)}
15	Для циклов DMA, отличных по типу от периферийного режима «Исполнение с изменением конфигурации», по окончании 2 ^R передач контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2 ^R . В периферийном режиме «Исполнение с изменением конфигурации» контроллер устанавливает значение chnl_useburst_set[C] в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2 ^R .
16	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет одну DMA передачу. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, контроллер выполняет 2 ^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
17	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1, а dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2 ^R (или число передач, указанное в регистре n_minus_1) DMA передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2 ^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2 ^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]

*) – Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2^R, то контроллер не очистит разряды chnl_useburst_set, и запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач.

При отключении канала контроллер осуществляет DMA передачи согласно правилам, представленным в таблице .

Таблица 47 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
19	Если dma_req[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
20	Если dma_sreq[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1 при условии dma_waitonreq[C] в 1 и chnl_useburst_set[C] в состоянии 0. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
21	dma_active[C] всегда удерживается в состоянии 0

7.13.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает примеры функционирования контроллера с использованием правил обмена данными, представленных ранее (Таблица 46):

- импульсный запрос на обработку;
- запрос по уровню на обработку;
- флаги завершения;
- флаги ожидания запроса на обработку.

Примечание – Все диаграммы, показанные далее на рисунках в этом подразделе (Рисунок 67 – Рисунок 71), подразумевают следующее:

- hready находится в состоянии 1;
- АНВ «ведомый» всегда дает ответ «ОКAY».

7.13.4.3 Импульсный запрос на обработку

Рисунок 67 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

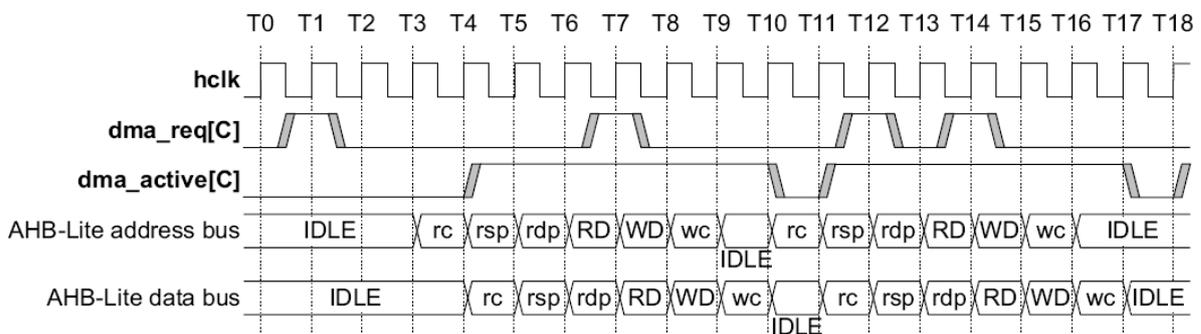


Рисунок 67 – Диаграмма работы при получении импульсного запроса

Пояснения к диаграмме на рисунке приведены ниже.

Таблица 48 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу C (Таблица 46, правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C

T4-T7	Контроллер считывает управляющую данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу С (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража
T7-T9	Контроллер выполняет передачу DMA по каналу С, где: RD – чтение данных; WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T10-T11	Контроллер удерживает dma_active[C] на, как минимум, один такт hclk (см. правило 5)
T11	Если канал С имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте Т7 был получен запрос на обработку (см. правила 2 и 3)
T12	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу С (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража
T14	Контроллер игнорирует запрос по каналу С из-за отложенного запроса, полученного на такте Т12
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T17-T18	Контроллер удерживает dma_active[C], как минимум, на один такт hclk (см. правило 5)
T18	Если канал С имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте Т12 был получен запрос на обработку (см. правила 2 и 3)

7.13.4.4 Запрос на обработку по уровню

Рисунок 68 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

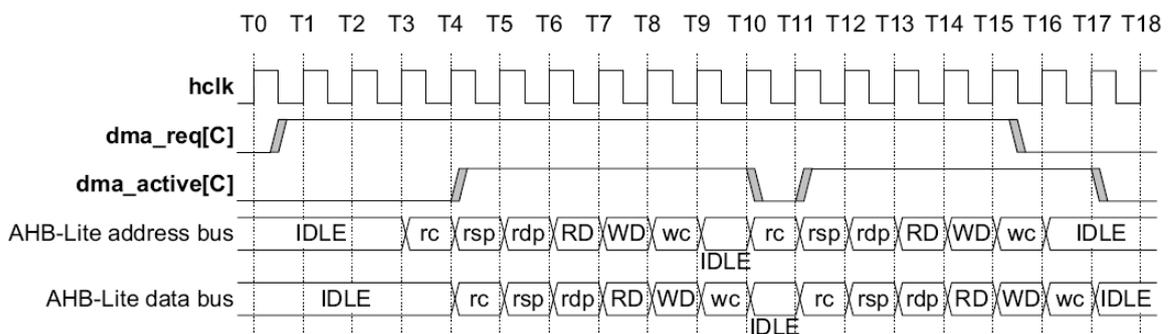


Рисунок 68 – Диаграмма работы при получении запроса на обработку по уровню

Пояснения к диаграмме (Рисунок 68) даны в таблице ниже (Таблица 49).

Таблица 49 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу С (Таблица 46, правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу С

T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает dma_active[C] на как минимум один такт hclk (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C] и начинает вторую DMA передачу по каналу C
T11-T14	Контроллер считывает управляющие данные канала
T14-T16	Контроллер выполняет передачу DMA по каналу C
T15-T16	Периферийный блок обнаруживает, что передача DMA началась и сбрасывает dma_req[C]
T16-T17	Контроллер осуществляет запись настроек канала channel_cfg
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)

При использовании запроса на обработку по уровню периферийный блок может не обладать достаточным быстродействием, чтобы вовремя снять сигнал запроса, в этом случае он должен установить сигнал dma_stall. Установка сигнала dma_stall предотвращает повторение выполненной передачи.

7.13.4.5 Флаги завершения

Рисунок 69 демонстрирует функционирование сигнала (флага) dma_done[] при следующих условиях:

- dma_stall и dma_waitonreq[] находятся в состоянии 0;
- dma_stall установлен в 1;
- dma_waitonreq[] установлен в 1.

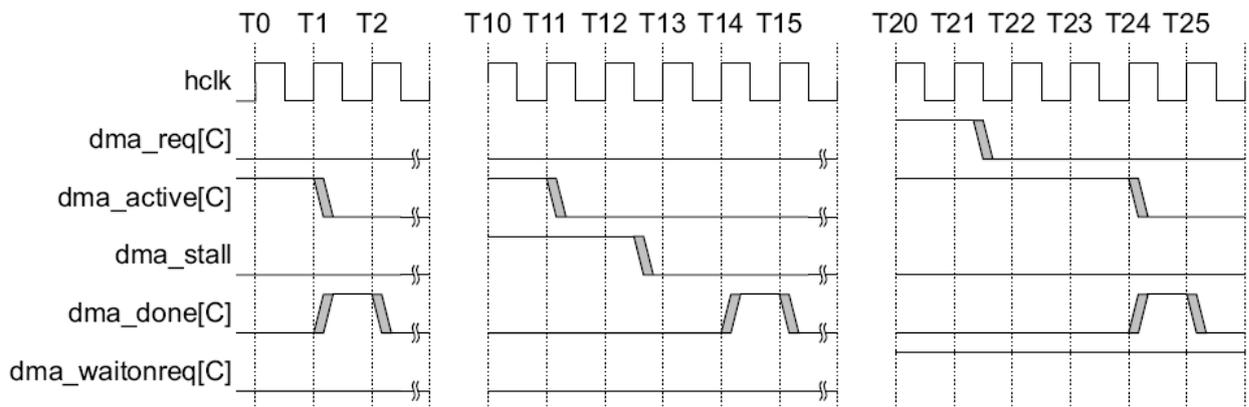


Рисунок 69 – Диаграммы функционирования dma_done

Пояснения к диаграмме (Рисунок 69) приведены в таблицах , , .

Таблица 50 – Пояснения функционирования dma_done, такты от T0 до T2

T1	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (Таблица 46, правило 4)
T1-T2	Контроллер завершает цикл DMA и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

Таблица 51 – Пояснения функционирования dma_done, такты от T10 до T15

T11	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T12-T13	Периферийный блок сбрасывает сигнал dma_stall
T14-T15	Контроллер завершает цикл DMA и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

Примечание к T11 – Контроллер не устанавливает сигнал dma_done[C], так как сигнал dma_stall установлен в 1 в предшествующем такте hclk (см. правила 9 и 12).

Таблица 52 – Пояснения функционирования dma_done, такты от T20 до T25

T20	Контроллер выполнил передачу DMA, но из-за установленного в 1 dma_waitonreq[C] он должен ожидать сброса в 0 сигнала dma_req[C], перед тем как сбросить dma_active[C] (см. правило 11) и установить dma_done[C] (см. правило 9)
T21-T25	Периферийный блок сбрасывает dma_req[C]
T24	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T24-T25	Контроллер завершает цикл DMA и, если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

7.13.4.6 Флаги ожидания запроса на обработку

Ниже приведены рисунки, которые демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2^R передач и одиночных передач:

- диаграмма работы контроллера DMA при использовании периферией dma_waitonreq;
- диаграмма работы контроллера DMA при использовании периферией dma_waitonreq совместно с dma_sreq.

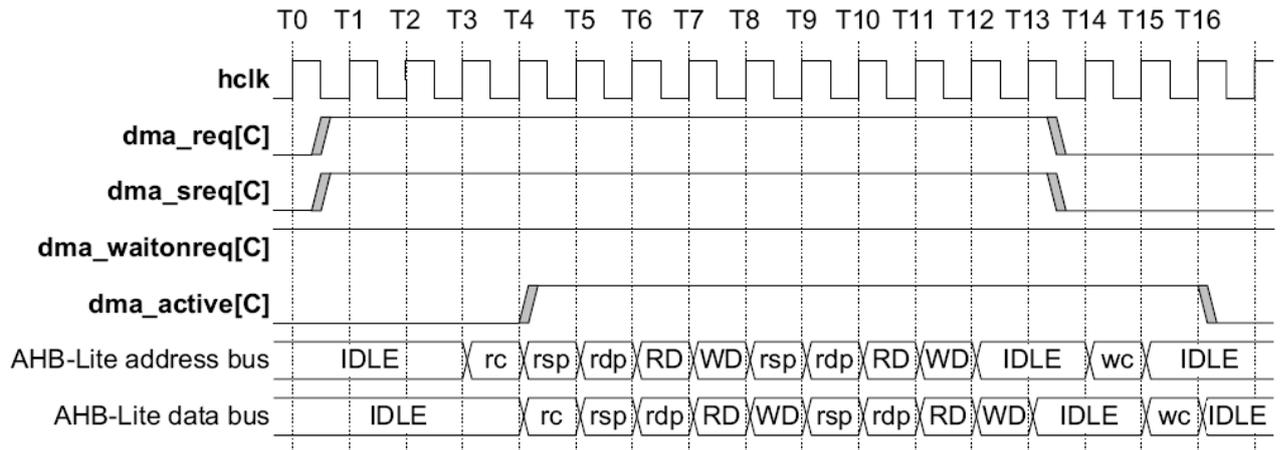


Рисунок 70 – Диаграмма работы контроллера DMA при использовании dma_waitonreq

Пояснения к диаграмме на рисунке приведены в таблице .

Таблица 53 – Пояснения работы контроллера DMA при использовании dma_waitonreq

T0-T16	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T3-T4	Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T11	Контроллер считывает 2 указателя адреса окончания данных rsp и rdp
T11-T13	Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C]
T15-T16	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T16	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11). Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2R (см. правило 15)

Рисунок 71 показывает работу контроллера DMA при установке dma_waitonreq в 1 и выполнении одиночной DMA передачи.

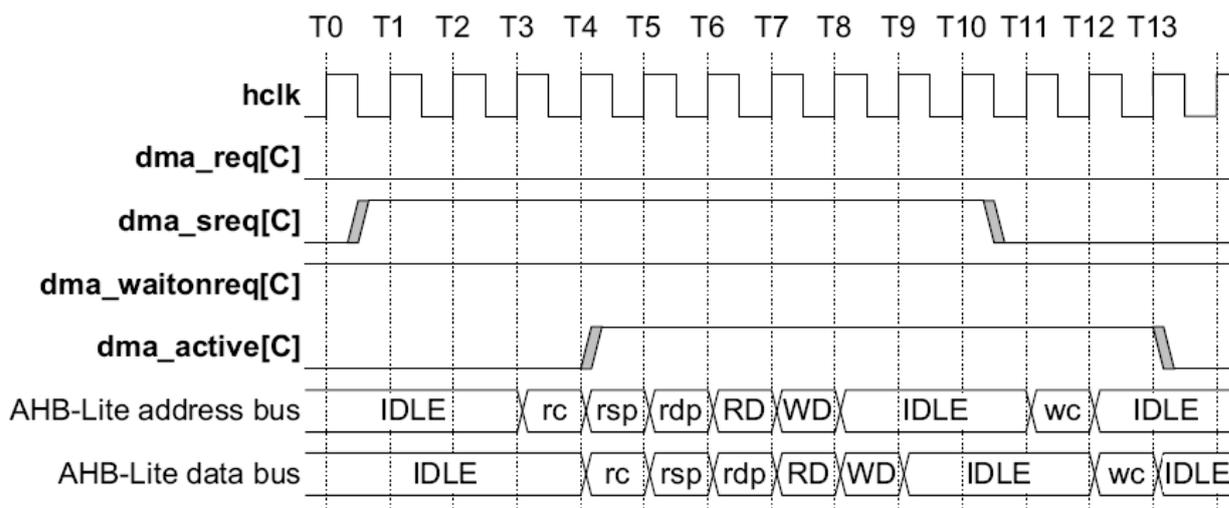


Рисунок 71 – Работа DMA при использовании dma_waitonreq совместно с dma_sreq

Пояснения к диаграмме на рисунке приведены в таблице .

Таблица 54 – Пояснения работы DMA при использовании dma_waitonreq совместно с dma_sreq

T0-T13	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_useburst_set[C] находится в состоянии 0 (см. правила 13 и 14)
T3-T4	Контроллер отвечает на dma_sreq[C] запрос (см. правила 16)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных. Это запрос в ответ на dma_sreq[], таким образом, R=0 и, следовательно, контроллер исполнит 1 DMA передачу
T10-T11	Периферийный блок сбрасывает сигнал dma_sreq[C]
T12-T13	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T13	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11)

7.13.4.7 Правила арбитража DMA

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет 4 разряда, которые определяют количество транзакций по шине AHB до повторения арбитража. Эти разряды задают степень R числа 2; изменение R напрямую устанавливает периодичность арбитража как 2 в степени R. Для примера, если R равно 4, то арбитраж будет проводиться через каждые 16 передач DMA.

Таблица 55 показывает возможную периодичность арбитража.

Таблица 55 – Периодичность арбитража в единицах передач по шине АНВ

Значение R	Периодичность арбитража каждые x передач DMA
b0000	1
b0001	2
b0010	4
b0011	8
b0100	16
b0101	32
b0110	64
b0111	128
b1000	256
b1001	512
b1010–b1111	1024

Внимание! Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При $N > 2^R$ (N – номер передачи) и в случае, если результат деления 2^R на N не целое число, контроллер всегда выполняет последовательность из 2^R передач до тех пор, пока не станет верным $N < 2^R$. Контроллер выполняет оставшиеся N передач в конце цикла DMA.

Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в разделе «Управляющие данные канала».

7.13.4.8 Приоритет

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

- номер канала
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда `chnl_priority_set`.

Канал номер 0 имеет высший уровень приоритета, и уровень приоритета снижается с увеличением номера канала. Таблица 56 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 56 – Уровень приоритета каналов DMA

Уровень приоритета в порядке его уменьшения	Номер канала	Уровень приоритета, установленный битом <code>chnl_priority_set</code>
Наивысший уровень приоритета	0	Высокий
-	1	Высокий
-	2	Высокий
.....
-	30	Высокий
-	31	Высокий
-	0	По умолчанию (низкий)
-	1	По умолчанию (низкий)
-	2	По умолчанию (низкий)
.....
-	30	По умолчанию (низкий)
Низший уровень приоритета	31	По умолчанию (низкий)

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. Рисунок 72 иллюстрирует процесс выбора следующего канала для обслуживания.

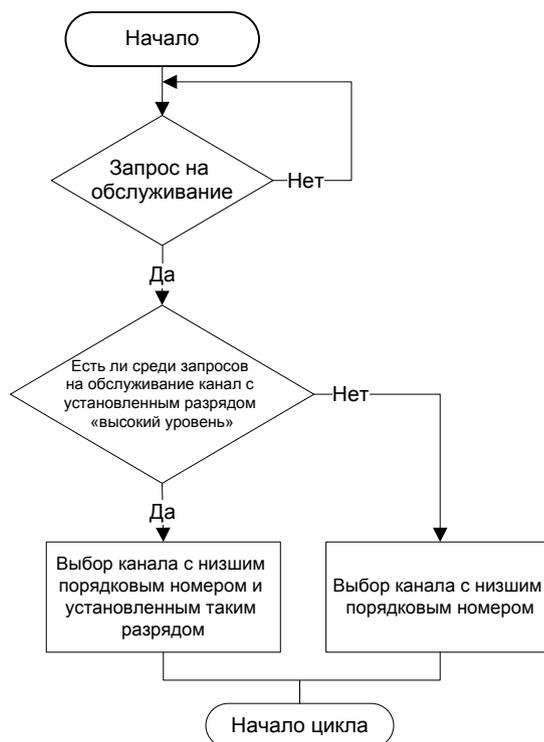


Рисунок 72 – Алгоритм выбора следующего канала для обслуживания

7.13.4.9 Типы циклов DMA

Разряды `cycle_ctrl` определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено ниже (Таблица 57).

Таблица 57 – Типы циклов DMA

cycle_ctrl	Описание
b000	Структура управляющих данных канала в запрещенном состоянии
b001	Обычный цикл DMA
b010	Автозапрос
b011	Режим «пинг-понг»
b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
b101	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала

Примечание – Разряды `cycle_ctrl` находятся в области памяти, отведенной под `channel_cfg` – см. раздел «Настройка управляющих данных канала».

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач

DMA по данному каналу. Поэтому, устанавливая значение R, необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие **типы циклов DMA**:

- недействительный;
- основной;
- автозапрос;
- «пинг-понг»;
- работа с памятью в режиме «исполнение с изменением конфигурации»;
- работа с периферией в режиме «исполнение с изменением конфигурации».

Недействительный

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

Основной

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала, и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

- 1 Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3.
- 2 Осуществление арбитража:
 - если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала;
 - если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), то контроллер переходит к шагу 1.
- 3 Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору на завершение цикла DMA.

Авто-запрос

Функционируя в данном режиме, контроллер ожидает получения одиночного запроса на обработку для разрешения работы и выполнения цикла DMA. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от процессора или периферийных блоков.

Контроллер позволяет выбрать для использования первичную или альтернативную структуру управляющих данных канала. После того, как разрешена работа канала, и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

- 1 Контроллер выполняет 2^R передач для канала C. Если число оставшихся передач 0, контроллер переходит к шагу 3.
- 2 Контроллер осуществляет арбитраж. Когда канал C становится каналом с самым высоким приоритетом, контроллер переходит к шагу 1.
- 3 Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору на завершение цикла DMA.

Пинг-понг

В данном режиме контроллер выполняет цикл DMA, используя одну из структур управляющих данных, а затем выполняет еще один цикл DMA, используя другую структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает «неправильную» структуру данных или пока процессор не запретит работу канала.

Рисунок 73 демонстрирует пример функционирования контроллера в режиме «пинг-понг».

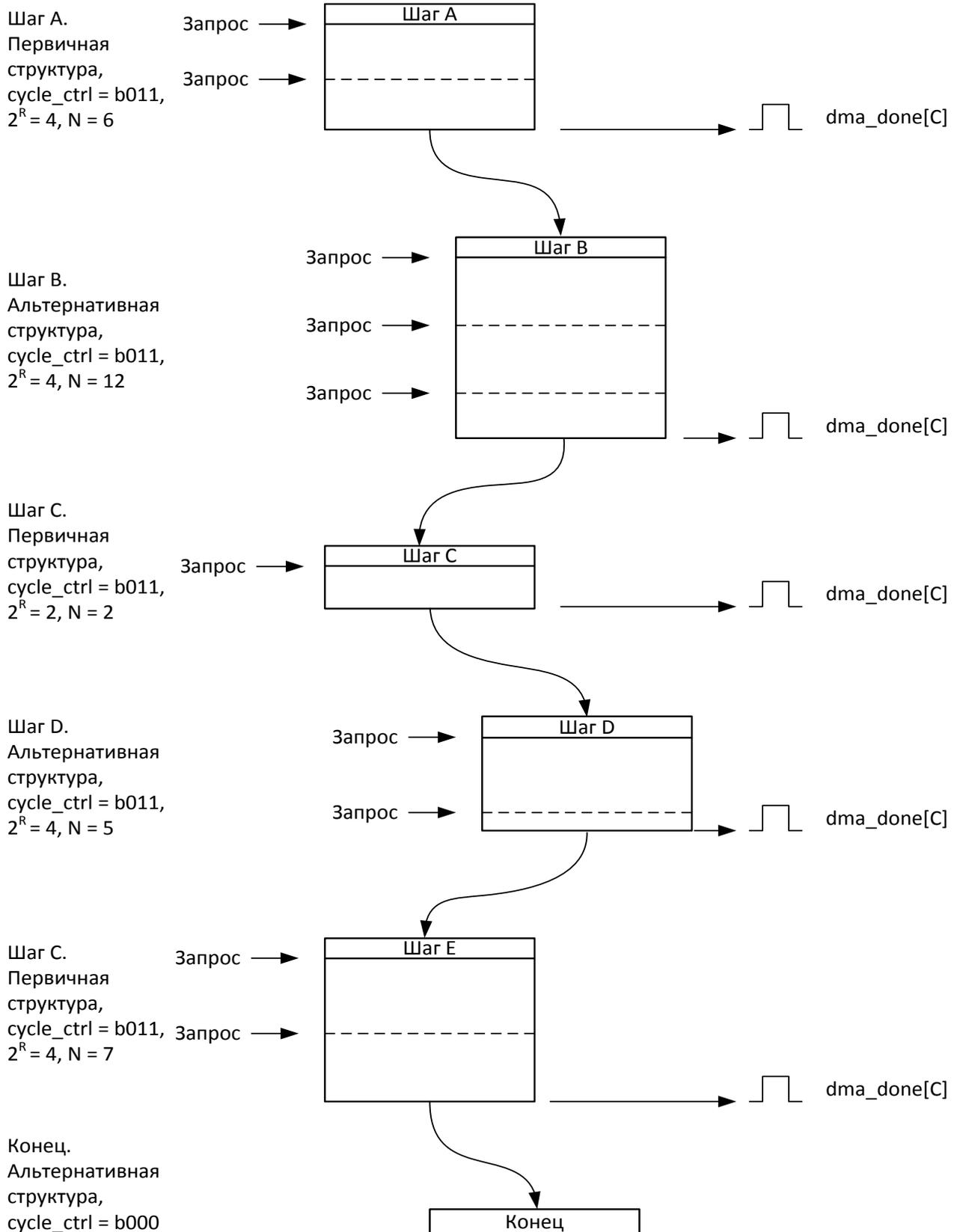


Рисунок 73 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к схеме на рисунке :

Шаг А Процессор устанавливает первичную структуру управляющих данных для шага А. Процессор устанавливает альтернативную структуру управляющих данных для шага В. Это позволит контроллеру преступить к шагу В незамедлительно после выполнения шага А, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.
Контроллер получает запрос и выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага А процессор может установить первичные управляющие данные канала для шага С. Это позволит контроллеру переключиться к шагу С незамедлительно после выполнения шага В, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг В:

Шаг В Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 4 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага В процессор может установить альтернативные управляющие данные канала для шага D.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг С:

Шаг С Контроллер выполняет 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага С процессор может установить первичные управляющие данные канала для шага Е.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D:

Шаг D Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов
Контроллер выполняет оставшуюся передачу DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После получения нового запроса на обработку от канала при условии его наивысшего приоритета выполняется шаг Е:

Шаг Е Контроллер выполняет 4 передачи DMA.
 Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
 Контроллер выполняет оставшиеся 3 передачи DMA.
 Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончании шага D контроллер установил cycle_ctrl в состояние b000, передачи DMA прекращаются.

Примечание – Для прерывания цикла DMA, исполняемого в режиме «пинг-понг», также возможен перевод режима работы контроллера на шаге Е в режим «Основной цикл DM» путем установки cycle_ctrl в 3'b001.

Режим работы с памятью «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще 4 передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание – После исполнения контроллером N передач с использованием первичных управляющих данных он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3'b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 58 перечисляет области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Таблица 58 – Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

№ бита	Обозначение	Значение	Описание
Области с константными значениями			
31...30	dst_inc	b'10	Контроллер производит инкремент адреса пословно
29...28	dst_size	b'10	Контроллер осуществляет передачу пословно
27...26	src_inc	b'10	Контроллер производит инкремент адреса пословно
25...24	src_size	b'10	Контроллер осуществляет передачу пословно
17...14	R_power	b'0010	Контроллер выполняет 4 передачи DMA
3	next_useburst	b'0	Для данного режима этот разряд должен быть равен 0
2...0	cycle_ctrl	b'100	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»

Области со значениями, определяемыми пользователем			
23...21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N ^{*)}	Настраивает контроллер на выполнение N передач DMA, где N кратно 4

*) – Так как R_{power} задает значение 4, то необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 74 демонстрирует пример функционирования в режиме работы с памятью «Исполнение с изменением конфигурации».

Инициализация:

- 1 Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=b100, 2^R=4, N=16.
- 2 Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b101, 2 ^R = 4, N = 3	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2 ^R = 2, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2 ^R = 8, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 ^R = 4, N = 4	0xFFFFFFFF

Memory scatter-gather transaction:

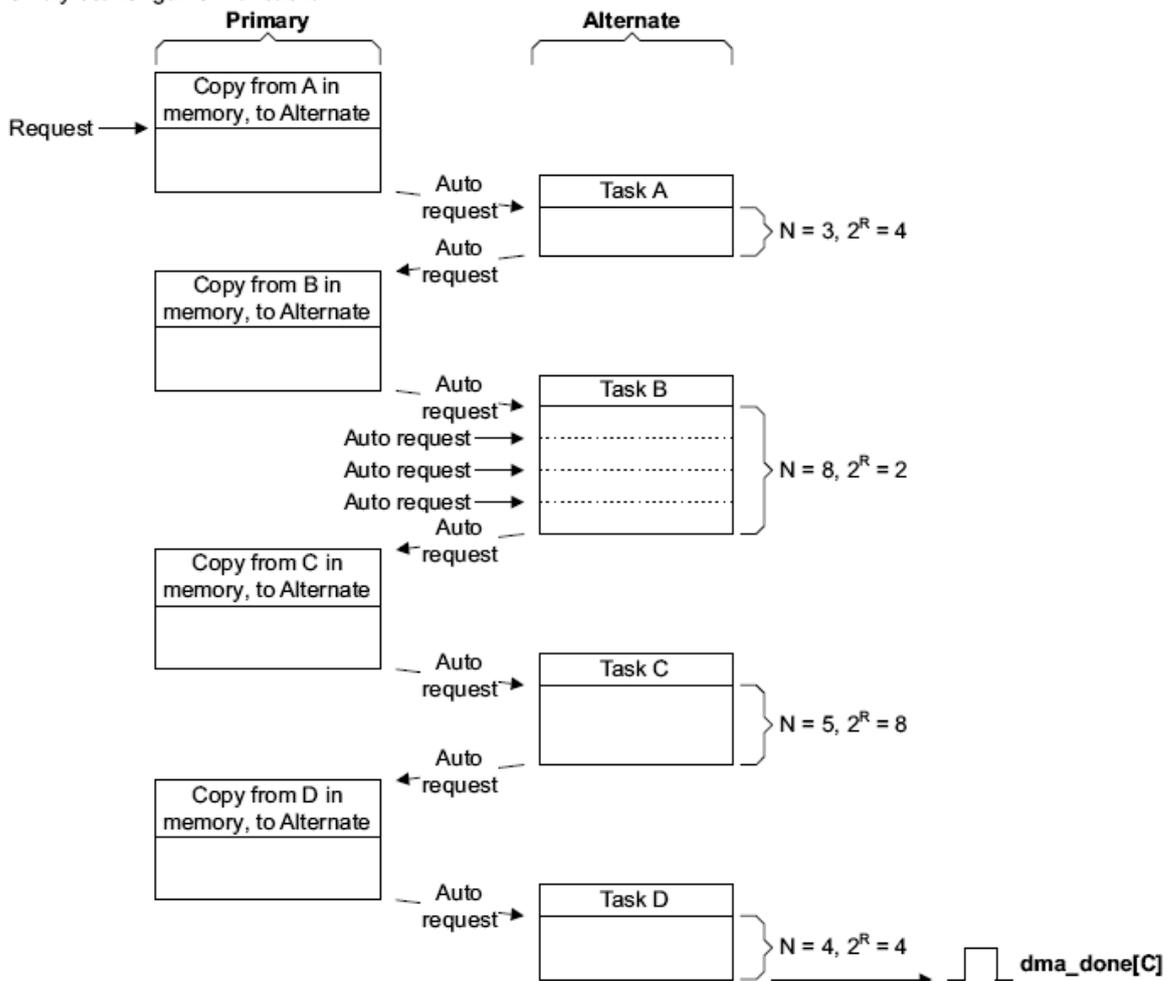


Рисунок 74 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Пояснения к схеме (Рисунок 74):

Инициализация:

- 1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации» путем установки `cycle_ctrl` в `b100`. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому `N` установлен в 16.
- 2 Процессор записывает управляющие данные для шагов А, В, С, D в область памяти с адресом, указанным в `src_data_end_ptr`.
- 3 Процессор разрешает работу канала DMA. Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]` или запроса от процессора. Порядок выполнения следующий:

Первичная, копирование А

По получении запроса на обслуживание контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг А

Контроллер выполняет шаг А. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование В

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг В

Контроллер выполняет шаг В. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование С

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг С

Контроллер выполняет шаг С. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование D

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D. Контроллер устанавливает `cycle_ctrl` первичных управляющих данных в `b000` для индикации о том, что эта структура управляющих данных является «неправильной». Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA. Контроллер устанавливает флаг `dma_done[C]` в состояние 1 на один такт сигнала `hclk` и входит в процедуру арбитража.

Режим работы с периферией «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач

контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Примечание – Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

После того, как этот цикл завершился, контроллер выполняет арбитраж и по получении запроса на обслуживание от периферии, имеющего наивысший приоритет, он выполняет еще 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание – После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3'b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла. В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных.

Таблица 59 перечисляет области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Таблица 59 – Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

№ бита	Обозначение	Значение	Описание
Области с константными значениями			
31...30	dst_inc	b'10	Контроллер производит инкремент адреса пословно
29...28	dst_size	b'10	Контроллер осуществляет передачу пословно
27...26	src_inc	b'10	Контроллер производит инкремент адреса пословно
25...24	src_size	b'10	Контроллер осуществляет передачу пословно
17...14	R_power	b'0010	Контроллер выполняет 4 передачи DMA
2...0	cycle_ctrl	b'110	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23...21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N ^{*)}	Настраивает контроллер на выполнение N передач DMA, где N кратно 4.
3	next_useburst	-	При установке в 1 контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой.

^{*)} – Так как R_power задает значение 4, то необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные.

Следующий рисунок (Рисунок 75) демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

Инициализация:

- 1 Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=b'110, 2^R=4, N=16.

- 2 Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b111, 2 ^R = 4, N = 3	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2 ^R = 2, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2 ^R = 8, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 ^R = 4, N = 4	0xFFFFFFFF

Peripheral scatter-gather transaction:

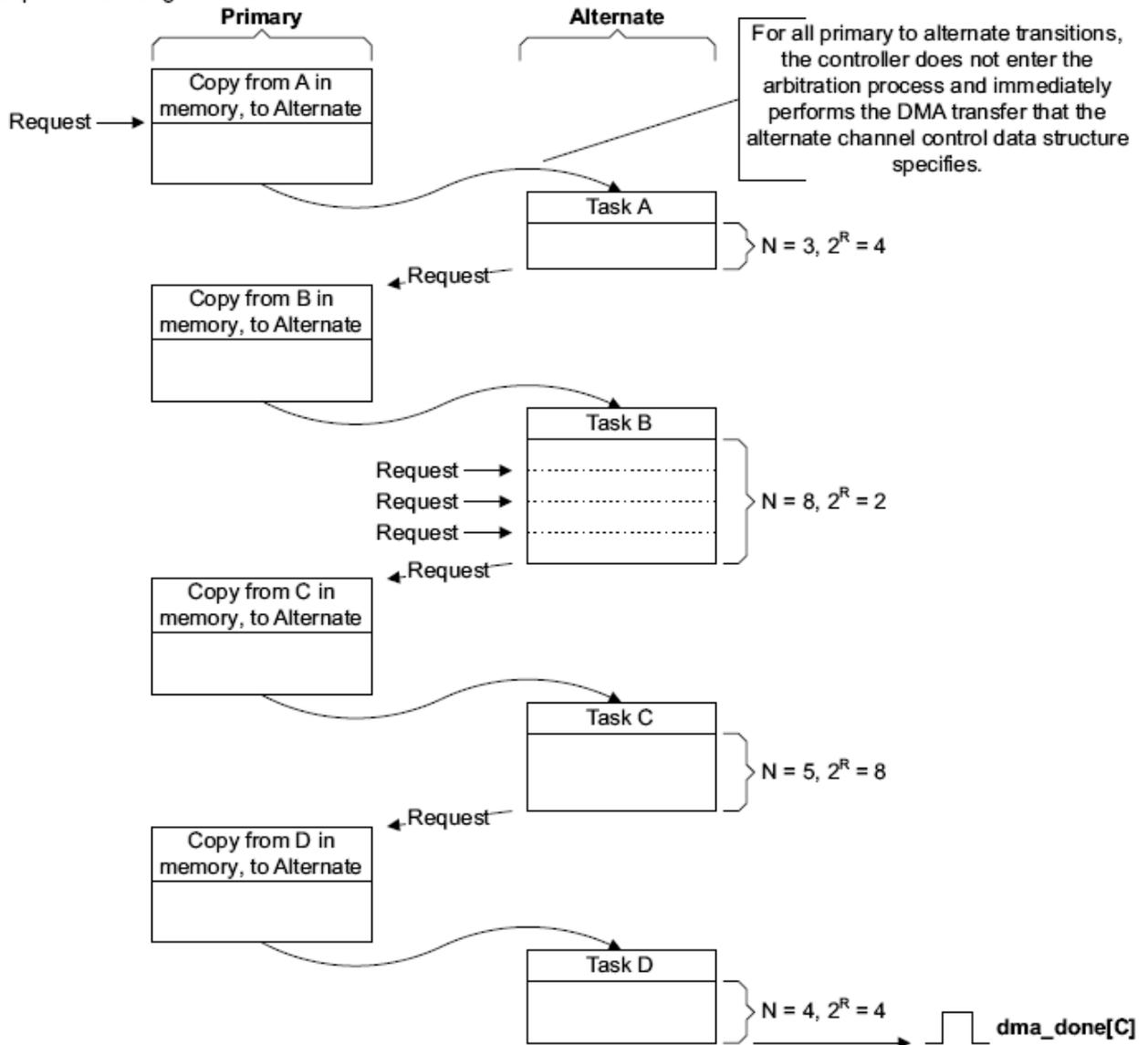


Рисунок 75 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Пояснения к схеме (Рисунок 75):

Инициализация:

- 1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки `csycle_ctrl` в `b110`. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому `N` установлен в 16.
- 2 Процессор записывает управляющие данные для шагов А, В, С, D в область памяти с адресом, указанным в `src_data_end_ptr`.
- 3 Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]`. Передачи выполняются следующим образом:

Первичная, копирование из области А памяти

По получению запроса на обслуживание, контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А.

Шаг А

Контроллер выполняет шаг А. По окончании контроллер проводит процедуру арбитража. Первичная, копирование из области В памяти. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В.

Шаг В

Контроллер выполняет шаг В. Для завершения задачи периферия должна установить последовательно 3 запроса. По окончании контроллер проводит процедуру арбитража. Первичная, копирование из области С памяти. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С.

Шаг С

Контроллер выполняет шаг С. По окончании контроллер проводит процедуру арбитража. После выставления периферией нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование из области D памяти

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D. Контроллер устанавливает `csycle_ctrl` первичных управляющих данных в `b000` для индикации о том, что эта структура управляющих данных является «неправильной».

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA. Контроллер устанавливает флаг `dma_done[C]` в состояние 1 на один такт сигнала `hclk` и входит в процедуру арбитража.

7.13.4.10 Индикация ошибок

При получении контроллером по шине АНВ ответа об ошибке контроллер выполняет следующие действия:

- отключает канал, связанный с ошибкой;
- устанавливает флаг `dma_err` в состояние 1.

После обнаружения процессором флага `dma_err` процессор определяет номер канала, который был активен в момент появления ошибки. Для этого он осуществляет следующее:

- чтение регистра `chnl_enable_set` с целью создания списка отключенных каналов;
- если канал установил флаг `dma_done[]`, то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги `dma_done[]`;
- процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги `dma_done[]`. Канал, по которому отсутствуют данные об установке флага `dma_done[]`, это и есть канал, с которым связана ошибка.

7.13.5 Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

- предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;
- иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 76 показывает область памяти, необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

Alternate data structure		Primary data structure	
Alternate_Ch_31	0x3F0	Primary_Ch_31	0x1F0
Alternate_Ch_30	0x3E0	Primary_Ch_30	0x1E0
Alternate_Ch_29	0x3D0	Primary_Ch_29	0x1D0
Alternate_Ch_28	0x3C0	Primary_Ch_28	0x1C0
Alternate_Ch_27	0x3B0	Primary_Ch_27	0x1B0
Alternate_Ch_26	0x3A0	Primary_Ch_26	0x1A0
Alternate_Ch_25	0x390	Primary_Ch_25	0x190
Alternate_Ch_24	0x380	Primary_Ch_24	0x180
Alternate_Ch_23	0x370	Primary_Ch_23	0x170
Alternate_Ch_22	0x360	Primary_Ch_22	0x160
Alternate_Ch_21	0x350	Primary_Ch_21	0x150
Alternate_Ch_20	0x340	Primary_Ch_20	0x140
Alternate_Ch_19	0x330	Primary_Ch_19	0x130
Alternate_Ch_18	0x320	Primary_Ch_18	0x120
Alternate_Ch_17	0x310	Primary_Ch_17	0x110
Alternate_Ch_16	0x300	Primary_Ch_16	0x100
Alternate_Ch_15	0x2F0	Primary_Ch_15	0x0F0
Alternate_Ch_14	0x2E0	Primary_Ch_14	0x0E0
Alternate_Ch_13	0x2D0	Primary_Ch_13	0x0D0
Alternate_Ch_12	0x2C0	Primary_Ch_12	0x0C0
Alternate_Ch_11	0x2B0	Primary_Ch_11	0x0B0
Alternate_Ch_10	0x2A0	Primary_Ch_10	0x0A0
Alternate_Ch_9	0x290	Primary_Ch_9	0x090
Alternate_Ch_8	0x280	Primary_Ch_8	0x080
Alternate_Ch_7	0x270	Primary_Ch_7	0x070
Alternate_Ch_6	0x260	Primary_Ch_6	0x060
Alternate_Ch_5	0x250	Primary_Ch_5	0x050
Alternate_Ch_4	0x240	Primary_Ch_4	0x040
Alternate_Ch_3	0x230	Primary_Ch_3	0x030
Alternate_Ch_2	0x220	Primary_Ch_2	0x020
Alternate_Ch_1	0x210	Primary_Ch_1	0x010
Alternate_Ch_0	0x200	Primary_Ch_0	0x000

Unused	0x00C
Control	0x008
Destination End Pointer	0x004
Source End Pointer	0x000

Рисунок 76 – Карта памяти для 32-х каналов, включая альтернативную структуру

Карта памяти, приведенная на рисунке , использует 1 Кбайт системной памяти. Контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть выровнен по границе 1 Кбайт – 0xXXXXXX000, далее 0xXXXXXX400, далее 0xXXXXXX800, далее 0xXXXXXXC00.

Существует возможность установить базовый адрес для первичной структуры управляющих данных путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит:

- от количества каналов, используемых в контроллере;
- от того, используется или нет альтернативная структура управляющих данных.

Таблица 60 перечисляет разряды адреса, обеспечивающие контроллеру доступ к различным элементам структуры управляющих данных.

Таблица 60 – Разряды адреса, соответствующие элементам структуры управляющих данных

Количество каналов, используемых в контроллере	Разряды адреса						
	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]
32	A	C[4]	C[3]	C[2]	C[1]	C[0]	0x0 0x4 0x8

Где А выбирает одну из структур управляющих данных канала:

- А = 0 выбирает первичную структуру управляющих данных;
- А = 1 выбирает альтернативную структуру управляющих данных.

C[x:0] Выбирает канал DMA.

Address[3:0] Выбирает один из управляющих элементов:

- 0x0 выбирает указатель конца данных источника;
- 0x4 выбирает указатель конца данных приемника;
- 0x8 выбирает конфигурацию управляющих данных;
- 0xC контроллер не имеет доступа к этому адресу. Если это необходимо, то возможно разрешить процессору использовать эти адреса в качестве системной памяти.

Примечание – Базовый адрес альтернативной структуры управляющих данных вычислять не обязательно, так как регистр alt_ctrl_base_ptr содержит эту информацию.

Таблица 61 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 61 – Разрешенные базовые адреса

Количество каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
32	0xFFFFFFFF000, 0xFFFFFFFF400, 0xFFFFFFFF800, 0xFFFFFFFFC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Эти 32-х разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA описывают следующие подразделы:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

7.13.5.1 Указатель конца данных источника

Область памяти под названием src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника.

Таблица 62 – Значения разрядов src_data_end_ptr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	src_data_end_ptr	Указатель последнего адреса данных источника

Перед тем, как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

7.13.5.2 Указатель конца данных приемника

Область памяти под названием `dst_data_end_ptr` содержит указатель на последний адрес месторасположения данных приемника.

Таблица 63 – Значения разрядов `dst_data_end_ptr`

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>dst_data_end_ptr</code>	Указатель на последний адрес данных приемника

Перед тем, как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

7.13.5.3 Разряды управления

Область памяти под названием `channel_cfg` обеспечивает управление каждой передачей DMA.

Таблица 64 – Значения разрядов `dst_data_end_ptr`

Номер	31	30	29	28	27	26	25	24	23...21	20...18	17...14	13...4	3	2...0
Доступ														
Сброс														
	<code>dst_inc[1:0]</code>		<code>dst_size[1:0]</code>		<code>src_inc[1:0]</code>		<code>src_size[1:0]</code>		<code>dst_prot_ctrl[2:0]</code>	<code>Src_prot_ctrl[2:0]</code>	<code>R_power[3:0]</code>	<code>n_minus_1[9:0]</code>	<code>next_useburst</code>	<code>cycle_ctrl[2:0]</code>

Таблица 65 объясняет назначение разрядов этой области памяти.

Таблица 65 – Назначение разрядов `channel_cfg`

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	<code>dst_inc[1:0]</code>	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово (16 разрядов); b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти <code>dst_data_end_ptr</code> . Разрядность данных источника = полуслово: b00 = зарезервировано; b01 = полуслово; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти <code>dst_data_end_ptr</code> . Разрядность данных источника = слово: b00 = зарезервировано;

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b01 = зарезервировано; b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти <code>dst_data_end_ptr</code>
29, 28	<code>dst_size[1:0]</code>	Размерность данных приемника <i>Примечание</i> – Значение этого поля должно быть равно значению поля <code>src_size</code>
27, 26	<code>src_inc[1:0]</code>	Шаг инкремента адреса источника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово; b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code> . Разрядность данных источника = полуслово: b00 = зарезервировано b01 = полуслово b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code> . Разрядность данных источника = слово: b00 = зарезервировано; b01 = зарезервировано; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code>
25, 24	<code>src_size[1:0]</code>	Задаёт размерность данных источника: b00 = байт; b01 = полуслово (в русском обычно слово); b10 = слово (в русском обычно двойное слово); b11 = зарезервировано
23...21	<code>dst_prot_ctrl[2:0]</code>	Задаёт состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд 23 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 22 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 21 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
20...18	<code>src_prot_ctrl[2:0]</code>	Задаёт состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд 20 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 19 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 18 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
17...14	<code>R_power[3:0]</code>	Задаёт количество передач DMA до выполнения контроллером процедуры арбитража. Возможные значения: b0000 - арбитраж производится после каждой передачи DMA; b0001 - арбитраж производится после 2 передач DMA; b0010 - арбитраж производится после 4 передач DMA;

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b0011 - арбитраж производится после 8 передач DMA; b0100 - арбитраж производится после 16 передач DMA; b0101 - арбитраж производится после 32 передач DMA; b0110 - арбитраж производится после 64 передач DMA; b0111 - арбитраж производится после 128 передач DMA; b1000 - арбитраж производится после 256 передач DMA; b1001 - арбитраж производится после 512 передач DMA; b1010 ÷ b1111 – арбитраж производится после 1024 передач DMA. Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024
13...4	n_minus_1[9:0]	Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл DMA. Необходимо установить эти разряды в значение, соответствующее размеру желаемого цикла DMA. 10-разрядное число плюс 1 задает количество передач DMA. Возможные значения: b0000000000 = 1 передача DMA; b0000000001 = 2 передачи DMA; b0000000010 = 3 передачи DMA; b0000000011 = 4 передачи DMA; b0000000100 = 5 передач DMA; b0000000101 = 6 передач DMA; b1111111111 = 1024 передачи DMA. Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA
3	next_useburst	Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «Исполнение с изменением конфигурации» и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные. <i>Примечание</i> – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в значение 0, если количество оставшихся передач DMA меньше, чем 2 ^R . Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C]. Если контроллер выполняет цикл DMA в режиме работы с периферией «Исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие данные, происходит следующее в зависимости от состояния next_useburst: 0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные. 1 - контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные.
2...0	cycle_ctrl[2:0]	Режим работы при выполнении цикла DMA: b000 Стоп . Означает, что структура управляющих данных является «неправильной»; b001 Основной . Контроллер должен получить новый запрос для окончания цикла DMA, перед этим

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		он должен выполнить процедуру арбитража; Авто-запрос. Контроллер автоматически осуществляет запрос на обработку по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла DMA;
		Пинг-понг. Контроллер выполняет цикл DMA используя одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллер выполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит <code>cycle_ctrl</code> поле в состоянии b001 или b 010;
		b100 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b100;
		b101 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b101;
		b110 Режим работы с периферией «исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b110;
		b111 Режим работы с периферией «исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b111

В начале цикла DMA или 2^R передачи DMA контроллер считывает значение `channel_cfg` из системной памяти. После выполнения $2R$ или N передач он сохраняет обновленное значение `channel_cfg` в системную память.

Контроллер не поддерживает значений `dst_size`, отличных от значений `src_size`. Если контроллер обнаруживает неравные значения этих полей, он использует значение `src_size` в качестве размера данных и приемника, и источника и при ближайшем обновлении поля `n_minus_1`, он также устанавливает значение поля `dst_size`, равное `src_size`.

После выполнения контроллером N передач, контроллер устанавливает значение поля `cycle_ctrl` в b000, делая тем самым `channel_cfg` данные «неправильными». Это позволяет избежать повторения выполненной передачи DMA.

7.13.5.4 Вычисление адреса

Для вычисления адреса источника передачи DMA, контроллер выполняет сдвиг влево значения `n_minus_1` на количество разрядов, соответствующее полю `src_inc`, и затем вычитает получившееся значение от значения указателя адреса конца данных источника. Подобным образом вычисляется адрес передатчика передачи DMA, контроллер выполняет сдвиг влево

значения n_minus_1 на количество разрядов, соответствующее полю dst_inc , и затем вычитает получившееся значение от значения указателя адреса конца данных приемника.

В зависимости от значения полей src_inc и dst_inc вычисления адресов приемника и источника выполняются по следующим уравнениям:

$src_inc=b00$ and $dst_inc=b00$

- адрес источника = $src_data_end_ptr - n_minus_1$
- адрес приемника = $dst_data_end_ptr - n_minus_1$.

$src_inc=b01$ and $dst_inc=b01$

- адрес источника = $src_data_end_ptr - (n_minus_1 \ll 1)$
- адрес приемника = $dst_data_end_ptr - (n_minus_1 \ll 1)$.

$src_inc=b01$ and $dst_inc=b10$

- адрес источника = $src_data_end_ptr - (n_minus_1 \ll 2)$
- адрес приемника = $dst_data_end_ptr - (n_minus_1 \ll 2)$.

$src_inc=b11$ and $dst_inc=b11$

- - адрес источника = $src_data_end_ptr$
- - адрес приемника = $dst_data_end_ptr$.

Таблица 66 перечисляет адреса приемника цикла DMA для 6 слов.

Таблица 66 – Цикла DMA для 6 слов с пословным инкрементом

Начальные значения channel_cfg перед циклом DMA				
$src\ size=b10, dst\ inc=b10, n\ minus\ 1=b101, cycle\ ctrl=1$				
DMA передачи	Указатель конца данных	Счетчик	Отличие ^{*)}	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
$src\ size=b10, dst\ inc=b10, n\ minus\ 1=0, cycle\ ctrl=0$				

* это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc .

Таблица 67 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 67 – Цикла DMA для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом DMA				
$src_size=b00, dst_inc=b01, n_minus_1=b1011, cycle_ctrl=1, R_power=b11$				
DMA передачи	Указатель конца данных	Счетчик	Отличие ^{*)}	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3

	0x5E7	9	0x12	0x5D5
	0x5E7	8	0x10	0x5D7
	0x5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Значения channel_cfg после 2^R передач DMA				
src_size=b00, dst_inc=b01, n_minus_1=b011, cycle_ctrl=1, R_power=b11				
DMA передачи	0x5E7	3	0x6	0x5E1
	0x5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=b00, dst_inc=b01, n_minus_1=0, cycle_ctrl=0**), R_power=b11				

* значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc.

** после окончания цикла DMA контроллер делает channel_cfg «неправильным», сбрасывая в 0 поле cycle_ctrl.

7.13.6 Описание регистров контроллера DMA

Данный раздел описывает регистры контроллера и управление контроллером через них. Раздел содержит следующие сведения:

- о регистровой модели контроллера;
- описание регистров.

Основные положения работы с регистровой модели контроллера:

- нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;
- необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;
- системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;
- все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 68 – Перечень регистров контроллера

Смещение отн. базового адреса	Наименование	Тип	Значение по сбросу	Описание
0xE004_2000	MDR_DMA			Контроллер DMA
0x000	STATUS	RO	0x-0nn0000*)	Статусный регистр DMA
0x004	CFG	WO	-	Регистр конфигурации DMA
0x008	CTRL_BASE_PTR.	R/W	0x00000000	Регистр базового адреса управляющих данных каналов
0x00C	ALT_CTRL_BASE_PTR	RO	0x000000nn**)	Регистр базового адреса альтернативных управляющих данных каналов
0x010	WAITONREQ_STATUS	RO	0x00000000	Регистр статуса ожидания запроса на обработку каналов
0x014	CHNL_SW_REQUEST	WO	-	Регистр программного запроса на обработку каналов
0x018	CHNL_USEBURST_SET	R/W	0x00000000	Регистр установки пакетного обмена каналов
0x01C	CHNL_USEBURST_CLR	WO	-	Регистр сброса пакетного обмена каналов
0x020	CHNL_REQ_MASK_SET	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
0x024	CHNL_REQ_MASK_CLR	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
0x028	CHNL_ENABLE_SET	R/W	0x00000000	Регистр установки разрешения каналов
0x02C	CHNL_ENABLE_CLR	WO	-	Регистр сброса разрешения каналов
0x030	CHNL_PRI_ALT_SET	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
0x034	CHNL_PRI_ALT_CLR	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
0x038	CHNL_PRIORITY_SET	R/W	0x00000000	Регистр установки приоритета каналов

Смещение отн. базового адреса	Наименование	Тип	Значение по сбросу	Описание
0x03C	CHNL_PRIORITY_CLR	WO	-	Регистр сброса приоритета каналов
0x040-0x048	-		-	Зарезервировано
0x04C	ERR_CLR	R/W	0x00000000	Регистр сброса флага ошибки
0x050	CHMUX0	R/W	0x00000000	Регистр назначения каналов 0,1,2,3
0x054	CHMUX1	R/W	0x00000000	Регистр назначения каналов 4,5,6,7
0x058	CHMUX2	R/W	0x00000000	Регистр назначения каналов 8,9,10,11
0x05C	CHMUX3	R/W	0x00000000	Регистр назначения каналов 12,13,14,15
0x060	CHMUX4	R/W	0x00000000	Регистр назначения каналов 16,17,18,19
0x064	CHMUX5	R/W	0x00000000	Регистр назначения каналов 20,21,22,23
0x068	CHMUX6	R/W	0x00000000	Регистр назначения каналов 24,25,26,27
0x06C	CHMUX7	R/W	0x00000000	Регистр назначения каналов 28,29,30,31
0x070-0xDFC	-	-		Зарезервировано

* значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от того, интегрирована ли схема тестирования.

** значение по сбросу зависит от количества каналов DMA, использованных в контроллере.

7.13.6.1 STATUS

Регистр STATUS является статусным регистром контроллера DMA.

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Таблица 70 перечисляет назначение разрядов регистра.

Таблица 69 – Статусный регистр DMA

Номер	31...28	27...21	20...16	15...8	7...4	3...1	0
Доступ	RO	U	RO	U	RO	U	RO
Сброс	0	0	0	0	0	0	0
	test_status	-	chnls_minus1	-	state	-	master_enable

Таблица 70 – Назначение разрядов регистра dma_status

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	test_status	Значение при чтении: 0x0 = контроллер не имеет интегрированной схемы тестирования; 0x1 = контроллер имеет интегрированную схему тестирования; 0x2 – 0xF = не определено
27...21	-	Не определено
20...16	chnls_minus1	Количество доступных каналов DMA минус 1. b11111 = контроллер имеет 32 канала DMA
15...8	-	Не определено
7...4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: b0000 = в покое; b0001 = чтение управляющих данных канала; b0010 = чтение указателя конца данных источника; b0011 = чтение указателя конца данных приемника; b0100 = чтение данных источника; b0101 = запись данных в приемник; b0110 = ожидание запроса на выполнение DMA; b0111 = запись управляющих данных канала; b1000 = приостановлен; b1001 = выполнен; b1010 = режим работы с периферией «Исполнение с изменением конфигурации»; b1011-b1111 = не определено
3...1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

7.13.6.2 CFG

Регистр CFG является регистром конфигурации контроллера DMA.

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

Таблица 72 перечисляет назначение разрядов регистра.

Таблица 71 – Регистр конфигурации DMA

Номер	31...8	7...5	4...1	0
Доступ	U	WO	U	WO
Сброс	0	0	0	0
	-	chnl_prot_ctrl	-	master_enable

Таблица 72 – Назначение разрядов регистра dma_cfg

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Не определено, следует записывать 0.
7...5	chnl_prot_ctrl	<p>Определяет уровни индикации сигналов HPROT[3:1] защиты шины АНВ-Lite:</p> <p>Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием;</p> <p>Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией;</p> <p>Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа.</p> <p><i>Примечания:</i></p> <p>1. Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1;</p> <p>2. Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0</p>
4...1	-	Не определено. Следует записывать 0.
0	master_enable	<p>Определяет состояние контроллера:</p> <p>0 – запрещает работу контроллера;</p> <p>1 – разрешает работу контроллера</p>

7.13.6.3 CTRL_BASE_PTR

Регистр CTRL_BASE_PTR является регистром базового адреса управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание – Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти, предназначенной контроллеру, зависит от количества каналов DMA, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов.

Значение, записываемое в регистр, должно быть выровнено по границе 1 Кбайт. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 74 перечисляет назначение разрядов регистра ctrl_base_ptr.

Таблица 73 – Регистр базового адреса управляющих данных каналов

Номер	31...10	9...0
Доступ	R/W	U
Сброс	0	0
	ctrl_base_ptr	-

Таблица 74 – Назначение разрядов регистра ctrl_base_ptr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	ctrl_base_ptr	Указатель на базовый адрес первичной структуры управляющих данных. См. соответствующий раздел
9...0	-	Не определено. Следует записывать 0

7.13.6.4 ALT_CTRL_BASE_PTR

Регистр ALT_CTRL_BASE_PTR является регистром базового адреса альтернативных управляющих данных каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов.

Таблица 76 перечисляет назначение разрядов регистра.

Таблица 75 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0
	Alt_ctrl_base_ptr

Таблица 76 – Назначение разрядов регистра alt_ctrl_base_ptr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	alt_ctrl_base_ptr	Указатель базового адреса альтернативной структуры управляющих данных

7.13.6.5 WAITONREQ_STATUS

Регистр WAITONREQ_STATUS является регистром статуса ожидания запроса на обработку каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 78 перечисляет назначение разрядов регистра.

Таблица 77 – Регистр статуса ожидания запроса на обработку каналов

Номер	31	2	1	0
Номер	RO	RO	RO	RO
Доступ	0	0	0	0
	dma_waitonreq_status for dma_waitnreg [31]	dma_waitonreq_status for dma_waitnreg [2]	dma_waitonreq_status for dma_waitnreg [1]	dma_waitonreq_status for dma_waitnreg [0]

Таблица 78 – Назначение разрядов регистра dma_waitonreq_status

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов DMA. При чтении: Разряд [C] =0 означает, что dma_waitonreq[C] в состоянии 0 Разряд [C] =1 означает, что dma_waitonreq[C] в состоянии 1

7.13.6.6 CHNL_SW_REQUEST

Регистр CHNL_SW_REQUEST является регистром программного запроса на обработку каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA.

Таблица 80 перечисляет назначение разрядов регистра.

Таблица 79 – Регистр программного запроса на обработку каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_sw_request for channel [31]	chnl_sw_request for channel [2]	chnl_sw_request for channel [1]	chnl_sw_request for channel [0]

Таблица 80 – Назначение разрядов регистра chnl_sw_request

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_sw_request	Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла DMA по соответствующему каналу DMA. При записи: Разряд [C] = 0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен; Разряд [C] = 1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен. Запись разряда, соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен

7.13.6.7 CHNL_USEBURST_SET

Регистр CHNL_USEBURST_SET является регистром установки пакетного обмена каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена

Таблица 82 перечисляет назначение разрядов регистра.

Таблица 81 – Регистр установки пакетного обмена каналов

Номер	31	2	1	0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	chnl_useburst_set for channel [31]	chnl_useburst_set for channel [2]	chnl_useburst_set for channel [1]	chnl_useburst_set for channel [0]

Таблица 82 – Назначение разрядов регистра chnl_useburst_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] =0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2 ^R передач. Разряд [C] =1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2 ^R передач. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0; Разряд [C] =1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2 ^R передач. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

После выполнения предпоследней передачи из 2^R передач, в том случае, если число оставшихся передач (N) меньше чем 2^R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание – При программировании channel_cfg значением N меньшим, чем 2^R, запрещена установка соответствующего разряда chnl_useburst_set в случае, если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1 после окончания цикла DMA, использующего альтернативные управляющие данные.

7.13.6.8 CHNL_USEBURST_CLR

Регистр CHNL_USEBURST_CLR является регистром сброса пакетного обмена каналов. Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[].

Таблица 84 перечисляет назначение разрядов регистра chnl_useburst_clr.

Таблица 83 – Регистр сброса пакетного обмена каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_useburst_clr for channel [31]	chnl_useburst_clr for channel [2]	chnl_useburst_clr for channel [1]	chnl_useburst_clr for channel [0]

Таблица 84 – Назначение разрядов регистра chnl_useburst_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от dma_sreq[]. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]; Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.13.6.9 CHNL_REQ_MASK_SET

Регистр CHNL_REQ_MASK_SET является регистром маскирования запросов на обслуживание каналов

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[]. Регистр возвращает при чтении состояние установок маскирования запросов от dma_sreq[] и dma_req[] на обслуживание каналов.

Таблица 86 перечисляет назначение разрядов регистра chnl_req_mask_set.

Таблица 85 – Регистр маскирования запросов на обслуживание каналов

Номер	31	2	1	0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	chnl_req_mask_set for dma_reg [31] and dma_sreg [31]	chnl_req_mask_set for dma_reg [2] and dma_sreg [2]	chnl_req_mask_set for dma_reg [1] and dma_sreg [1]	chnl_req_mask_set for dma_reg [0] and dma_sreg [0]

Таблица 86 – Назначение разрядов регистра chnl_req_mask_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_req_mask_set	Отключает обработку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на поступающие запросы; Разряд [C] = 1 означает, что канал DMA C не выполняет циклы DMA в ответ на поступающие запросы. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_clr регистр для разрешения установки запросов; Разряд [C] = 1 отключает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.13.6.10 CHNL_REQ_MASK_CLR

Регистр CHNL_REQ_MASK_CLR является регистром очистки маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[].

Таблица 88 перечисляет назначение разрядов регистра chnl_req_mask_clr.

Таблица 87 – Регистр очистки маскирования запросов на обслуживание каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_req_mask_clr for dma_reg [31] and dma_sreg [31]	chnl_req_mask_clr for dma_reg [2] and dma_sreg [2]	chnl_req_mask_clr for dma_reg [1] and dma_sreg [1]	chnl_req_mask_clr for dma_reg [0] and dma_sreg [0]

Таблица 88 – Назначение разрядов регистра chnl_req_mask_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_req_mask_clr	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов; Разряд [C] =1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.13.6.11 CHNL_ENABLE_SET

Регистр CHNL_ENABLE_SET является регистром установки разрешения каналов.

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA.

Таблица 90 перечисляет назначение разрядов регистра chnl_enable_set.

Таблица 89 – Регистр установки разрешения каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_enable_set for channel [31]	chnl_enable_set for channel [2]	chnl_enable_set for channel [1]	chnl_enable_set for channel [0]

Таблица 90 – Назначение разрядов регистра chnl_enable_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_enable_set	<p>Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек.</p> <p>При чтении: Разряд [C] = 0 означает, что канал DMA C отключен; Разряд [C] = 1 означает, что работа канала DMA C разрешена.</p> <p>При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала; Разряд [C] = 1 разрешает работу канала DMA C.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

7.13.6.12 CHNL_ENABLE_CLR

Регистр CHNL_ENABLE_CLR является регистром сброса разрешения каналов. Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA. Таблица 92 перечисляет назначение разрядов регистра chnl_enable_clr.

Таблица 91 – Регистр сброса разрешения каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_enable_clr for channel 31	chnl_enable_clr for channel 2	chnl_enable_clr for channel 1	chnl_enable_clr for channel 0

Таблица 92 – Назначение разрядов регистра chnl_enable_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_enable_clr	<p>Установка соответствующего разряда запрещает работу соответствующего канала DMA.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала;</p> <p>Разряд [C] = 1 запрещает работу канала DMA C.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p><i>Примечание</i> – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях:</p> <ul style="list-style-type: none"> - при завершении цикла DMA; - при чтении из channel_cfg с полем cycle_ctrl установленным в b000; - при появлении ошибки на шине AHB-Lite

7.13.6.13 CHNL_PRI_ALT_SET

Регистр CHNL_PRI_ALT_SET является регистром установки первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала DMA с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов DMA (какую структуру управляющих данных использует каждый канал DMA).

Таблица 94 перечисляет назначение разрядов регистра chnl_pri_alt_set.

Таблица 93 – Регистр установки первичной/альтернативной структуры управляющих данных каналов

Номер	31	2	1	0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	chnl_pri_alt_set for channel [31]	chnl_pri_alt_set for channel [2]	chnl_pri_alt_set for channel [1]	chnl_pri_alt_set for channel [0]

Таблица 94 – Назначение разрядов регистра chnl_pri_alt_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_pri_alt_set	<p>Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек.</p> <p>При чтении:</p> <p>Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных;</p> <p>Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0;</p> <p>Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p><i>Примечание</i> – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях:</p> <ul style="list-style-type: none"> - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: <ul style="list-style-type: none"> - «пинг-понг»; - работа с памятью «Исполнение с изменением конфигурации»; - работа с периферией «Исполнение с изменением конфигурации»

7.13.6.14 CHNL_PRI_ALT_CLR

Регистр CHNL_PRI_ALT_CLR является регистром сброса первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных.

Таблица 96 перечисляет назначение разрядов регистра chnl_pri_alt_clr.

Таблица 95 – Регистр сброса первичной/альтернативной структуры управляющих данных каналов

Номер	31	2	1	0
Доступ	WO	WO	WO	WO
Сброс	0	0	0	0
	chnl_pri_alt_clr for channel [31]	chnl_pri_alt_clr for channel [2]	chnl_pri_alt_clr for channel [1]	chnl_pri_alt_clr for channel [0]

Таблица 96 – Назначение разрядов регистра chnl_pri_alt_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_pri_alt_clr	<p>Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA.</p> <p>При записи:</p> <ul style="list-style-type: none"> Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных; Разряд [C] = 1 подключает использование первичной структуры управляющих данных каналом DMA C. <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p><i>Примечание</i> – Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях:</p> <ul style="list-style-type: none"> - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: <ul style="list-style-type: none"> - «пинг-понг»; - работа с памятью «Исполнение с изменением конфигурации»; - работа с периферией «Исполнение с изменением конфигурации»

7.13.6.15 CHNL_PRIORITY_SET

Регистр CHNL_PRIORITY_SET является регистром установки приоритета каналов. Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA. Таблица 98 перечисляет назначение разрядов регистра chnl_priority_set.

Таблица 97 – Регистр установки приоритета каналов

Номер	31	2	1	0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	chnl_priorit_set for channel [31]	chnl_priority_set for channel [2]	chnl_priority_set for channel [1]	chnl_priority_set for channel [0]

Таблица 98 – Назначение разрядов регистра chnl_priority_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_priority_set	<p>Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA.</p> <p>При чтении:</p> <ul style="list-style-type: none"> Разряд [C] = 0 означает, что каналу DMA с присвоен уровень приоритета по умолчанию; Разряд [C] = 1 означает, что каналу DMA с присвоен высокий уровень приоритета. <p>При записи:</p> <ul style="list-style-type: none"> Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета. <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

7.13.6.16 CHNL_PRIORITY_CLR

Регистр CHNL_PRIORITY_CLR является регистром сброса приоритета каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.

Таблица 100 перечисляет назначение разрядов регистра chnl_priority_clr.

Таблица 99 – Регистр сброса приоритета каналов

Номер	31		2	1	0
Доступ	WO		WO	WO	WO
Сброс	0		0	0	0

chnl_prioirit_clr for channel [31]		chnl_priority_clr for channel [2]	chnl_priority_clr for channel [1]	chnl_priority_clr for channel [0]
------------------------------------	-------	--	-----------------------------------	-----------------------------------	-----------------------------------

Таблица 100 – Назначение разрядов регистра chnl_priority_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
[31:0]	chnl_priority_clr	Установка разряда присваивает соответствующему каналу DMA уровень приоритета по умолчанию. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_set регистр для установки каналу С высокого уровня приоритета. Разряд [C] = 1 устанавливает каналу DMA С уровень приоритета по умолчанию. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.13.6.17 ERR_CLR

Регистр ERR_CLR является регистром сброса флага ошибки.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err.

Таблица 102 перечисляет назначение разрядов регистра err_clr.

Таблица 101 – Регистр сброса флага ошибки

Номер	31...1	0
Доступ	U	R/W
Сброс	0	0
	-	err_clr

Таблица 102 – Назначение разрядов регистра err_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...1	-	Не определено. Следует записывать 0
0	err_clr	<p>Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err.</p> <p>При чтении: Разряд [C] = 0 означает, что dma_err находится в состоянии 0; Разряд [C] = 1 означает, что dma_err находится в состоянии 1.</p> <p>При записи: Разряд [C] = 0 не дает эффекта. Состояние dma_err останется неизменным; Разряд [C] = 1 сбрасывает сигнал (флаг) dma_err в состояние 0.</p> <p><i>Примечание</i> – При сбросе сигнала dma_err одновременно с появлением ошибки на шине АНВ-Lite, то приоритет отдается ошибке, и, следовательно, значение регистра (и dma_err) останется неизменным (несброшенным)</p>

7.13.6.18 CHMUX[x]

Base ADDR=	0xE004_2000				Offset=	0x050 0x054 0x058 0x05C 0x060 0x064 0x068 0x06C									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANNEL[x*4+3]								CHANNEL[x*4+2]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNEL[x*4+1]								CHANNEL[x*4+0]							

Бит	Имя	Значение	Описание
31...24	CHANNEL[x*4+3]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+3
23...16	CHANNEL[x*4+2]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+2
15...8	CHANNEL[x*4+1]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+1
7...0	CHANNEL[x*4+0]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+0

Таблица 103 – Виртуальные номера запросов DMA

Номера реальных каналов DMA	Тип SREQ	Тип REQ	Описание
0	ETH0_EVENT[0]	-	Запрос от ETH0
1	ETH0_EVENT[1]	-	Запрос от ETH0
2	-	-	Зарезервировано
3	-	-	Зарезервировано
4	SPW0_RX_SREQ	SPW0_RX_BREQ	Запрос приемника SPW0
5	-	-	Зарезервировано
6	SPW0_TX_SREQ	SPW0_TX_BREQ	Запрос передатчика SPW0
7	-	-	Зарезервировано
8	TMR0_REQ	-	Запрос REQ от Таймера 0
9	TMR1_REQ	-	Запрос REQ от Таймера 1
10	TMR2_REQ	-	Запрос REQ от Таймера 2
11	TMR3_REQ	-	Запрос REQ от Таймера 3
12	-	-	Зарезервировано
13	-	-	Зарезервировано
14	TMR0_REQ1	-	Запрос REQ1 от Таймера 0
15	TMR1_REQ1	-	Запрос REQ1 от Таймера 1
16	TMR2_REQ1	-	Запрос REQ1 от Таймера 2

Номера реальных каналов DMA	Тип SREQ	Тип REQ	Описание
17	TMR3_REQ1	-	Запрос REQ1 от Таймера 3
18	-	-	Зарезервировано
19	-	-	Зарезервировано
20	TMR0_REQ2	-	Запрос REQ2 от Таймера 0
21	TMR1_REQ2	-	Запрос REQ2 от Таймера 1
22	TMR2_REQ2	-	Запрос REQ2 от Таймера 2
23	TMR3_REQ2	-	Запрос REQ2 от Таймера 3
24	-	-	Зарезервировано
25	-	-	Зарезервировано
26	TMR0_REQ3	-	Запрос REQ3 от Таймера 0
27	TMR1_REQ3	-	Запрос REQ3 от Таймера 1
28	TMR2_REQ3	-	Запрос REQ3 от Таймера 2
29	TMR3_REQ3	-	Запрос REQ3 от Таймера 3
30	-	-	Зарезервировано
31	-	-	Зарезервировано
32	TMR0_REQ4	-	Запрос REQ4 от Таймера 0
33	TMR1_REQ4	-	Запрос REQ4 от Таймера 1
34	TMR2_REQ4	-	Запрос REQ4 от Таймера 2
35	TMR3_REQ4	-	Запрос REQ4 от Таймера 3
36	-	-	Зарезервировано
37	-	-	Зарезервировано
38	SSP0_TX_SREQ	SSP0_TX_BREQ	Запрос передатчика SSP0
39	-	-	Зарезервировано
40	-	-	Зарезервировано
41	-	-	Зарезервировано
42	SSP0_RX_SREQ	SSP0_RX_BREQ	Запрос приемника SSP0
43	-	-	Зарезервировано
44	-	-	Зарезервировано
45	-	-	Зарезервировано
46	UART0_TX_SREQ	UART0_TX_BREQ	Запрос передатчика UART0
47	UART1_TX_SREQ	UART1_TX_BREQ	Запрос передатчика UART1
48	-	-	Зарезервировано
49	-	-	Зарезервировано
50	UART0_RX_SREQ	UART0_RX_BREQ	Запрос приемника UART0
51	UART1_RX_SREQ	UART1_RX_BREQ	Запрос приемника UART1
52	-	-	Зарезервировано
53	-	-	Зарезервировано
54	ADC0_SREQ0	-	Запрос REQ0 от ADC0
55	ADC0_SREQ1	-	Запрос REQ1от ADC0
56	ADC0_SREQ2	-	Запрос REQ2от ADC0
57	ADC0_SREQ3	-	Запрос REQ3от ADC0
58	ADC0_SREQ4	-	Запрос REQ4от ADC0
59	ADC0_SREQ5	ADC0_BREQ5	Запрос REQ5от ADC0
60	ADC1_SREQ0	-	Запрос REQ0 от ADC1
61	ADC1_SREQ1	-	Запрос REQ1от ADC1
62	ADC1_SREQ2	-	Запрос REQ2от ADC1
63	ADC1_SREQ3	-	Запрос REQ3от ADC1
64	ADC1_SREQ4	-	Запрос REQ4от ADC1
65	ADC1_SREQ5	ADC1_BREQ5	Запрос REQ5от ADC1
66	DAC0_SREQ	DAC0_BREQ	Запрос REQ от DAC0
67	DAC1_SREQ	DAC1_BREQ	Запрос REQ от DAC1
68	-	-	Зарезервировано
69	-	-	Зарезервировано
70	-	-	Зарезервировано
71	-	-	Зарезервировано
72	-	-	Зарезервировано
73	-	-	Зарезервировано

Номера реальных каналов DMA	Тип SREQ	Тип REQ	Описание
74	-	-	Зарезервировано
75	-	-	Зарезервировано
76	-	-	Зарезервировано
77	-	-	Зарезервировано
78	-	-	Зарезервировано
79	-	-	Зарезервировано
80	-	-	Зарезервировано
81	-	-	Зарезервировано
82	-	-	Зарезервировано
83	-	-	Зарезервировано
84	-	-	Зарезервировано
85	-	-	Зарезервировано
86	-	-	Зарезервировано
87	-	-	Зарезервировано
88	-	-	Зарезервировано
89	-	-	Зарезервировано
90	-	-	Зарезервировано
91	-	-	Зарезервировано
92	-	-	Зарезервировано
93	-	-	Зарезервировано
94	-	-	Зарезервировано
95	-	-	Зарезервировано

7.14 Контроллер портов ввода-вывода (PORTx)

Порты ввода-вывода предназначены для приема и формирования внешних электрических сигналов. Для каждого вывода микроконтроллера может быть назначена одна из 16-ти функций. В качестве функции вывода выступает использование данного вывода как ввода-вывода сигналов периферийных контроллеров, таких как внешняя шина, UART, SSP и так далее, либо полностью программное управление – пользовательский порт. Для некоторых выводов может быть назначена аналоговая функция, например, канал ввода аналогового сигнала в АЦП или выдачи сигнала с ЦАП.

Порты ввода-вывода могут использоваться пользовательскими задачами без дополнительного программного драйвера. Для того, чтобы избежать взаимовлияния работы различных задач друг на друга, например, при возникновении перехода с одной задачи на другую в момент выполнения операции чтения-модификации-записи, все регистры порта имеют виртуальный регистр установки битов в 1 и виртуальный регистр сброса битов в 0. Таким образом, чтобы установить какой-либо бит регистра в 1 нет необходимости выполнять операции чтения и модификации, достаточно записать 1 в соответствующий бит регистра с префиксом “S”, и это взведет соответствующий бит. Для сброса этого бита необходимо записать 1 в регистр с префиксом “C”. Запись 0 в биты никакого эффекта не несет. При чтении из всех регистров, кроме RXTX, возвращается состояние триггеров. При чтении регистра RXTX возвращается значение с выводов микросхемы.

Каждый вывод порта может служить источником прерываний от портов. Схема формирования запроса прерывания представлена на рисунке ниже (Рисунок 77).

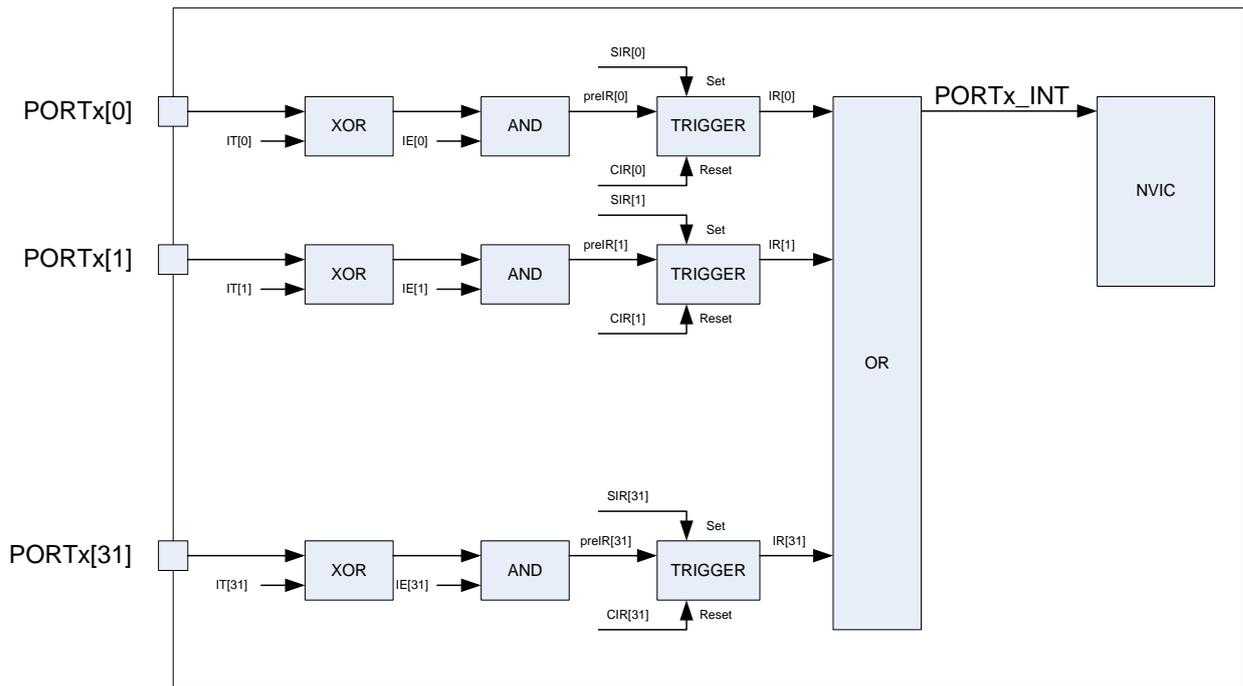


Рисунок 77 – Схема формирования запроса прерывания

В регистре IR запроса прерывания порта факт прерывания фиксируется только при разрешении запроса. При этом запрещение запроса прерывания IE не сбрасывает флаг IR, если он появился до запрета. Сброс запроса прерывания возможен только сбросом соответствующего разряда регистра IR. Формирование прерываний, их обработка и сброс возможны только при включенном тактировании порта.

Для портов ввода-вывода реализована защита от «перегрузки». Если вывод выдает логическую “1”, но на линии уровень сигнала соответствует логическому “0”, вырабатывается флаг защиты от перегрузки HCUR. При выдаче логического “0” механизм защиты аналогичен. При возникновении флага защиты может быть сгенерировано прерывание от порта. Для исправления ситуации необходимо программно перевести вывод в третье состояние или, в

случае, когда перегрузки кратковременны, сбросить флаг записью 1 в соответствующий бит регистра HCUR.

7.14.1 Описание регистров

Таблица 104 – Описание регистров контроллера портов ввода-вывода

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
0x4008_0000				
0x4008_1000				
0x4008_2000				
0x4008_3000				
0x4008_4000				
	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	RXTX		Регистр данных порта
	0x0000_0008	SRXTX		Регистр установки порта
	0x0000_000C	CRXTX		Регистр сброса порта
	0x0000_0010	SOE		Регистр установки направления вывода порта
	0x0000_0014	COE		Регистр сброса направления вывода порта
	0x0000_0018	SFUNC0		Регистр установки функции вывода порта (7-0)
	0x0000_001C	SFUNC1		Регистр установки функции вывода порта (15-8)
	0x0000_0020	SFUNC2		Регистр установки функции вывода порта (23-16)
	0x0000_0024	SFUNC3		Регистр установки функции вывода порта (31-24)
	0x0000_0028	CFUNC0		Регистр сброса функции вывода порта (7-0)
	0x0000_002C	CFUNC1		Регистр сброса функции вывода порта (15-8)
	0x0000_0030	CFUNC2		Регистр сброса функции вывода порта (23-16)
	0x0000_0034	CFUNC3		Регистр сброса функции вывода порта (31-24)
	0x0000_0038	SANALOG		Регистр установки типа вывода порта
	0x0000_003C	CANALOG		Регистр сброса типа вывода порта
	0x0000_0040	SPULLUP		Регистр установки подтяжки к питанию вывода порта
	0x0000_0044	CPULLUP		Регистр сброса подтяжки к питанию вывода порта
	0x0000_0048	SPULLDOWN		Регистр установки подтяжки к земле вывода порта
	0x0000_004C	CPULLDOWN		Регистр сброса подтяжки к земле вывода порта
	0x0000_0050	SPD		Регистр установки типа драйвера вывода порта
	0x0000_0054	CPD		Регистр сброса типа драйвера вывода порта
	0x0000_0058	SPWR0		Регистр установки скорости вывода порта (15-0)
	0x0000_005C	SPWR1		Регистр установки скорости вывода порта (31-16)
	0x0000_0060	CPWR0		Регистр сброса скорости вывода порта (15-0)
	0x0000_0064	CPWR1		Регистр сброса скорости вывода порта (31-16)
	0x0000_0068	SCL		Регистр установки ограничения по току вывода порта
	0x0000_006C	CCL		Регистр сброса ограничения по току вывода порта
	0x0000_0070	SIE		Регистр установки разрешения прерывания от вывода
	0x0000_0074	CIE		Регистр сброса разрешения прерывания от вывода
	0x0000_0078	SIT		Регистр установки активного уровня прерывания
	0x0000_007C	CIT		Регистр сброса активного уровня прерывания
	0x0000_0080	SIR		Регистр установки флага прерывания
	0x0000_0084	CIR		Регистр сброса флага прерывания
	0x0000_0088	HCUR		Регистр флага превышения выходного тока

7.14.1.1 KEY

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0000												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name: KEY															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока

7.14.1.2 RXTX

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0004												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXTX[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXTX[15:0]															

Бит	Имя	Значение	Описание
31...0	RXTX	0x00000000	<p>Регистр прямого доступа данных порта При чтении отображает состояние на выводах порта. При записи осуществляет запись в выходные триггера порта RXTX[0] соответствует выводу порта с номером 0 RXTX[1] соответствует выводу порта с номером 1 и так далее.</p> <p>Внимание! При выполнении операции чтение-модификация-запись возможно возникновения ошибок, например: - вывод порта настроен на выдачу высокого уровня (в выходном триггере 1), но при этом нагрузка на выводе такая, что уровень стал ниже нижней границы высокого уровня, в этом случае при чтении регистра RXTX может быть считан 0. И при его последующей записи измениться значение в выходном триггере с 1 на 0. - при выполнении операции чтение-модификации-записи произошло переключение задачи, либо уход на обработку прерывания сразу после выполнения чтения. В другой задаче или обработчике прерывания изменяется значение в выходных триггерах и осуществляется возврат в текущую задачу или выход из обработчика. И выполняется вторая часть модификация и запись, но при этом эти действия выполняются над данными полученными до ухода и по их завершению могут быть изменены данные записанные в другой задаче, либо обработчике прерываний. Для избегания такого рода проблем рекомендуется использовать биты установки и сброса значений, например, SRXTX и CRXTX и другие аналогичные.</p>

7.14.1.3 SRXTX

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0008												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRXTX[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRXTX[15:0]															

Бит	Имя	Значение	Описание
31...0	SRXTX	0x00000000	<p>Регистр установки порта</p> <p>При чтении отображает состояние выходного триггеров порта. При записи осуществляет запись в выходные триггера порта SRXTX[0] соответствует выводу порта с номером 0 SRXTX[1] соответствует выводу порта с номером 1 и так далее.</p> <p>При записи 1 производится установка в 1 выходного триггера порта, при записи 0, значение выходного триггера не изменяется.</p> <p>Внимание! При чтении из регистра возвращается значение, записанное в выходные триггера, но при этом, если вывод, например, не настроен на выход, считанное значение может отличаться от реального значения на соответствующем выводе порта</p>

7.14.1.4 CRXTX

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_000C												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRXTX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRXTX[15:0]															

Бит	Имя	Значение	Описание
31...0	CRXTX	0x00000000	<p>Регистр сброса порта</p> <p>При чтении отображает состояние выходного триггеров порта. При записи осуществляет запись в выходные триггера порта CRXTX[0] соответствует выводу порта с номером 0 CRXTX[1] соответствует выводу порта с номером 1 и так далее.</p> <p>При записи 1 производится установка в 0 выходного триггера порта, при записи 0, значение выходного триггера не изменяется.</p> <p>Внимание! При чтении из регистра возвращается значение, записанное в выходные триггера, но при этом, если вывод, например, не настроен на выход, считанное значение может отличаться от реального значения на соответствующем выводе порта</p>

7.14.1.5 SOE

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0010												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE[15:0]															

Бит	Имя	Значение	Описание
31...0	SOE[31:0]	0x00000000	<p>Регистр установки направления вывода порта</p> <p>При записи 1 производится установка в 1 сигнала направления порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SOE[0] соответствует выводу порта с номером 0 SOE[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – вывод работает на вход 1 – вывод работает на выход</p> <p>Значение направления, задаваемое регистром SOE и COE, имеет смысл при функционировании порта в режиме пользовательского вывода FUNC= 0000 При выборе других функций для вывода, его направление задается в соответствии с выбранной функцией автоматически</p>

7.14.1.6 СОЕ

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0014												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
СОЕ[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
СОЕ[15:0]															

Бит	Имя	Значение	Описание
31...0	СОЕ[31:0]	0x00000000	<p>Регистр установки направления вывода порта</p> <p>При записи 1 производится установка в 0 сигнала направления порта, при записи 0, значение выходного триггера не изменяется.</p> <p>СОЕ[0] соответствует выводу порта с номером 0 СОЕ[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – вывод работает на вход 1 – вывод работает на выход</p> <p>Значение направления, задаваемое регистром SOE и СОЕ, имеет смысл при функционировании порта в режиме пользовательского вывода FUNC= 0000. При выборе других функций для вывода, его направление задается в соответствии с выбранной функцией автоматически.</p>

7.14.1.7 SFUNCx

Base ADDR=	0x4008_0000	Offset=	0x0000_0018												
	0x4008_1000		0x0000_001C												
	0x4008_2000		0x0000_0020												
	0x4008_3000		0x0000_0024												
	0x4008_4000														
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FUNC7[3:0]				FUNC6[3:0]				FUNC5[3:0]				FUNC4[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUNC3[3:0]				FUNC2[3:0]				FUNC1[3:0]				FUNC0[3:0]			

Бит	Имя	Значение	Описание
31...28	FUNC7[3:0]	0x0	Функция, выполняемая выводом При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0000 – пользовательский вывод 0001 – Функция 1 ... 1111 – Функция 15 Конкретное назначение функции смотрите в описании выводов микросхемы. Выбор аналоговой функции приоритета. SFUNC0[3...0] – вывод 0 SFUNC0[7...4] – вывод 1 SFUNC0[11...8] – вывод 2 SFUNC0[15...12] – вывод 3 SFUNC0[19...16] – вывод 4 SFUNC0[23...20] – вывод 5 SFUNC0[27...24] – вывод 6 SFUNC0[31...28] – вывод 7 SFUNC1[3...0] – вывод 8 SFUNC1[7...4] – вывод 9 SFUNC1[11...8] – вывод 10 SFUNC1[15...12] – вывод 11 SFUNC1[19...16] – вывод 12 SFUNC1[23...20] – вывод 13 SFUNC1[27...24] – вывод 14 SFUNC1[31...28] – вывод 15 SFUNC2[3...0] – вывод 16 SFUNC2[7...4] – вывод 17 SFUNC2[11...8] – вывод 18 SFUNC2[15...12] – вывод 19 SFUNC2[19...16] – вывод 20 SFUNC2[23...20] – вывод 21 SFUNC2[27...24] – вывод 22 SFUNC2[31...28] – вывод 23 SFUNC3[3...0] – вывод 24 SFUNC3[7...4] – вывод 25 SFUNC3[11...8] – вывод 26 SFUNC3[15...12] – вывод 27 SFUNC3[19...16] – вывод 28 SFUNC3[23...20] – вывод 29 SFUNC3[27...24] – вывод 30 SFUNC3[31...28] – вывод 31
27...24	FUNC6[3:0]	0x0	
23...20	FUNC5[3:0]	0x0	
19...16	FUNC4[3:0]	0x0	
15...12	FUNC3[3:0]	0x0	
11...8	FUNC2[3:0]	0x0	
7...4	FUNC1[3:0]	0x0	
3...0	FUNC0[3:0]	0x0	

7.14.1.8 CFUNCx

Base ADDR=	0x4008_0000	Offset=	0x0000_0028												
	0x4008_1000		0x0000_002C												
	0x4008_2000		0x0000_0030												
	0x4008_3000		0x0000_0034												
	0x4008_4000														
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FUNC7[3:0]				FUNC6[3:0]				FUNC5[3:0]				FUNC4[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUNC3[3:0]				FUNC2[3:0]				FUNC1[3:0]				FUNC0[3:0]			

Бит	Имя	Значение	Описание
31...28	FUNC7[3:0]	0x0	Функция, выполняемая выводом. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0000 – пользовательский вывод 0001 – Функция 1 ... 1111 – Функция 15 Конкретное назначение функции смотрите в описании выводов микросхемы. CFUNC0[3...0] – вывод 0 CFUNC0[7...4] – вывод 1 CFUNC0[11...8] – вывод 2 CFUNC0[15...12] – вывод 3 CFUNC0[19...16] – вывод 4 CFUNC0[23...20] – вывод 5 CFUNC0[27...24] – вывод 6 CFUNC0[31...28] – вывод 7 CFUNC1[3...0] – вывод 8 CFUNC1[7...4] – вывод 9 CFUNC1[11...8] – вывод 10 CFUNC1[15...12] – вывод 11 CFUNC1[19...16] – вывод 12 CFUNC1[23...20] – вывод 13 CFUNC1[27...24] – вывод 14 CFUNC1[31...28] – вывод 15 CFUNC2[3...0] – вывод 16 CFUNC2[7...4] – вывод 17 CFUNC2[11...8] – вывод 18 CFUNC2[15...12] – вывод 19 CFUNC2[19...16] – вывод 20 CFUNC2[23...20] – вывод 21 CFUNC2[27...24] – вывод 22 CFUNC2[31...28] – вывод 23 CFUNC3[3...0] – вывод 24 CFUNC3[7...4] – вывод 25 CFUNC3[11...8] – вывод 26 CFUNC3[15...12] – вывод 27 CFUNC3[19...16] – вывод 28 CFUNC3[23...20] – вывод 29 CFUNC3[27...24] – вывод 30 CFUNC3[31...28] – вывод 31
27...24	FUNC6[3:0]	0x0	
23...20	FUNC5[3:0]	0x0	
19...16	FUNC4[3:0]	0x0	
15...12	FUNC3[3:0]	0x0	
11...8	FUNC2[3:0]	0x0	
7...4	FUNC1[3:0]	0x0	
3...0	FUNC0[3:0]	0x0	

7.14.1.9 SANALOG

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0038												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SANALOG[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SANALOG [15:0]															

Бит	Имя	Значение	Описание
31...0	SANALOG[31:0]	0x00000000	<p>Регистр установки типа вывода порта</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SANALOG[0] соответствует выводу порта с номером 0 SANALOG[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – вывод выполняет аналоговую функцию 1 – вывод выполняет цифровую функцию</p> <p>При выборе аналоговой функции выключается вход вывода микросхемы, и из RXTX всегда будет считана 1 в соответствующем разряде.</p>

7.14.1.10 CANALOG

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_003C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CANALOG[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CANALOG [15:0]															

Бит	Имя	Значение	Описание
31...0	CANALOG[31:0]	0x00000000	<p>Регистр сброса типа вывода порта</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>CANALOG[0] соответствует выводу порта с номером 0 CANALOG[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – вывод выполняет аналоговую функцию 1 – вывод выполняет цифровую функцию</p> <p>При выборе аналоговой функции выключается вход вывода микросхемы, и из RXTX всегда будет считана 1 в соответствующем разряде</p>

7.14.1.11 SPULLUP

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0040												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPULLUP[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPULLUP [15:0]															

Бит	Имя	Значение	Описание
31...0	SPULLUP [31:0]	0x00000000	<p>Регистр установки подтяжки к питанию вывода порта</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SPULLUP[0] соответствует выводу порта с номером 0 SPULLUP[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – нет подтяжки 1 – есть подтяжка к питанию Ucc через резистор R_{PULLUP} с типовым значением 50 кОм</p>

7.14.1.12 CPULLUP

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0044												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPULLUP[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPULLUP [15:0]															

Бит	Имя	Значение	Описание
31...0	CPULLUP [31:0]	0x00000000	<p>Регистр сброса подтяжки к питанию вывода порта</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SPULLUP[0] соответствует выводу порта с номером 0 SPULLUP[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – нет подтяжки 1 – есть подтяжка к питанию Ucc через резистор R_{PULLUP} с типовым значением 50 кОм</p>

7.14.1.13 SPULLDOWN

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0048												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPULLDOWN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPULLDOWN [15:0]															

Бит	Имя	Значение	Описание
31...0	SPULLDOWN [31:0]	0x00000000	<p>Регистр установки подтяжки к земле вывода порта</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SPULLDOWN[0] соответствует выводу порта с номером 0 SPULLDOWN[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – нет подтяжки 1 – есть подтяжка к земле через резистор R_{PULLDOWN} с типовым значением 50 кОм</p>

7.14.1.14 CPULLDOWN

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_004C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPULLDOWN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPULLDOWN [15:0]															

Бит	Имя	Значение	Описание
31...0	CPULLDOWN [31:0]	0x00000000	<p>Регистр сброса подтяжки к земле вывода порта</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SPULLDOWN[0] соответствует выводу порта с номером 0 SPULLDOWN[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – нет подтяжки 1 – есть подтяжка к земле через резистор R_{PULLDOWN} с типовым значением 50 кОм</p>

7.14.1.15 SPD

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0050												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPD[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPD [15:0]															

Бит	Имя	Значение	Описание
31...0	SPD [31:0]	0x00000000	<p>Регистр установки типа драйвера вывода порта</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>SPD[0] соответствует выводу порта с номером 0 SPD[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – драйвер с активным высоким и низким уровнями 1 – драйвер с активным низким уровнем (открытый сток)</p>

7.14.1.16 CPD

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0054												
------------	---	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPD[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPD [15:0]															

Бит	Имя	Значение	Описание
31...0	CPD [31:0]	0x00000000	<p>Регистр сброса типа драйвера вывода порта</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>CPD[0] соответствует выводу порта с номером 0 CPD[1] соответствует выводу порта с номером 1 и так далее.</p> <p>0 – драйвер с активным высоким и низким уровнями 1 – драйвер с активным низким уровнем (открытый сток)</p>

7.14.1.17 SPWRx

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0058 0x0000_005C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWR15		PWR14		PWR13		PWR12		PWR11		PWR10		PWR9		PWR8	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWR7		PWR6		PWR5		PWR4		PWR3		PWR2		PWR1		PWR0	

Бит	Имя	Значение	Описание
31...30	PWR15[1:0]	0x0	Регистр задания скорости драйвера
29...28	PWR14[1:0]	0x0	
27...26	PWR13[1:0]	0x0	При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.
25...24	PWR12[1:0]	0x0	
23...22	PWR11[1:0]	0x0	00 – драйвер выключен 01 – медленный фронт 300 нс 10 – средний фронт 100 нс 11 – быстрый фронт 10 нс
21...20	PWR10[1:0]	0x0	
19...18	PWR9[1:0]	0x0	SPWR0[1:0] – вывод 0 SPWR0[3:2] – вывод 1 SPWR0[5:4] – вывод 2 SPWR0[7:6] – вывод 3 SPWR0[9:8] – вывод 4
17...16	PWR8[1:0]	0x0	
15...14	PWR7[1:0]	0x0	SPWR0[11:10] – вывод 5 SPWR0[13:12] – вывод 6 SPWR0[15:14] – вывод 7 SPWR0[17:16] – вывод 8 SPWR0[19:18] – вывод 9 SPWR0[21:20] – вывод 10 SPWR0[23:22] – вывод 11 SPWR0[25:24] – вывод 12 SPWR0[27:26] – вывод 13 SPWR0[29:28] – вывод 14 SPWR0[31:30] – вывод 15
13...12	PWR6[1:0]	0x0	
11...10	PWR5[1:0]	0x0	SPWR1[1:0] – вывод 16 SPWR1[3:2] – вывод 17 SPWR1[5:4] – вывод 18 SPWR1[7:6] – вывод 19 SPWR1[9:8] – вывод 20 SPWR1[11:10] – вывод 21 SPWR1[13:12] – вывод 22 SPWR1[15:14] – вывод 23 SPWR1[17:16] – вывод 24 SPWR1[19:18] – вывод 25 SPWR1[21:20] – вывод 26 SPWR1[23:22] – вывод 27 SPWR1[25:24] – вывод 28 SPWR1[27:26] – вывод 29 SPWR1[29:28] – вывод 30 SPWR1[31:30] – вывод 31
9...8	PWR4[1:0]	0x0	
7...6	PWR3[1:0]	0x0	
5...4	PWR2[1:0]	0x0	
3...2	PWR1[1:0]	0x0	
1...0	PWR0[1:0]	0x0	

7.14.1.18 CPWRx

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0060 0x0000_0064												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWR15		PWR14		PWR13		PWR12		PWR11		PWR10		PWR9		PWR8	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWR7		PWR6		PWR5		PWR4		PWR3		PWR2		PWR1		PWR0	

Бит	Имя	Значение	Описание	
31...30	PWR0[1:0]	0x0	Регистр сброса скорости драйвера	
29...28	PWR0[1:0]	0x0		
27...26	PWR0[1:0]	0x0		
25...24	PWR0[1:0]	0x0	При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.	
23...22	PWR0[1:0]	0x0		
21...20	PWR0[1:0]	0x0		
19...18	PWR0[1:0]	0x0	00– драйвер выключен	
17...16	PWR0[1:0]	0x0	01 – медленный фронт 300 нс	
15...14	PWR0[1:0]	0x0	10 – средний фронт 100 нс	
13...12	PWR0[1:0]	0x0	11 – быстрый фронт 10 нс	
11...10	PWR0[1:0]	0x0	CPWR0[1:0] – вывод 0 CPWR0[3:2] – вывод 1 CPWR0[5:4] – вывод 2 CPWR0[7:6] – вывод 3 CPWR0[9:8] – вывод 4 CPWR0[11:10] – вывод 5 CPWR0[13:12] – вывод 6 CPWR0[15:14] – вывод 7 CPWR0[17:16] – вывод 8 CPWR0[19:18] – вывод 9 CPWR0[21:20] – вывод 10 CPWR0[23:22] – вывод 11 CPWR0[25:24] – вывод 12 CPWR0[27:26] – вывод 13 CPWR0[29:28] – вывод 14 CPWR0[31:30] – вывод 15	
9...8	PWR0[1:0]	0x0		
7...6	PWR0[1:0]	0x0		
5...4	PWR0[1:0]	0x0		
3...2	PWR0[1:0]	0x0		
1...0	PWR0[1:0]	0x0		
				CPWR1[1:0] – вывод 16
				CPWR1[3:2] – вывод 17
				CPWR1[5:4] – вывод 18
				CPWR1[7:6] – вывод 19
				CPWR1[9:8] – вывод 20
				CPWR1[11:10] – вывод 21
				CPWR1[13:12] – вывод 22
				CPWR1[15:14] – вывод 23
				CPWR1[17:16] – вывод 24
			CPWR1[19:18] – вывод 25	
			CPWR1[21:20] – вывод 26	
			CPWR1[23:22] – вывод 27	
			CPWR1[25:24] – вывод 28	
			CPWR1[27:26] – вывод 29	
			CPWR1[29:28] – вывод 30	
			CPWR1[31:30] – вывод 31	

7.14.1.19 SCL

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0068												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CL[15:0]															

Бит	Имя	Значение	Описание
31...0	CL[31:0]	0x00000000	<p>Регистр установки разрешения ограничения по току для вывода</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – нет контроля ограничения по току 1 – есть контроль ограничения по току</p> <p>CL[0] соответствует выводу порта с номером 0 CL[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.20 CCL

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_006C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CL[15:0]															

Бит	Имя	Значение	Описание
31...0	CL[31:0]	0x00000000	<p>Регистр сброса разрешения ограничения по току для вывода</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – нет контроля ограничения по току 1 – есть контроль ограничения по току</p> <p>CL[0] соответствует выводу порта с номером 0 CL[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.21 SIE

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000					Offset=	0x0000_0070									
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
IE[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IE[15:0]																

Бит	Имя	Значение	Описание
31...0	IE[31:0]	0x00000000	<p>Регистр установки разрешения прерывания по выводу</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – прерывание и фиксация прерывания запрещены 1 – прерывание и фиксация прерывания разрешены</p> <p>IE[0] соответствует выводу порта с номером 0 IE[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.22 CIE

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0074												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IE[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE[15:0]															

Бит	Имя	Значение	Описание
31...0	IE[31:0]	0x00000000	<p>Регистр сброса разрешения прерывания по выводу</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – прерывание и фиксация прерывания запрещены 1 – прерывание и фиксация прерывания разрешены</p> <p>IE[0] соответствует выводу порта с номером 0 IE[1] соответствует выводу порта с номером 1 и так далее</p>

7.14.1.23 SIT

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0078												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT[15:0]															

Бит	Имя	Значение	Описание
31...0	IT[31:0]	0x00000000	<p>Регистр установки рабочего уровня прерывания по выводу</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – прерывание и фиксация прерывания по низкому уровню 1 – прерывание и фиксация прерывания по высокому уровню</p> <p>Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE</p> <p>IT[0] соответствует выводу порта с номером 0 IT[1] соответствует выводу порта с номером 1 и так далее</p>

7.14.1.24 СИТ

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_007C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT[15:0]															

Бит	Имя	Значение	Описание
31...0	IT[31:0]	0x00000000	<p>Регистр сброса рабочего уровня прерывания по выводу</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – прерывание и фиксация прерывания по низкому уровню 1 – прерывание и фиксация прерывания по высокому уровню</p> <p>Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE</p> <p>IT[0] соответствует выводу порта с номером 0 IT[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.25 SIR

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0080												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR[15:0]															

Бит	Имя	Значение	Описание
31...0	IR[31:0]	0x00000000	<p>Регистр установки запроса прерывания по выводу</p> <p>При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – нет запроса прерывания 1 – есть запрос прерывания</p> <p>Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE</p> <p>IR[0] соответствует выводу порта с номером 0 IR[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.26 CIR

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0084												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR[15:0]															

Бит	Имя	Значение	Описание
31...0	IR[31:0]	0x00000000	<p>Регистр сброса запроса прерывания по выводу</p> <p>При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.</p> <p>0 – нет запроса прерывания 1 – есть запрос прерывания</p> <p>Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE</p> <p>IR[0] соответствует выводу порта с номером 0 IR[1] соответствует выводу порта с номером 1 и так далее.</p>

7.14.1.27 HCUR

Base ADDR=	0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000	Offset=	0x0000_0088												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HCUR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCUR[15:0]															

Бит	Имя	Значение	Описание
31...0	HCUR[31:0]	0x00000000	<p>Регистр флага превышения по току по выводу</p> <p>При записи 1 производится сброс флага, если при этом закончилось событие превышения</p> <p>0 – нет превышения по току 1 – есть превышение по току</p> <p>HCUR[0] соответствует выводу порта с номером 0 HCUR [1] соответствует выводу порта с номером 1 и так далее</p>

7.15 Контроллер Ethernet (EthernetMAC)

Память блока контроллера Ethernet имеет встроенную ECC-защиту. Встроенный PHY обеспечивает работу на скорости до 10 Мбит/с. Контроллер MAC поддерживает режимы работы 10 и 100 Мбит/с. При режиме 100 Мбит/с необходимо применение внешнего PHY, подключаемого по интерфейсу MII. В самом протоколе Ethernet на уровне пакетов используется защита данных по CRC.

Встроенная память контроллера EthernetMAC отображается в адресное пространство на уровне арбитража шины АНВ, при этом регистры управление контроллером Ethernet отображены в адресное пространство периферии. Отображаемая память EthernetMAC снабжена защитой ECC аналогично памяти RAMD. Передаваемые и принимаемые данные по сети Ethernet защищены контрольными суммами CRC согласно протоколу стандарта Ethernet. При приеме по сети Ethernet данные формируются в 32-х разрядные слова, для которых генерируется ECC и записываются в память. При считывании со стороны процессора проверка ECC производится на входе в процессор. При записи со стороны процессора данные в память записываются с проверкой ECC и генерацией новой ECC, аналогично работе памяти RAMD.

Объем памяти одного контроллера Ethernet составляет 8 Кбайт.

При работе блок MAC выполняет две основные функции:

- формирование пакета уровня звена данных протоколов Ethernet/IEEE802.3 и передача его на физический уровень;
- прием с физического уровня и разбор пакета уровня звена данных протоколов Ethernet/IEEE802.3.

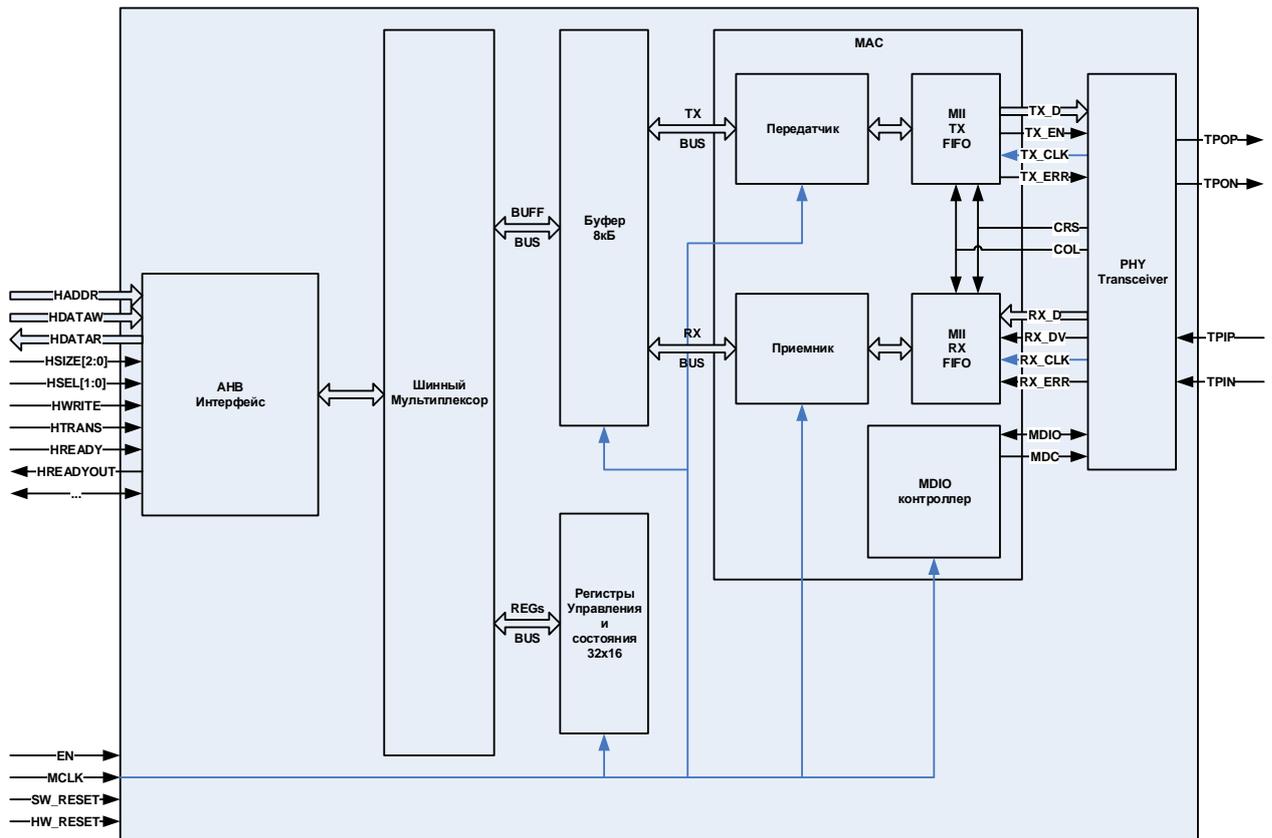


Рисунок 78 – Структурная схема контроллера Ethernet

Работа блока возможна в полно- и полудуплексном режимах. Переключение режима осуществляется битом G_CFG.HD_EN: («1» – полудуплексный режим работы).

Также в общее управление блоком входит разрешение обработки пакета PAUSE (G_CFG.PAUSE_EN), управление размером окна распознаваний коллизий (G_CFG.ColWnd) и режимом работы буферов приемника и передатчика (G_CFG.BUFF_MODE).

Режимы работы буферов:

- линейный (BUFF_MODE=00b);
- автоматический (BUFF_MODE=01b);
- режим FIFO (BUFF_MODE=10b).

Режимы различаются способом обработки указателей границ пустых и полных областей буферов приемника и передатчика:

- В линейном режиме границы начала пустой области буфера приемника (head_R) и передатчика (tail_X) определяются записью в соответствующие регистры управления.
- В автоматическом режиме эти границы определяются автоматически – по адресам последнего чтения или записи в соответствующие области.
- В режиме FIFO границы определяются автоматически, при этом обращение к приемнику производится через адрес 0x0000, а к передатчику через адрес 0x0004. Размер области памяти контроллера Ethernet выделяемой под буфер приёмника и буфер передатчика можно менять с помощью регистра DELIMITER. По сбросу граница между буфером приёмника и передатчика устанавливается посередине, т.е. в 0x1000.

7.15.1 Передача пакета

Для передачи пакета(ов) необходима предварительная настройка блока передатчика MAC. Настройка параметров работы передатчика осуществляется посредством регистра управления передатчика X_CFG:

- разрешение работы передатчика (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- выбор события при передаче, выводимого на вывод EVNT[1] (поле EVNT_MODE);
- управление дополнением пакета до минимальной длины PAD-ами (бит PAD_EN);
- управление дополнением пакета преамбулой (бит PRE_EN), SFD добавляется в любом случае;
- управление дополнением пакета, автоматически подсчитываемым полем CRC (CRC_EN);
- управление интервалом между отправлением пакетов (бит IPG_EN);
- управление максимальным числом повторений (поле RtryCnt).

Далее необходимо записать пакет для передачи в буфер передатчика.

Пакет для передачи содержит три поля (все поля должны быть выровнены по границе слова буферного ОЗУ):

- поле управления передачей пакета;
- собственно, данные пакета уровня звена данных;
- поле состояния передачи пакета.

Поле управления содержит количество передаваемых данных пакета в байтах. Поле состояния заполняется по завершении процедуры отправки пакета (успешной или нет) и содержит статусную информацию по отправке пакета: о наличии ошибок при его передаче, о количестве попыток передачи пакета и прочее.

Если выбран линейный режим работы буферов, то после помещения пакета для передачи, необходимо записать в регистр управления tail_X границу пустой области (адрес, следующий за последним словом пакета).

По завершении передачи пакета, блок MAC выставит один из флагов прерываний передатчика.

Во время отправки пакета или по его завершении выставляется событие, запрограммированное в поле X_CFG.EVNT_MODE.

7.15.2 Принцип работы передатчика

Передатчик начинает работать, прочитав ненулевое поле длины из буфера передатчика. Для этого необходимо, чтобы буфер передатчика был не пуст (ненулевая разница между значениями head_X и tail_X) и передатчику было разрешено работать (X_CFG.EN=1). Прочитав слово управления, передатчик перемещает указатель head_X на первое слово пакета данных. При получении управляющего слова, в передатчике также фиксируется вся управляющая информация для работы передатчика, препятствуя срыву передачи текущего пакета и позволяя сменить настройки для отправки следующего пакета во время передачи текущего.

По завершении передачи в слово, следующее за последним словом данных, записывается статусная информация отправки пакета.

7.15.3 Прием пакета

Для приема пакета(ов) необходима предварительная настройка блока приемника MAC. Настройка параметров работы приемника осуществляется посредством регистра управления приемника R_CFG:

- разрешение работы приемника (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- выбор события при передаче, выводимого на вывод EVNT[0] (поле EVNT_MODE);
- управление разрешением приема пакетов:
 - длины меньше минимально разрешенной (SF_EN);
 - длины больше максимально разрешенной (LF_EN);
 - пакетов управления (CF_EN);
 - пакетов, содержащих ошибки (EF_EN);
- управление фильтрацией по MAC-адресу:
 - разрешение приема пакетов с заданным MAC-адресом (UCA_EN);
 - разрешение приема пакетов с широковещательным MAC-адресом (BCA_EN);
 - разрешение приема пакетов с групповым MAC-адресом (MCA_EN);
 - разрешение приема пакетов с любым MAC-адресом (AC_EN).

Для приема пакета необходимо, чтобы в буфере приемника было достаточно пустого места для принимаемого пакета.

Принятый пакет содержит 2 поля (все поля выровнены по границе слова буферного ОЗУ):

- поле состояния приема пакета;
- собственно, данные пакета уровня звена данных.

Поле состояния заполняется по успешном завершении процедуры приема пакета и содержит количество байт в пакете (включая заголовок пакет уровня звена данных), а также статусную информацию по приему пакета, о наличии ошибок при приеме.

Если выбран линейный режим работы буферов, то указание начала свободной для приема данных области указывается в регистре head_R.

По завершении приема пакета, блок MAC выставит один из флагов прерываний приемника.

Во время приема пакета или по его завершении выставляется событие, запрограммированное в поле R_CFG.EVNT_MODE.

7.15.4 Принцип работы приемника

Приемник начинает работать сразу же после разрешения работы приемника в регистре R_CFG (R_CFG.EN=1), после обнаружения свободного места в буфере приемника. Обнаружив наличие свободного места, приемник фиксирует всю управляющую информацию для работы приемника, препятствуя срыву приема изменениями настроек, и позволяя сменить настройки для приема следующего пакета. Затем переходит в режим ожидания данных на входе и после поступления данных – в режим приема. По завершении приема в слово, следующее за

последним словом данных, записывается статусная информация по приему пакета. Пакеты, отброшенные по причине ошибок в них или не прошедшие фильтрацию по MAC-адресу, переводят приемник в режим ожидания нового пакета. Таким образом, не изменяется общее состояние приемника, изменяется лишь состояние регистра флагов прерываний.

7.15.5 Линейный режим работы буферов

Данный режим включается сбросом поля BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b00). В данном режиме всё управление границами свободных областей в буферах осуществляется вручную.

7.15.6 Автоматический режим работы буферов

Для включения данного режима необходимо установить значение 1 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE=2'b01). В данном режиме в буфере автоматически отслеживаются указатели границ достоверных данных для передачи и приема по адресу записи в буфер передатчика и адресу чтения из буфера приемника. В данном режиме нет необходимости ручного управления границей свободного места в приемнике и передатчике через запись в соответствующие регистры. Это позволяет упростить алгоритм запуска передачи и приема, а также передавать данные одновременно с их помещением в буфер передатчика. Граница достоверных данных в буфере приемника перемещается по завершении приема пакета и, таким образом, данный режим не допускает одновременного приема пакета и его чтения из буфера

7.15.7 Режим FIFO работы буферов

Режим FIFO отличается от предыдущих режимов полностью автоматическим отслеживанием данных в буфере. В данный режим модель переводится установкой значения 2 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b10). В данном режиме чтение/запись в буферы статусной и управляющей информации, а также данных осуществляются через один адрес: 0x0000 – для приемника, и 0x0004 – для передатчика. Для обеспечения корректной работы в режиме отладки необходимо использовать биты G_CFG.DBG_XF_EN и G_CFG.DBG_RF_EN для корректной работы со средствами отладки (для исключения некорректной обработки указателей буферов при обновлении карты памяти средствами отладки). Данный режим позволяет работать на максимальной скорости, если инструментальные средства управляющего контроллера не обеспечивают режимы адресации с автоинкрементом и циклической буферизацией.

7.15.8 События приемника и передатчика

В блоке MAC присутствуют 2 вывода индикации событий: события передатчика (EVNT[1]) и приемника (EVNT[0]). Основным назначением этих выводов является информирование управляющего процессора или DMA-контроллера о наличии данных для перемещения. Выводы EVNT программируемые. В качестве источника события могут быть выбраны:

- состояние буфера;
- начало приема/передачи пакета;
- завершение приема/передачи пакета;
- перемещение слова из/в буфера.

Первое событие предназначено для непрерывного обмена информацией с контролем состояния приема/передачи пакетов при использовании высокоуровневых протоколов. Следующие два события обеспечивают прием и передачу пакетов в интерактивном режиме с непосредственным контролем их приема и передачи. Последнее событие предназначено для непосредственного контроля записи/чтения слов данных в/из буферов, и основное назначение этого события – режим отладки.

7.15.9 Прерывания

Прием и передача пакетов сопровождаются не только формированием событий, но и формированием прерываний, которые служат для непосредственного отражения оперативной статусной информации о состоянии Ethernet-контроллера. Вывод прерывания один, он обеспечивает общую индикацию наличия флагов в регистре флагов прерываний (IFR), разрешенных регистром маски прерываний (IMR). Все прерывания маскируемые. 1 в бите регистра маски прерываний (IMR) разрешает соответствующее прерывание, 0 – запрещает соответствующее прерывание. Прерывания делятся на три группы:

- прерывания MDIO интерфейса;
- прерывания передатчика;
- прерывания приемника.

Прерывания MDIO интерфейса информируют о завершении затребованной операции по MDIO интерфейсу.

Прерывания передатчика показывают состояние отправки пакетов, включая информацию об успешной отправке или наличии ошибок.

Прерывания приемника отражают состояние приема пакета, включая информацию о приеме пакета без ошибок или наличии ошибок при приеме.

Все флаги прерываний кумулятивные. Сброс флагов производится чтением регистра, если бит RCLR_EN регистра G_CFGI установлен ($G_CFG.I.RCLR_EN = 1$), или записью 1 в соответствующий разряд регистра IFR.

7.15.10 Режим детерминированного времени доставки

Данный режим является расширением стандарта IEEE 802.3/Ethernet для обеспечения детерминированного времени доставки. Режим включается установкой бита DTRM_EN регистра G_CFGI ($G_CFG.I.DTRM_EN = 1$). Данный режим может использоваться только в полнодуплексном режиме работы (значение бита $G_CFG.I.HD_EN = 1$ блокирует данный режим).

В данном режиме для начала передачи пакета выделяется интервал размером, задаваемым регистром JitterWnd (размер окна = $JitterWnd + 1$), с периодом задаваемым регистром BAG (период = $BAG + 1$). Единица измерения периода и размера джиттера задается регистром PSC в тактах основной частоты работы блока (размер единицы = $PSC + 1$).

7.15.11 Режим КЗ

В блоке MAC для целей тестирования алгоритмов обработки данных предусмотрен режим короткого замыкания (КЗ). В данном режиме выход передатчика переключается на вход приемника. Также, в данном режиме блок принудительно переводится в полнодуплексный режим работы.

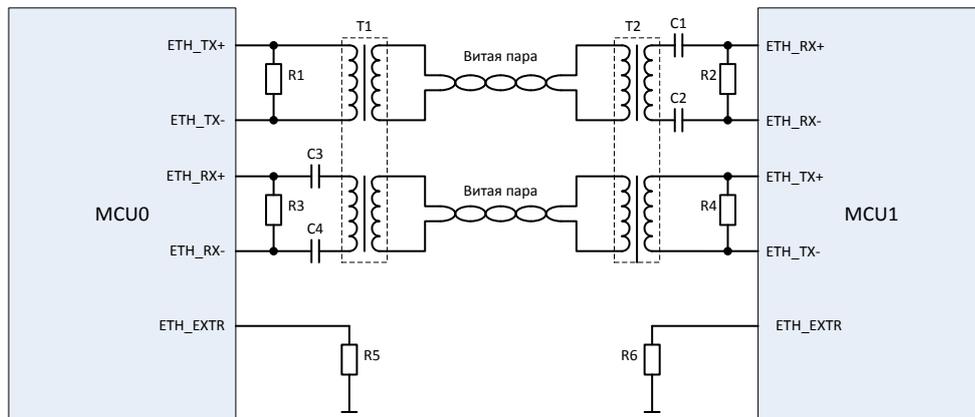
7.15.12 Режимы отладки

В блоке MAC предусмотрено различное поведение блока в режиме отладки. Это поведение определяется полем $G_CFG.H.DBG_MODE$. Переход из штатного режима работы в один из отладочных режимов осуществляется при подаче сигнала $DBG_EN = 1$ на вход модуля. Доступны три режима отладки:

- FreeRun ($G_CFG.H.DBG_MODE = 2'b0X$). В этом случае блок продолжает работать в штатном режиме.
- Halt ($G_CFG.H.DBG_MODE = 2'b10$). Останов приемника и передатчика осуществляется сразу после перехода в данный режим.
- Stop ($G_CFG.H.DBG_MODE = 2'b11$). Останов приемника и передатчика осуществляется по завершении прима/передачи текущего пакета.

7.15.13 Блок PHY

Схема включения интерфейса Ethernet с использованием встроенных приемопередатчиков приведена на рисунке .



- R1, R2, R3, R4 – 100 Ом ± 1 %;
- T1, T2 – трансформаторы для Ethernet линии;
- C1, C2, C3, C4 – конденсаторы 1 нФ;
- R5, R6 – резистор 120 кОм ± 1 %.

Рисунок 79 – Схема подключения к линии

Управление режимами блока осуществляется через регистры PHY_CTRL и PHY_STAT блока MAC. При этом через регистр PHY_CTRL осуществляется программный сброс блока, а также настройка режимов его работы после сброса. В регистре PHY_STAT отражается информация о текущем состоянии блока PHY. Для корректной работы блока необходимо настроить частоту тактирования EPHYCLK равной 80 МГц через регистр EPHY_CLK блока CLKCNTNTR.

7.15.14 Описание регистров

Таблица 105 – Описание регистров контроллера Ethernet

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x2100_8000				Буфер данных контроллера Ethernet 0
0x2100_0000				Регистры управления контроллера Ethernet 0
	0x0000_0000	DELIMITER	RW, 0x1000	Регистр границы буферов приемника и передатчика
	0x0000_0002	MAC_ADDRESS		Регистр индивидуального MAC-адреса
	0x0000_0002	MAC_T	RW, 0x78AB	Младшая* часть индивидуального MAC-адреса
	0x0000_0004	MAC_M	RW, 0x3456	Средняя часть индивидуального MAC-адреса
	0x0000_0006	MAC_H	RW, 0x0012	Старшая часть индивидуального MAC-адреса
	0x0000_0008	HASH		HASH-таблица групповых адресов
	0x0000_0008	HASH0	RW, 0x0000	Младшая часть HASH-таблицы
	0x0000_000A	HASH1	RW, 0x0000	Средняя часть HASH-таблицы
	0x0000_000C	HASH2	RW, 0x0000	Средняя часть HASH-таблицы
	0x0000_000E	HASH3	RW, 0x8000	Старшая часть HASH-таблицы

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0x0000_0010	IPG	RW, 0x0060	Регистр задания межпакетного интервала для полнодуплексного режима
	0x0000_0012	PSC	RW, 0x0050	Регистр задания предделителя шага изменения значений BAG и JitterWnd (1 мкс при частоте 50 МГц)
	0x0000_0014	BAG	RW, 0x0200	Регистр задания периода следования пакетов (100 мкс при частоте 50 МГц)
	0x0000_0016	JITTERWND	RW, 0x0005	Регистр задания джиттера при передаче пакетов (5 мкс при частоте 50 МГц)
	0x0000_0018	R_CFG	RW, 0x0507	Регистр управления приемника
	0x0000_001A	X_CFG	RW, 0x01FA	Регистр управления передатчика
	0x0000_001C	G_CFGI	RW, 0x4880	Регистр общего управления блоком, младшее полуслово
	0x0000_001E	G_CFGh	RW, 0x3000	Регистр общего управления блоком, старшее полуслово
	0x0000_0020	IMR	RW, 0x0000	Регистр маски прерываний
	0x0000_0022	IFR	RW, 0x0000	Регистр флагов прерываний
	0x0000_0024	MDIO_CTRL	RW, 0x0000	Регистр управления канала MDIO интерфейса MII
	0x0000_0026	MDIO_DATA	RW, 0x0000	Регистр данных канала MDIO интерфейса MII
	0x0000_0028	R_Head	RW, 0x0000	Указатель начала области действительных данных приемника (указывает на первое непустое слово)
	0x0000_002A	X_TAIL	RW, 0x1000	Указатель конца области действительных данных передатчика (указывает на первое пустое слово)
	0x0000_002C	R_TAIL	R, 0x0000	Указатель начала области действительных данных приемников (указывает на первое непустое слово)
	0x0000_002E	X_HEAD	R, 0x1000	Указатель начала области действительных данных передатчика (указывает на первое непустое слово)
	0x0000_0030	STAT	R, 0x0303	Регистр статуса
	0x0000_0032	RCOUNTER	R, 0x0000	Счетчик количества принятых пакетов (циклический, без насыщения)
	0x0000_0034	PHY_CONTROL	RW,	Регистр управления PHY
	0x0000_0036	PHY_STATUS	R	Регистр флагов статуса PHY
	0x0000_0038	PHY_CONTROL_A	RW	Регистр управления аналогового PHY

* – указано назначение для режима MSB.

7.15.14.1 Поле управления передачей пакета

Base ADDR=												Offset=			
REG Name:		CHIPID													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Length[15:0]															

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	Length[15:0]		Количество байт в пакете

7.15.14.2 Поле состояния передачи пакета

Base ADDR=		Offset=														
REG Name:		CHIPID														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-									-	LC	RL	RCOUNT[3:0]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Бит	Имя	Значение	Описание
31...23	-		Зарезервировано
22	-	0	Зарезервировано
21	LC		Флаг индикации Latecollision во время передачи пакета 1 – произошла Latecollision во время передачи пакета; 0 – Latecollision во время передачи пакета не происходила.
20	RL		Флаг исчерпания попыток передачи пакета 1 – превышено разрешенное количество попыток передачи пакета; 0 – количество попыток передачи пакета не превысило разрешенного значения;
19...16	RCOUNT[3:0]		Число попыток передачи пакета
15...0	-		Зарезервировано

7.15.14.3 Поле состояния приема пакета

Base ADDR=		Offset=														
REG Name:		CHIPID														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
					UCA	BCA	MCA	SMB_ERR	CRC_ERR	DN_ERR	LEN_ERR	SF_ERR	LF_ERR	CF_ERR	PF_ERR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Length[15:0]																

Бит	Имя	Значение	Описание
31...27			Зарезервировано
26	UCA		Признак индивидуального пакета (MAC-адрес соответствует установленному) 1 – MAC-адрес принятого пакета совпадает с MAC-адресом Ethernet-контроллера; 0 – MAC-адрес принятого пакета не совпадает с MAC-адресом Ethernet-контроллера.
25	BCA		Признак широковещательного пакета (MAC = FF_FF_FF) 1 – принят широковещательный пакет; 0 – широковещательный пакет не принят.
24	MCA		Признак группового пакета (MAC соответствует HASH) 1 – принят пакет, удовлетворяющий фильтрации по HASH-таблице; 0 – принятый пакет не удовлетворяет фильтрации по HASH-таблице или фильтрация отключена.
23	SMB_ERR		Признак наличия в пакете ошибочных nibbles 1 – наличие 0 – отсутствие
22	CRC_ERR		Признак несоответствия CRC пакета 1 – произошла ошибка сравнения CRC-пакета с вычисленной CRC; 0 – CRC-пакета и вычисленной CRC совпадают
21	DN_ERR		Количество бит в пакете не кратно 8 1 – не кратно 8 0 – кратно 8
20	LEN_ERR		Признак несоответствия между реальной длиной и длиной, указанной в поле длины – 13,14 октеты 1 – несоответствие 0 – соответствие
19	SF_ERR		Признак недостаточной длины пакета 64 октетов 1 – ошибочная длина 0 – корректная длина
18	LF_ERR		Признак превышение длины пакета 1518 октетов 1 – превышение 0 – норма
17	CF_ERR		Признак пакета управления (фильтрация по специальным MAC и тэгам в поле длины – 13,14 – октеты) 1 – пакет управления 0 – другой пакет
16	PF_ERR		Признак пакета PAUSE 1 – пакет PAUSE 0 – другой пакет
15...0	Length[15:0]		Количество байт в пакете, включая заголовок и CRC

7.15.14.4 DELIMITER

Base ADDR=	0x2100_0000	Offset=	0x0000_0000												
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELIMITER[15:0]															

Бит	Имя	Значение	Описание
15...0	DELIMITER[15:0]	1000	Смещение границы между буферами приёмника и передатчика относительно базового адреса буфера данных контроллера (не путать с базовым адресом регистров контроллера)

7.15.14.5 MAC_T

Base ADDR=	0x2100_0000	Offset=	0x0000_0002												
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_T[15:0]															

Бит	Имя	Значение	Описание
15...0	MAC_T[15:0]	78AB	Младшая часть индивидуального MAC-адреса

7.15.14.6 MAC_M

Base ADDR=	0x2100_0000	Offset=	0x0000_0004												
REG Name:	KEY														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_M[15:0]															

Бит	Имя	Значение	Описание
15...0	MAC_M[15:0]	3456	Средняя часть индивидуального MAC-адреса

7.15.14.7 MAC_H

Base ADDR=		0x2100_0000				Offset=		0x0000_0006							
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_H[15:0]															

Бит	Имя	Значение	Описание
15...0	MAC_H[15:0]	0012	Старшая часть MAC-адреса

7.15.14.8 R_CFG

Base ADDR=		0x2100_0000				Offset=		0x0000_0018							
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	.	BE	MSB1st	.	EVNT_MODE			SF_EN	LF_EN	CF_EN	EF_EN	AC_EN	UCA_EN	BCA_EN	MCA_EN

Бит	Имя	Значение	Описание
15	EN		Разрешение работы приемника. 0 – приемник остановлен; 1 – разрешена работа.
14	-		Зарезервировано
13	BE		Порядок следования байт в слове. 0 – LittleEndian; 1 – BigEndian.
12	MSB1st		Порядок следования бит при приеме байтов данных. 0 – первым принимается LSB; 1 – первым принимается MSB.
11	-		Зарезервировано
10...8	EVNT_MODE[2:0]		Выбор режима работы вывода EVNT[1]. 3'b000 – RFIFO не пуст; 3'b001 – RFIFO почти не пуст; 3'b010 – RFIFO наполовину пуст; 3'b011 – RFIFO почти не полон; 3'b100 – RFIFO не полон; 3'b101 – прием пакета завершен; 3'b110 – приемник положил данных в буфер; 3'b111 – приемник отбросил пакет.
7	SF_EN		Разрешение приема пакетов длиной меньше минимальной. 0 – выключено; 1 – включено.
6	LF_EN		Разрешение приема пакетов длиной больше максимальной. 0 – выключено; 1 – включено.
5	CF_EN		Разрешение приема управляющих пакетов. 0 – выключено;

			1 – включено.
4	EF_EN		Разрешение приема пакетов с ошибками. 0 – выключено; 1 – включено.
3	AC_EN		Прием пакетов без фильтрации MAC-адреса. 0 – выключен; 1 – включен.
2	UCA_EN		Прием пакетов с MAC-адресом, указанным в регистре MAC_Address. 0 – выключен; 1 – включен.
1	BCA_EN		Прием пакетов с широковещательным MAC-адресом. 0 – выключен; 1 – включен.
0	MCA_EN		Прием пакетов с групповым MAC-адресом с фильтрацией по HAS-таблице. 0 – выключен; 1 – включен.

7.15.14.9 X_CFG

Base ADDR=		0x2100_0000				Offset=		0x0000_001A							
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	.	BE	MSB1st	.	EVNT_MODE			PAD_EN	PRE_EN	CRC_EN	IPG_EN	RtryCnt			

Бит	Имя	Значение	Описание
15	EN		Разрешение работы передатчика. 0 – остановлен; 1 – разрешена работа.
14	-		Зарезервировано
13	BE		Порядок следования байт в слове передатчика. 0 – LittleEndian; 1 – BigEndian.
12	MSB1st		Порядок следования бит при передаче байтов данных. 0 – первым передается LSB; 1 – первым передается MSB.
11	-		Зарезервировано
10...8	EVNT_MODE[2:0]		Выбор режима работы вывода EVNT[1]. 3'b000 – XFIFO пуст; 3'b001 – XFIFO почти пуст; 3'b010 – XFIFO наполовину полон; 3'b011 – XFIFO почти полон; 3'b100 – XFIFO полон; 3'b101 – отправка пакета завершена; 3'b110 – передатчик считал слово данных из буфера; 3'b111 – передатчик начал очередную попытку передачи пакета.
7	PAD_EN		Дополнение пакета до минимальной длины PAD-ами. 0 – выключено; 1 – включено.
6	PRE_EN		Дополнение пакета преамбулой. 0 – выключено;

			1 – включено.
5	CRC_EN		Дополнение пакета автоматически высчитанным CRC. 0 – выключено; 1 – включено.
4	IPG_EN		Режим выдержки паузы между отправкой пакетов. 0 – выключен; 1 – включен.
3...0	RtryCnt[3:0]		Максимальное количество попыток отправки пакета

7.15.14.10 G_CFGI

Base ADDR= 0x2100_0000					Offset=					0x0000_001C					
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCLR_EN	BUFF_MODE[1:0]		EXT_EN	HD_EN	DTRM_EN	PAUSE_EN	ColWnd[7:0]							

Бит	Имя	Значение	Описание
15	-		Зарезервировано
14	RCLR_EN		Сброс регистров статуса 0 – производится запись в регистры статуса; 1 – регистры статуса сбрасываются при чтении
13:12	BUFF_MODE[1:0]		Режим работы буфера. 2'b00 – линейный режим; 2'b 01 – режим с автоматическим изменением указателей; 2'b 10 – режим FIFO; 2'b 11 – зарезервировано (линейный режим)
11	EXT_EN		Включение режима дополнения коротких пакетов до размера slotTime полем "Extension" (При приеме отбрасывание слова осуществляется по полю length пакета, если оно отражает длину пакета). 0 – выключен; 1 – включен
10	HD_EN		Полудуплексный режим работы. 0 – выключен; 1 – включен
9	DTRM_EN		Режим детерминированного времени доставки. 0 – выключен; 1 – включен
8	PAUSE_EN		Режим автоматической обработки пакета PAUSE. 0 – выключен; 1 – включен
7:0	ColWnd[7:0]		Размер «окна коллизий»

7.15.14.11 G_CFGh

Base ADDR= 0x2100_0000					Offset=					0x0000_001E					
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

13	-		Зарезервировано
12	CRS_LOST		Индикатор потери несущей во время передачи в полудуплексном режиме работы
11	LC		Индикатор наличия LateCollision в линии
10	UNDF		Индикатор опустошения буфера передатчика
9	XF_ERR		Индикатор наличия ошибок при передаче пакета
8	XF_OK		Индикатор успешной отправки пакета
7	SF		Индикатор приема пакета длиной менее минимальной
6	LF		Индикатор приема пакета длиной более максимальной
5	CF		Индикатор приема управляющих пакетов
4	CRC_ERR		Индикатор наличия несовпадения CRC пакета принятых данных с CRC пакета
3	SMB_ERR		Индикатор наличия ошибок в данных при приеме пакета
2	OVF		Индикатор переполнения буфера приемника
1	MISSED_F		Индикатор потери пакета из-за отсутствия места в буфере приемника
0	RF_OK		Индикатор успешно принятого пакета

Примечание – Индикатор в состоянии единицы означает наличие события, в нуле – отсутствие события.

7.15.14.13 MDIO_CTRL

Base ADDR=		0x2100_0000		Offset=		0x0000_0024									
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDY	PRE_EN	OP	PHY_A				DIV				RG_A				

Бит	Имя	Значение	Описание
15	RDY		Управление/индикатор обмена по MDIO После записи команды необходимо установить в единицу для инициирования исполнения команды в регистре MDIO_CTRL после одного такта сбрасывается в ноль и снова устанавливается в единицу после завершения цикла обмена по интерфейсу MDIO.
14	PRE_EN		Режим передачи. 1 – с передачей преамбулы (32 бита «1»); 0 – без передачи преамбулы.
13	OP		Операция. 1 – чтение; 0 – запись.
12:8	PHY_A[4:0]		Адрес модуля PHY
7:5	DIV[2:0]		Коэффициент деления основной частоты для работы MDIO интерфейса $MDC = ETH_CLK / [(DIV+1)*16]$
4:0	RG_A[4:0]		Номер регистра PHY

7.15.14.14 STAT

Base ADDR=		0x2100_0000				Offset=		0x0000_0030								
REG Name:																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
			X_FULL	X_AFULL	X_HALF	X_AEMPTY	X_EMPTY	RCOUNT			R_FULL	R_AFULL	R_HALF	R_AEMPTY	R_EMPTY	

Бит	Имя	Значение	Описание
15...13	-		Зарезервировано
12	X_FULL		1 — буфер передатчика полон 0 — буфер передатчика не полон
11	X_AFULL		1 — буфер передатчика почти полон 0 — буфер передатчика не в состоянии почти полон
10	X_HALF		1 — буфер передатчика полуполон 0 — буфер передатчика не полуполон
9	X_AEMPTY		1 — буфер передатчика почти пуст 0 — буфер передатчика не в состоянии почти пуст
8	X_EMPTY		1 — буфер передатчика пуст 0 — буфер передатчика не пуст
7...5	R_COUNT		Количество принятых, но не считанных пакетов 0...6 — количество пакетов 7 — количество несчитанных пакетов ≥ 7
4	R_FULL		1 — буфер приемника полон 0 — буфер приемника не полон
3	R_AFULL		1 — буфер приемника почти полон 0 — буфер приемника не в состоянии почти полон
2	R_HALF		1 — буфер приемника полуполон 0 — буфер приемника не полуполон
1	R_AEMPTY		1 — буфер приемника почти пуст 0 — буфер приемника не в состоянии почти пуст
0	R_EMPTY		1 — буфер приемника пуст 0 — буфер приемника не пуст

7.15.14.15 PHY_CONTROL

Base ADDR=		0x2100_0000				Offset=		0x0000_0034									
REG Name:																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RST	EXT_EN			LINK_PERIOD							BASE_2	DIR	EARLY_DV	HALFD	DLB	LB	

Бит	Имя	Значение	Описание
15	RST		Сброс встроенного контроллера PHY-уровня 0 – штатный режим 1 – сброс PHY модуля
14	EXT_EN		Переключение на работу с внешним контроллером PHY-уровня 0 – внутренний блок PHY. 1 – внешний блок PHY (MII интерфейс).
13...12	-		Зарезервировано
11...6	LINK_PERIOD		Период следования LINK импульсов -1. Период ожидания LINK импульсов вдвое больше заданного периода следования LINK импульсов (диапазон 1...64 мс)
5	BASE_2		Переключение на работу с коаксиальным кабелем в режиме полудуплексного приема-передачи 0 – подключение по витой паре 1 – подключение по коаксиальному кабелю
4	DIR		Порядок передачи битов в полубайте 1 – инверсный 0 – прямой
3	EARLY_DV		Включение формирования сигнала RxDV одновременно с сигналом CRS (зарезервировано)
2	HALFD		Включение режима полудуплексного приема-передачи 0 – полнодуплексный режим 1 – полудуплексный режим
1	DLB		Включение тестового замыкания ПРД на ПРМ на входе контроллера PHY-уровня 0 – штатный режим 1 – режим КЗ
0	LB		Включение тестового замыкания ПРД на ПРМ на выходе контроллера PHY-уровня до аналоговой части ПРМ/ПРД 0 – штатный режим 1 – режим КЗ

7.15.14.16 PHY_STATUS

Base ADDR=		0x2100_0000				Offset=		0x0000_0036							
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		INT_JAM	INT_JAB	INT_POL	INT_LOSS	LED_STAT	LED_LINK								

Бит	Имя	Значение	Описание
15...14	-		Зарезервировано
13	INT_JAM		Индикатор встроенного контроллера PHY-уровня о передаче JAM последовательности в случае коллизии (для тестовых нужд) 0 – штатный режим 1 – осуществляется передача JAM-последовательности
12	INT_JAB		Индикатор встроенного контроллера PHY-уровня о превышении времени передачи максимально разрешенной (для тестовых нужд) 0 – штатный режим 1 – передача текущего пакета ведется более максимально допустимого интервала
11	INT_POL		Индикатор встроенного контроллера PHY-уровня о смене полярности сигналов в линии PPM 0 – нормальная полярность импульсов LINK 1 – инверсная полярность импульсов LINK
10	INT_LOSS		Индикатор встроенного контроллера PHY-уровня о наличии подключения в линии 0 – линия подключена 1 – потеря синхронизации
9	LED_STAT		Индикатор встроенного контроллера PHY-уровня о наличии коллизии в линии (для тестовых нужд) 0 – штатный режим 1 – на линии имеется коллизия
8	LED_LINK		Индикатор встроенного контроллера PHY-уровня о наличии несущей в линии (для тестовых нужд) 0 – штатный режим 1 – в линии обнаружена несущая
7..0	-		Зарезервировано

7.15.14.17 PHY_CONTROL_A

Base ADDR=		0x2100_0000				Offset=		0x0000_0038							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_PHY	ISRC			TRIMR[7:0]							TRIM[2:0]			SELR	

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15	EN_PHY	0	Бит разрешения работы аналогового PHY 1 – разрешен 0 – запрещен
14	ISRC	0	Бит увеличение рабочего тока передатчика при низком напряжении питания 1 – должен быть установлен при питании ниже 4,0 В; 0 – должен быть сброшен при питании выше 4,0 В
13...12	-		Зарезервировано
11...4	TRIMR[7:0]	0000_0000	Вход подстройки внутреннего резистора источника опорного тока 1000_0000 Типовое значение опорного тока (10 мкА) 0000_0000 Минимальный опорный ток (5 мкА) 1111_1111 Максимальный опорный ток (15 мкА)
3...1	TRIM[2:0]	000	Вход цифрового тримминга источника опорного напряжения Типовое значение 100 (1,2 В). 000 – максимальное напряжение (1,25 В) 111 – минимальное напряжение (1,15 В)
0	SELR	0	Вход выбора резистора источника опорного тока. 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен); 1 – используется внешний стабильный резистор на выводе ETH_EXTR на землю

7.16 Контроллер SpaceWire (SpaceWire)

7.16.1 Кодирование данных

Каналы SpaceWire – это последовательные высокоскоростные (от 2 до 400 Мб/с) полнодуплексные каналы (DS-линки) типа "точка-точка". Стандарт SpaceWire использует Data-Strobe (DS)-кодирование для передачи данных, которое изображено на рисунке .

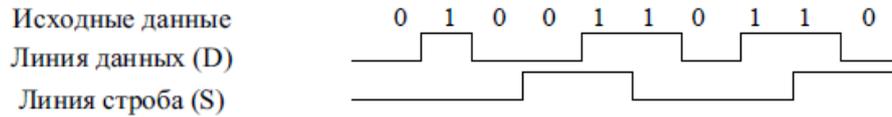


Рисунок 80 – Data-Strobe (DS)-кодирование

Данная схема позволяет кодировать синхросигнал передачи в линиях данных таким образом, что синхросигнал может быть восстановлен операцией исключающее "или" двух линий D и S. Линия D повторяет значение бита данных, т.е. имеет высокий потенциал для бита логической "1" и низкий потенциал для логического "0". Линия S изменяет свое состояние каждый раз, когда состояние линии D остается неизменным для следующего бита.

Канал SpaceWire включает две пары дифференциальных сигналов, одна пара D и S передающая, другая пара D и S принимающая. В результате для двунаправленного канала требуется восемь однонаправленных линий передачи данных.

7.16.2 Скорость интерфейса

Скорость работы интерфейса на передачу определяется частотой синхронизации SPWCLK, которая должна быть равна частоте контроллера SpaceWire(=CPUCLK) или ее удвоенному значению. Каналы приема и передачи в интерфейсе асинхронны. Скорость интерфейса на прием задается внешним передатчиком. Для корректной работы приемника скорость в канале приема не должна превышать частоту CPUCLK более чем в 2 раза. Например, если частота CPUCLK равна 40 МГц, максимально возможная скорость интерфейса не будет превышать 80 Мбит/с.

7.16.3 Представление символов

Битовый поток состоит из символов данных и управления. Тип символа определяется в зависимости от значения флага данные/управление. Символы данных содержат восьмиразрядные значения данных, передаваемых младшими разрядами вперед. Каждый символ данных состоит из бита паритета, контрольного флага и восьми бит данных. Контрольный флаг для символа данных равен нулю. Структура символа данных показана на рисунке .



Рисунок 81 – Символ данных Space Wire

Управляющий символ состоит из бита контроля по четности, флага данные/управление и двух бит кода управления. Значение флага данные/управление равное 1 указывает, что данный символ является символом управления. Возможные управляющие символы показаны на рисунке . Символ расширения ESC используется для создания дополнительных

управляющих символов – маркеров. В спецификации SpaceWire символ ESC используется исключительно для формирования NULL-маркера или маркера времени. Все другие применения этого символа считаются некорректными и рассматриваются как ошибка расширения. NULL-маркер состоит из символа расширения ESC и следующим за ним символом управления потоком FCT. Бит контроля по четности P в середине маркера NULL равен 0. NULL-маркер передается всегда при отсутствии передач символов данных и символов управления в канале. Это позволяет сохранить активность канала и определить ошибку рассоединения.



Рисунок 82 – Управляющие символы Space Wire

Маркер времени состоит из символа расширения ESC и символа данных. Бит контроля по четности P в середине маркера времени равен 1. Этот маркер используется для поддержания единого системного времени в сети.

Бит контроля по четности добавляется в символы данных или управления для определения ошибок передачи по каналу (Рисунок 83).

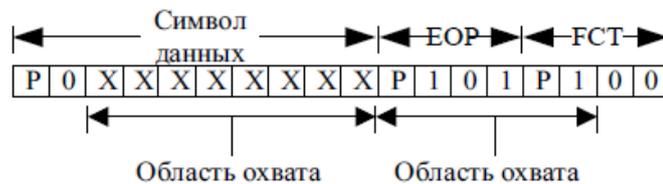


Рисунок 83 – Область, контролируемая битом паритета

Бит контроля по четности охватывает восемь бит данных предыдущего символа данных или два бита управляющего кода предыдущего символа управления и флаг управления текущего символа, как показано на рисунке. Бит P используется для проверки нечетности, т. е. количество единичных битов в области охвата бита контроля по четности (включая сам бит P) должно быть нечетным.

7.16.4 Диаграмма состояний

Инициализация канала SpaceWire представляет собой сложный процесс синхронного запуска обеих его сторон. При запуске стороны обмениваются служебными символами, информируя друг друга о степени готовности к нормальной работе. При этом используется алгоритм "обмена подтверждениями" и прохождение через определенные состояния интерфейса канала. При запуске интерфейс канала последовательно проходит через следующие состояния (Рисунок 89):

- ErrorReset (Сброс);
- ErrorWait (Ожидание);
- Ready (Готовность);

- Started (Запуск);
- Connecting (Соединение);
- Run (Рабочий режим).

Для того чтобы контролировать состояния канала интерфейса используются биты STATE регистра STATUS. Также возможно формирование прерывания при изменении статуса соединения установкой бита INTENLINK в регистре CONTROL.

В случае возникновения ошибок из любого состояния интерфейс канала сбрасывается и переходит в состояние ErrorReset (Сброс). Ошибки делятся на четыре типа:

- прием символов, не ожидаемых в данном состоянии;
- ошибки таймаута;
- ошибки приема (ошибки расширения символов, ошибки контроля четности, ошибки рассоединения);
- ошибки кредитования.

Для того чтобы контролировать эти ошибки используются флаги ошибок PERR, DSCERR, ESCERR, CREDERR в регистре STATUS. Также возможно формирование прерывание по любому из вышеперечисленных флагов установкой бита INTENERR в регистре CONTROL.

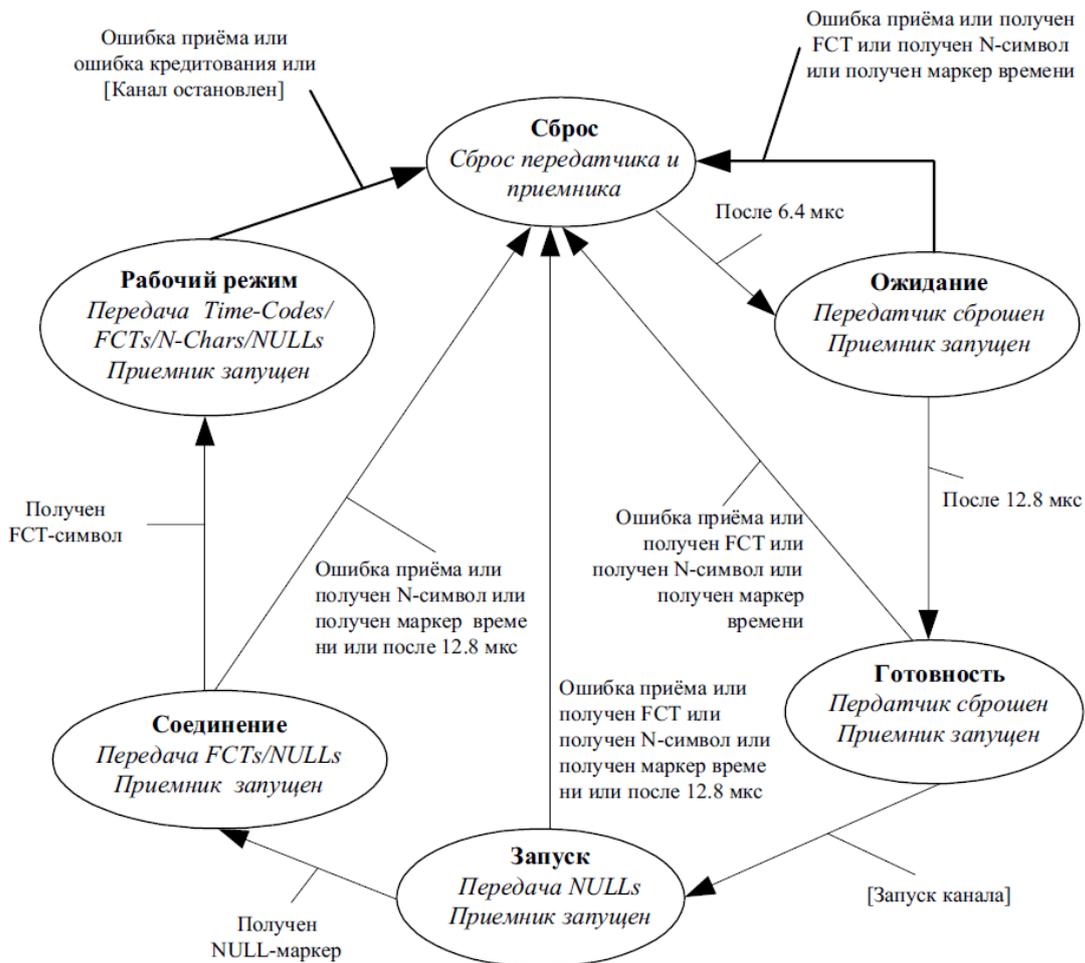


Рисунок 84 – Диаграмма состояния интерфейса канала Space Wire

Определение **ошибки рассоединения** возможно только после приема первого бита после состояния ErrorReset (Сброс). Рассоединение определяется в случае, если на линиях данных и строба не было изменения сигнала в течение 850 нс. Для измерения этого интервала времени необходимо в регистре PAUSE с помощью бит DSCTIME задать количество тактов частоты синхронизации, необходимое на интервал работы счетчика 850 нс.

Ошибка кредитования возникает в случае приема символов данных и отсутствия свободного места в буфере приемника. Первоначально интерфейс канала находится в

состоянии ErrorReset (Сброса). Приемник и передатчик канала сбрасываются. Через 6,4 мкс интерфейс канала переходит в состояние ErrorWait (Ожидания). Запускается приемник канала. Через 12,8 мкс интерфейс канала переходит в состояние Ready (Готовности). В состоянии Ready (Готовности) интерфейс канала ожидает команду от конечного узла на запуск канала. Для разрешения перехода из состояния Ready в состояние Started после получения команды на запуск необходимо установить бит LINKSTART или AUTOSTART в регистре CONTROL. После команды на запуск канала интерфейс канала переходит в состояние Started (Запуска). Запускается передатчик канала. Передатчик канала начинает передавать NULL-маркеры. Для измерения интервалов времени 6,4 мкс, а также 12,8 мкс необходимо в регистре PAUSE с помощью бит RESTIME задать количество тактов частоты синхронизации необходимое на интервал работы счетчика 6,4 мкс.

В состоянии Started (Запуска) интерфейс канала находится до тех пор, пока не будет установлен признак "получен NULL-маркер" (этот признак устанавливается в состояниях ErrorWait (Ожидание), Ready (Готовность) и Started (Запуск) при приеме NULL-маркера). Если в течение 12,8 мкс признак "получен NULL-маркер" не был установлен, и не установлен бит AUTOSTART в регистре CONTROL, канал сбрасывается. При установленном признаке приема NULL-маркера интерфейс канала переходит в состояние Connecting (Соединения) и начинает передавать FCT символы. Если в течение 12,8 мкс FCT символ не был получен, канал также сбрасывается. При приеме FCT символа интерфейс канала переходит в состояние Run (Рабочего режима) и готов нормально функционировать.

Следует отметить, что до перехода в состояние Run интерфейс работает на скорости 10 Мбит/с и использует биты DIVCNTDEF регистра DIV для ее обеспечения. После перехода в состояние Run интерфейс работает на скорости установленной с помощью бит DIVCNT регистра DIV.

Определение ошибок контроля по четности и расширения символов управления, получение информационных символов и символа FCT разрешено только после приема первого NULL-маркера. Определение первого NULL-маркера включает в себя проверку трех бит контроля по четности (Рисунок 90).



Рисунок 85 – Декодирование NULL-последовательности

Проверяются:

- бит Р, охватывающий флаг управления ESC-символа;
- бит Р, охватывающий ESC-символ;
- бит Р, охватывающий FCT-символ.

Следовательно, NULL-маркер может быть принят, и условие "принят NULL-маркер" установлено, если принята последовательность бит 011101000, как показано на рисунке. При включении канала символ, следующий за первым NULL-маркером – управляющий символ (NULL-маркер или FCT-символ). Следовательно, последний бит контроля по четности NULL-маркера равен 0. Если ошибка контроля по четности произошла в первом NULL-маркере, то NULL-маркер не будет получен, и признак "принят NULL-маркер" не будет установлен.

7.16.5 Инициализация

Для работы с контроллером SpaceWire должны быть заданы параметры тактового сигнала.

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 21 для SpaceWire регистра PER0_CLK). В регистре SPW0_CLK установить бит EN_CLK, разрешающий выдачу частоты SPWCLK для контроллера SpaceWire, задать коэффициент деления DIV тактовой частоты MAXCLK для задания SPWCLK.

Далее в регистр DIV записать делители SPWCLK для задания скорости работы контроллера до перехода в состояние Run = 10 Мбит/с (DIVCNTDEF), а также скорости передачи данных (DIVCNT).

Для измерения интервала времени 850 нс необходимо в регистре PAUSE битами DSCTIME задать количество тактов частоты синхронизации необходимое на интервал работы счётчика 850 нс.

Для измерения интервалов времени 6,4 мкс, а также 12,8 мкс необходимо в регистре PAUSE с помощью бит RESTIME задать количество тактов частоты синхронизации необходимое на интервал работы счётчика 6,4 мкс.

При работе через встроенный приёмопередатчик необходимо установить биты EN_BNG, OUT_EN и EN_PHY в регистре PHY_CNTR.

В регистр CONTROL записать биты разрешения перехода из состояния Ready в Started (LINKSTART и AUTOSTART), а также разрешить прерывания по необходимым событиям. Включить интерфейс битом SPWEN.

После прохождения процедуры установления связи между устройствами и перехода контроллера в режим нормального функционирования Run можно принимать и передавать данные.

7.16.6 Run (Рабочий режим)

В рабочем режиме добавляется возможность приема/передачи двух типов символов, передача которых недоступна и ошибочна в других состояниях, а именно Time-Codes (маркеры времени) и N-Chars (символы данных).

В стандарте SpaceWire предусмотрен механизм обмена метками времени между узлами. Источник меток времени в сети SpaceWire должен быть один. Посылка времени приведена на рисунке и на уровне обмена представляет собой слово данных, следующее за командой ESC. Посылка времени имеет наивысший приоритет и отправляется сразу же после окончания отправки текущей команды. Для отправки посылок времени в контроллере предусмотрен регистр TIME, в котором биты TIMEIN содержат маркер времени для передачи. При установке бита TICKIN регистра TIME инициируется процесс передачи содержимого TIMEIN в канал. После чего TICKIN автоматически сбрасывается в ноль. Использование механизма обмена времени в соответствии с протоколом предусматривает наличие 6-ти разрядного счетчика времени (биты TIMEOUT в регистре TIME) в каждом из узлов сети. Оставшиеся два бита из восьми являются командными флагами и зарезервированы для будущего использования (биты CTRLOUT в регистре TIME). В узле источнике времени оба контрольных флага должны быть установлены в 0 (биты CTRLIN в регистре TIME). При получении маркера времени в регистре STATUS устанавливается флаг GOTTIME. Может быть сгенерировано прерывание в момент установки этого флага разрешением маски INTENTIME в регистре CONTROL.

Каждому элементу сети SpaceWire (терминальным узлам и маршрутизаторам) рекомендуется иметь буфер данных (не менее 56 байт). В нашем случае размер буфера данных 64 байта. Узел (маршрутизатор) является ответственным за управление буфером данных. Буфер может быть реализован несколькими методами. Наиболее подходящие для стандарта SpaceWire – FIFO-буферы. Данные из FIFO-буфера передающей стороны канала могут быть переданы только в случае получения FCT-символа (одного или нескольких) с противоположной стороны. Принимающая сторона принимает данные до тех пор, пока есть свободное место в ее FIFO-буфере. Как только в FIFO-буфере принимающей стороны освобождается место для восьми информационных символов, информация об этом должна немедленно попасть передающей стороне канала в виде FCT-символа. Каждый FCT-символ символизирует о восьми

свободных байтах с противоположной стороны. Непрерывно может быть передано не более семи FCT-символов подряд. О состоянии FIFO сигнализируют флаги регистра STATUS, а именно TXF и TXE для FIFO передачи, RXF и RXFNE для FIFO приема. Может быть сгенерировано прерывания по событию установки любого из вышеописанных флагов, если они разрешены масками прерываний INTETXF, INTETXE, INTERXF, INTENRXNE в регистре CONTROL.

Для передачи данных в/из памяти конечного узла может использоваться прямой доступ в память (DMA – Direct Memory Address). Если передача DMA уже подготовлена, возможна передача сразу нескольких FCT-символов для повышения скорости передачи данных. Для разрешения этого режима работы используются биты DMARXEN и DMATXEN регистра CONTROL. Для чтения принятых данных и записи передаваемых данных используются регистры FIFORX и FIFOTX соответственно.

Передача и прием FCT-символов контроллером происходит автоматически независимо от режима работы с DMA или без него. Контроллер имеет счетчик числа символов, разрешенных к передаче (счетчик передачи), и счетчик числа символов, для приема которых есть свободное пространство в буфере приемника (счетчик приема).

Так как символы N-Chars образуют пакеты, ограниченные символами EOP либо EEP, требуется хранить информацию о длине пакетов и типах конца пакетов. Для этих целей служат регистры дескрипторов приема RX_DESC0-RXDESC15 и дескрипторов передачи TXDESC0-TXDESC15. Старший 31 разряд регистров дескрипторов является признаком того, что дескриптор содержит действительные данные. В случае если 31 разряд любого из регистров дескрипторов равен нулю, то этот регистр не содержит достоверной информации. Данные пакета, соответствующие дескрипторам, поступают в FIFO, и так как 64 байта недостаточно для хранения всего пакета, то данные пакета необходимо переносить в/из внешней памяти SRAM большего размера посредством DMA или регистрового чтения FIFORX и FIFOTX. В случае завершения приема или передачи пакета происходит установка флагов TXDESC или RXDESC регистра STATUS. Может быть сгенерировано прерывания по событию установки любого из вышеописанных флагов, если они разрешены масками прерываний INTENRXDES или INTENTXDES в регистре CONTROL.

Имеется возможность формирования статистики принятых пакетов нулевой и ненулевой длины посредством регистров счётчиков принятых пакетов CNTRX0_PACK и CNTRX_PACK соответственно.

7.16.7 Передача сообщений

Для передачи пакета необходимо:

- 1 Определить номер по порядку NUM_TXDESC дескриптора TXDESC[i], определяющего передаваемый пакет;
- 2 В TXDESC[i] задать тип передаваемого пакета EEP или EOP битами 29...30;
- 3 Задать размер битами 0...24;
- 4 Инициировать передачу установкой бита 31;
- 5 Если пакет ненулевой заполнить данными FIFOTX.

7.16.8 Прием сообщений

Для приема пакета необходимо:

- 1 Вычислить дескриптор RXDESC[i], определяющий принятый пакет путем проверки бита 31;
- 2 Определить тип принимаемого пакета EEP или EOP, определяемого битами 29...30;
- 3 Определить длину пакета, определяемого битами 0...24;
- 4 Если пакет ненулевой вычитать данные из FIFORX.

7.16.9 Передача и прием маркера времени

Значение принятого маркера времени определяется битами 8...13 регистра TIME, значение передаваемого – битами 0...5. Для передачи маркера необходимо взвести бит 16, прием осуществляется по флагу GOTTIME регистра STATUS.

7.16.10 Описание регистров

Таблица 106 – Описание регистров контроллера интерфейса SpaceWire

Базовый адрес	Название	Описание
0x4008_8000	SPW	Контроллер интерфейса SpaceWire
Смещение		
0x0000_0000	CONTROL	Регистр управление контроллером
0x0000_0004	STATUS	Регистр состояния контроллера
0x0000_0008	DIV	Регистр делителей частоты контроллера
0x0000_000C	TIME	Регистр маркеров времени
0x0000_0010	PAUSE	Регистр задержек установления соединения
0x0000_0014	FIFORX	FIFO данных приёмника
0x0000_0018	FIFOTX	FIFO данных передатчика
0x0000_001C	CNTRX_PACK	Регистр счетчика принятых пакетов ненулевой длины
0x0000_0020	CNTRX0_PACK	Регистр счетчика принятых пакетов нулевой длины
0x0000_0024	NUM_TXDESC	Регистр номера дескриптора передатчика, планируемого к обработке
0x0000_0028	NUM_RXDESC	Регистр номера дескриптора приёмника, планируемого к заполнению
0x0000_002C	PHY_CTRL	Регистр управления аналоговым приемопередатчиком
0x0000_0040- 0x0000_007C	RXDESC0-SPW_RXDESC15	Память дескрипторов приёмника
0x0000_0080- 0x0000_00BC	TXDESC0-SPW_TXDESC15	Память дескрипторов передатчика

7.16.10.1 CONTROL

Base ADDR=		0x4008_8000				Offset=		0x0000_0000							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERXF34	INTENTXDES	INTENRXDES	INTETXF	INTETXE	INTERXF	INTENRXNE	DMARXEN	DMATXEN	INTENLINK	INTENERR	INTENTIME	AUTO START	LINK START	LINK DIS	SPWEN

Бит	Имя	Значение	Описание
31...16			Зарезервировано
15	INTERXF34		Разрешение прерывания при заполнении 3/4 FIFO приёмника: 0 – прерывание запрещено 1 – прерывание разрешено
14	INTENTXDES		Разрешение прерывания по завершению передачи пакета данных с признаком конца пакета: 0 – прерывание запрещено 1 – прерывание разрешено
13	INTENRXDES		Разрешение прерывания по завершению приёма пакета данных с признаком конца пакета: 0 – прерывание запрещено 1 – прерывание разрешено
12	INTE TXF		Разрешение прерывания при полном заполнении FIFO передатчика: 0 – прерывание запрещено 1 – прерывание разрешено
11	INTE TXE		Разрешение прерывания при отсутствии данных в FIFO передатчика: 0 – прерывание запрещено 1 – прерывание разрешено
10	INTERXF		Разрешение прерывания при полном заполнении FIFO приёмника: 0 – прерывание запрещено 1 – прерывание разрешено
9	INTENRXNE		Разрешение прерывание при наличии в FIFO приёмника хотя бы одного 32-х разрядного слова данных: 0 – прерывание запрещено 1 – прерывание разрешено
8	DMARXEN		Разрешение запроса DMA при заполнении FIFO приёмника: 0 –запрос запрещён 1 – запрос разрешён
7	DMATXEN		Разрешение запроса DMA при опустошении FIFO передатчика: 0 –запрос запрещён 1 – запрос разрешён

Бит	Имя	Значение	Описание
6	INTENLINK		Разрешение прерывания при изменении статуса соединения (соединение установлено/соединение разорвано): 0 – прерывание запрещено 1 – прерывание разрешено
5	INTENERR		Разрешение прерывания при возникновении одной из ошибок: DSCERR, PERR, ESCERR или CREDERR 0 – прерывание запрещено 1 – прерывание разрешено
4	INTENTIME		Разрешение прерывания при приёме маркера времени: 0 – прерывание запрещено 1 – прерывание разрешено
3	AUTOSTART		Разрешение перехода блока Linkinterface из состояния Ready в состояние Started с ожиданием символа NULL (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 0 – переход запрещен; 1 – переход разрешен. В отличие от LinkStart при AutoStart не происходит перехода по таймеру из состояния Started в состояние ErrorReset. Блок Linkinterface переходит из состояния Started в состояние Connecting при получении символа NULL, или переходит в состояние ErrorReset при получении какого-либо другого символа
2	LINKSTART		Разрешение перехода блока Linkinterface из состояния Ready в состояние Started (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 0 – переход запрещен; 1 – переход разрешен
1	LINKDIS		Перевод блока Linkinterface из состояния Run в состояние ErrorReset (LinkDisable): 0 – нет перевода в состояние ErrorReset. Разрешение выполнение команд LinkStart, AutoStart; 1 - перевод в состояние ErrorReset
0	SPWEN		Разрешение работы интерфейса SpaceWire: 1 – работа интерфейса разрешена 0 – интерфейс находится в состоянии сброса Установка бита в единицу разрешается только после инициализации регистров SPW_DIV и SPW_PAUSE.

7.16.10.2 STATUS

Base ADDR=		0x4008_8000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXF34	TXDESC	RXDESC	TXF	TXE	RXF	RXNE	GOT TIME	CRED ERR	ESC ERR	DSC ERR	PERR			STATE

Бит	Имя	Значение	Описание
31...15			Зарезервировано
14	RXF34		Флаг заполненности 3/4 FIFO приёмника: 0 – FIFO не полно 1 – FIFO полно
13	TXDESC		Флаг завершения передачи пакета данных с признаком конца пакета: 0 – передача пакета не завершена 1 – передача пакета завершена При выполнении операции записи 1 в этот разряд он устанавливается в 0
12	RXDESC		Флаг завершения приёма пакета данных с признаком конца пакета: 0 – приём пакета не завершён 1 – приём пакета завершён При выполнении операции записи 1 в этот разряд он устанавливается в 0
11	TXF		Флаг полной заполненности FIFO передатчика: 0 – FIFO не полно 1 – FIFO полно
10	TXE		Флаг отсутствия данных в FIFO передатчика: 0 – FIFO не пусто 1 – FIFO пусто
9	RXF		Флаг полной заполненности FIFO приёмника: 0 – FIFO не полно 1 – FIFO полно
8	RXNE		Флаг наличия данных в FIFO приёмника: 0 – FIFO пусто 1 – FIFO содержит данные
7	GOTTIME		Признак приема маркера времени из канала связи SpaceWire: 1 – принят маркер времени; 0 – нет принятого маркера времени. При выполнении операции записи 1 в этот разряд он устанавливается в 0

Бит	Имя	Значение	Описание
6	CREVERR		Признак ошибки кредитования: 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0
5	ESCERR		Признак ошибки в ESC последовательности (получен неверный символ (пп. 7.3 и 8.9.2.3 стандарта ECSS-E-ST-50-12C): 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0
4	DSCERR		Признак рассоединения с каналом связи SpaceWire: 1 – произошло рассоединение; 0 – нет рассоединения. При выполнении операции записи 1 в этот разряд он устанавливается в 0
3	PERR		Признак ошибки четности: 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0
2...0	STATE[2:0]		Состояние блока Linkinterface (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 000 – Error Reset (исходное состояние); 001 – Error Wait; 010 – Ready; 011 – Started; 100 – Connecting; 101 – Run.

7.16.10.3 DIV

Base ADDR=		0x4008_8000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVCNTDEF								DIVCNT							

Бит	Имя	Значение	Описание
31...16			Зарезервировано
15...8	DIVCNTDEF		<p>Делитель для задания скорости передачи данных в канал связи SpaceWire равной 10Мбит/с скорость передачи= $SPWCLK / DIVCNTDEF$, DIVCNTDEF не равно 0, где $SPWCLK$ – частота тактирования передатчика. Частота $SPWCLK$ должна быть равна частоте контроллера либо быть больше её в 2 раза. До перехода контроллера в состояние Run скорость передачи данных 10 Мбит/с. В состоянии Run скорость передачи данных определяется делителем DIVCNT</p>
7...0	DIVCNT		<p>Делитель для задания скорости передачи данных в канал связи SpaceWire скорость передачи= $SPWCLK / DIVCNT$, DIVCNT не равно 0 где $SPWCLK$- частота тактирования передатчика. Частота $SPWCLK$ должна быть равна частоте контроллера либо быть больше её в 2 раза</p>

7.16.10.4 TIME

Base ADDR=		0x4008_8000					Offset=		0x0000_000C						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															TICKIN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRLOUT		TIMEOUT						CTRLIN		TIMEIN					

Бит	Имя	Значение	Описание
31...17	-		Зарезервировано
16	TICKIN		При установке этого бита инициируется процесс передачи маркера времени с контрольными разрядами. После этого TICKIN автоматически сбрасывается в ноль.
15...14	CTRLOUT		Значение контрольных бит, принятых из сети последними
13...8	TIMEOUT		Значение маркера времени, принятого из сети последним
7...6	CTRLIN		Значение контрольных бит для передачи при установке бита TICKIN
5...0	TIMEIN		Значение маркера времени для передачи при установке бита TICKIN

7.16.10.5 PAUSE

Base ADDR=		0x4008_8000					Offset=		0x0000_0010						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															RESTIME

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTIME								DSCTIME							

Бит	Имя	Значение	Описание
31...19			Зарезервировано
18...8	RESTIME		Количество тактов частоты процессорного ядра необходимое на интервал работы счетчика 6.4 мкс.
7...0	DSCTIME		Количество тактов частоты процессорного ядра необходимое на интервал работы счетчика 850 нс.

7.16.10.6 FIFORX

Base ADDR=		0x4008_8000				Offset=		0x0000_0014								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DATA[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DATA[15:0]																

Бит	Имя	Значение	Описание
31...0	DATA		<p>В блоке приема из байтов данных формируются слова разрядности 32. При формировании слов первый поступивший байт размещается в разрядах 7:0, второй – в разрядах 15:8, третий – в разрядах 23:16 и четвертый – в разрядах 31:24. Распределение символов данных по разрядам слова данных производится по счетчику байт. В случае получения кода конца пакета данные выравниваются по границам 32-х разрядных слов путём добавления “лишних” байт.</p> <p>FIFO размером 16 слов.</p>

7.16.10.7 FIFOTX

Base ADDR=		0x4008_8000				Offset=		0x0000_0018								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DATA[31:16]																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DATA[15:0]																

Бит	Имя	Значение	Описание
31...0	DATA		<p>Блок передачи разбивает слова на отдельные байты. При этом из последовательности байтов в соответствии с информацией, содержащейся в дескрипторе, удаляются “лишние” байты – байты, добавленные для выравнивания пакетов по границам слов, и вставляются символы концов пакетов EOP или EEP. Блок передачи вначале передает в Linkinterface байт данных, находящийся в разрядах 7:0 слова, затем байт, находящийся в разрядах 15:8, затем байт, находящийся в разрядах 23:15, затем байт из разрядов 31:24 тридцатидвухразрядного слова.</p> <p>FIFO размером 16 слов.</p>

7.16.10.8 CNTRX_PACK

Base ADDR=		0x4008_8000				Offset=		0x0000_001C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOTNULLCNT															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTNULLCNT															

Бит	Имя	Значение	Описание
31...0	NOTNULLCNT		Содержимое 32-разрядного регистра SPW_CNTRX_PACK увеличивается на 1 каждый раз, когда из канала связи SpaceWire принимается символ конца пакета EOP, если ему предшествовал хотя бы один символ данных, что означает принятие пакета ненулевой длины. При записи (любых данных), значение регистра обнуляется.

7.16.10.9 CNTRX0_PACK

Base ADDR=		0x4008_8000				Offset=		0x0000_0020							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NULLCNT															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NULLCNT															

Бит	Имя	Значение	Описание
31...0	NULLCNT		Содержимое 32-разрядного регистра SPW_CNTRX0_PACK увеличивается на 1 при приеме символа EOP вслед за символом EOP, что эквивалентно принятию пакета нулевой длины. При записи (любых данных), значение регистра обнуляется.

7.16.10.10 NUM_TXDESC

Base ADDR=		0x4008_8000				Offset=		0x0000_0024							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												NUM_TXDESC			

Бит	Имя	Значение	Описание
31...4	-		Зарезервировано
3...0	NUM_TXDESC		Дескрипторы передатчика SPW_TXDESC0-SPW_TXDESC15 вычитываются из памяти, начиная с номера 0 и заканчивая номером 15, после этого процесс повторяется. Если дескриптор в 31-ом разряде не содержит единицу, то после передачи пакета, определяемого предыдущим дескриптором, процесс увеличения номера дескриптора останавливается. Процесс возобновляется после установки в единицу 31-го разряда дескриптора с номером, содержащимся в регистре SPW_NUM_TXDESC. После инициирования передачи пакета, 31-й разряд дескриптора автоматически сбрасывается в ноль, а значение регистра SPW_NUM_TXDESC увеличивается на единицу.

7.16.10.11 NUM_RXDESC

Base ADDR=		0x4008_8000				Offset=		0x0000_0028							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												NUM_RXDESC			

Бит	Имя	Значение	Описание
31...4	-		Зарезервировано
3...0	NUM_RXDESC		Дескрипторы приёмника SPW_RXDESC0-SPW_RXDESC15 заполняются в памяти, начиная с номера 0 и заканчивая номером 15, после этого процесс повторяется.

7.16.10.12 PHY_CNTR

Base ADDR=		0x4008_8000				Offset=		0x0000_002C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											TST[2]	TST[1]	TST[0]	TRIMR[7:6]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TRIMR[5:0]						EN_BNG	TRIM[2:0]			SELR	OUT_EN	.	EN_PHY	IR_UP	IR_DOWN	

Бит	Имя	Значение	Описание
31...21	-	0	Зарезервировано
20...18	TST[2:0]	000	Регистр задания тестового режима 000 – 001 – ... 111 –
17...10	TRIMR[7:0]	0	Подстройка внутреннего резистора источника опорного тока 1000_0000 Типовое значение опорного тока (10 мкА) 0000_0000 Минимальный опорный ток (5 мкА) 1111_1111 Максимальный опорный ток (15 мкА) <i>Для функционирования должно быть задано типовое значение</i>
9	EN_BNG	0	Сигнал разрешения работы блока REF для SPW 0 – блок выключен 1 – блок включен
8...6	TRIM[2:0]	0	Регистр цифрового тримминга источника опорного напряжения Типовое значение 100 (1,2 В). 000 – максимальное напряжение (1,25 В) 111 – минимальное напряжение (1,15 В)
5	SELR	0	Выбор резистора источника опорного тока 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен) 1 – используется внешний стабильный резистор на выводе SPW_EXTR на землю.
4	OUT_EN	0	Разрешение передачи приёмопередатчика 0 – выключен 1 – включен
3	-		Зарезервировано
2	EN_PHY	0	Разрешение работы приёмопередатчика 0 – выключен 1 – включен
1	IR_UP	0	Увеличение тока 0 – по умолчанию 1 – увеличить на 12.5%
0	IR_DOWN	0	Уменьшение тока 0 – по умолчанию 1 – снизить на 12.5%

7.16.10.13 RXDESCx

Base ADDR=		0x4008_8000				Offset=		0x0000_0040							
REG Name:								0x0000_007C							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDES	RXDES														
	RW0	R0	R0	R0											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDES															

Бит	Имя	Значение	Описание
31...0	RXDES		<p>Дескриптор пакета имеет следующую структуру:</p> <ul style="list-style-type: none"> · бит 31 – признак заполнения дескриптора действительными данными. До запуска приёма, все 31-е биты дескрипторов области приёма должны быть обнулены программно, записью единицы; контроллер не обнуляет 31-е биты, а только записывает '1' в 31-е биты принятых дескрипторов; · биты 30...29 – тип конца пакета: 01 – EOP, 10 – EEP; · биты 28...25 – не используются (0000); · биты 24...0 – размер пакета в байтах. <p>Дескриптор пакета формируется в счетчике принятых символов N-Char. При поступлении символа данных N-Char счетчик увеличивается на 1, при поступлении символа конца пакета значение счетчика переписывается в разряды 24...0 текущего дескриптора, а сам счетчик сбрасывается в 0.</p> <p>Всего 16 регистров.</p>

7.16.10.14 TXDESCx

Base ADDR=		0x4008_8000				Offset=		0x0000_0080							
REG Name:								..							
								0x0000_00BC							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDES															
RW0	RW0	R0	R0												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDES															

Бит	Имя	Значение	Описание
31...0	TXDES		<p>Дескриптор пакета имеет следующую структуру:</p> <ul style="list-style-type: none"> · бит 31 – признак заполнения дескриптора действительными данными. · Бит 31 автоматически сбрасывается в ноль после инициирования передачи пакета. · биты 30...29 – тип конца пакета: 01 – EOP, 10 – EEP; · биты 28...25 – не используются (0000); · биты 24...0 – размер пакета в байтах. <p>Данные из SPW_FIFOTX в блок Linkinterface выдаются побайтно. Преобразование 32-хразрядных слов в байты осуществляется в блоке преобразования под управлением счетчика переданных байт. В счетчик заносится размер пакета из разрядов 24...0 дескриптора передаваемого пакета. После передачи каждого байта этот счетчик уменьшается на 1. По достижении счетчиком значения 0, в поток передаваемых данных вставляется символ конца пакета EOP или EEP в зависимости от значения разрядов 30...29 дескриптора, а в счетчик заносится размер следующего передаваемого пакета из следующего дескриптора.</p> <p>При установке размера пакета в байтах равного нулю происходит передача пакета нулевой длины. Максимально возможный размер пакета в байтах 0x1FFFFFFE.</p> <p>Всего 16 регистров.</p>

7.17 Контроллер таймеров общего назначения (TIMER32x)

Все блоки таймеров выполнены на основе 32-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного предделителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный (сначала прямой до определенного значения, а затем обратный).

Каждый из четырех таймеров микроконтроллера содержит 32-битный счетчик, 16-битный предделитель частоты и 4-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизацией, внешними сигналами или другими таймерами.

Помимо составляющего основу таймера счетчика, в каждый блок таймера также входит четырехканальный блок захвата/сравнения. Данный блок выполняет, как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры имеют в своем составе 4 канала схем захвата, ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания и запросы DMA.

Особенности:

- 32-битный вверх, вниз, вверх/вниз счетчик;
- 16-разрядный программируемый предварительный делитель частоты;
- до четырех независимых 32-битных каналов захвата на один таймер. Каждый из каналов захвата может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала. В случае захвата имеется дополнительная возможность генерировать прерывание и/или запрос DMA;
- четыре 32-битных регистра сравнения (совпадения), которые позволяют осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание и/или запрос DMA при совпадении;
- имеется до четырёх внешних выводов, соответствующих регистрам совпадения со следующими возможностями:
 - сброс в НИЗКИЙ уровень при совпадении;
 - установка в ВЫСОКИЙ уровень при совпадении;
 - переключение (инвертирование) при совпадении;
 - при совпадении состояние выхода не изменяется;
 - переключение при некотором условии.

7.17.1 Функционирование

Таймер предназначен для того, чтобы подсчитывать циклы периферийной тактовой частоты F_{dts} или какие-либо внешние события и произвольно генерировать прерывания, запросы DMA или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются восьмью регистрами совпадения. Кроме того, в микроконтроллере имеются четыре входа захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание или запрос DMA.

Структурная схема блока таймера представлена на рисунке .

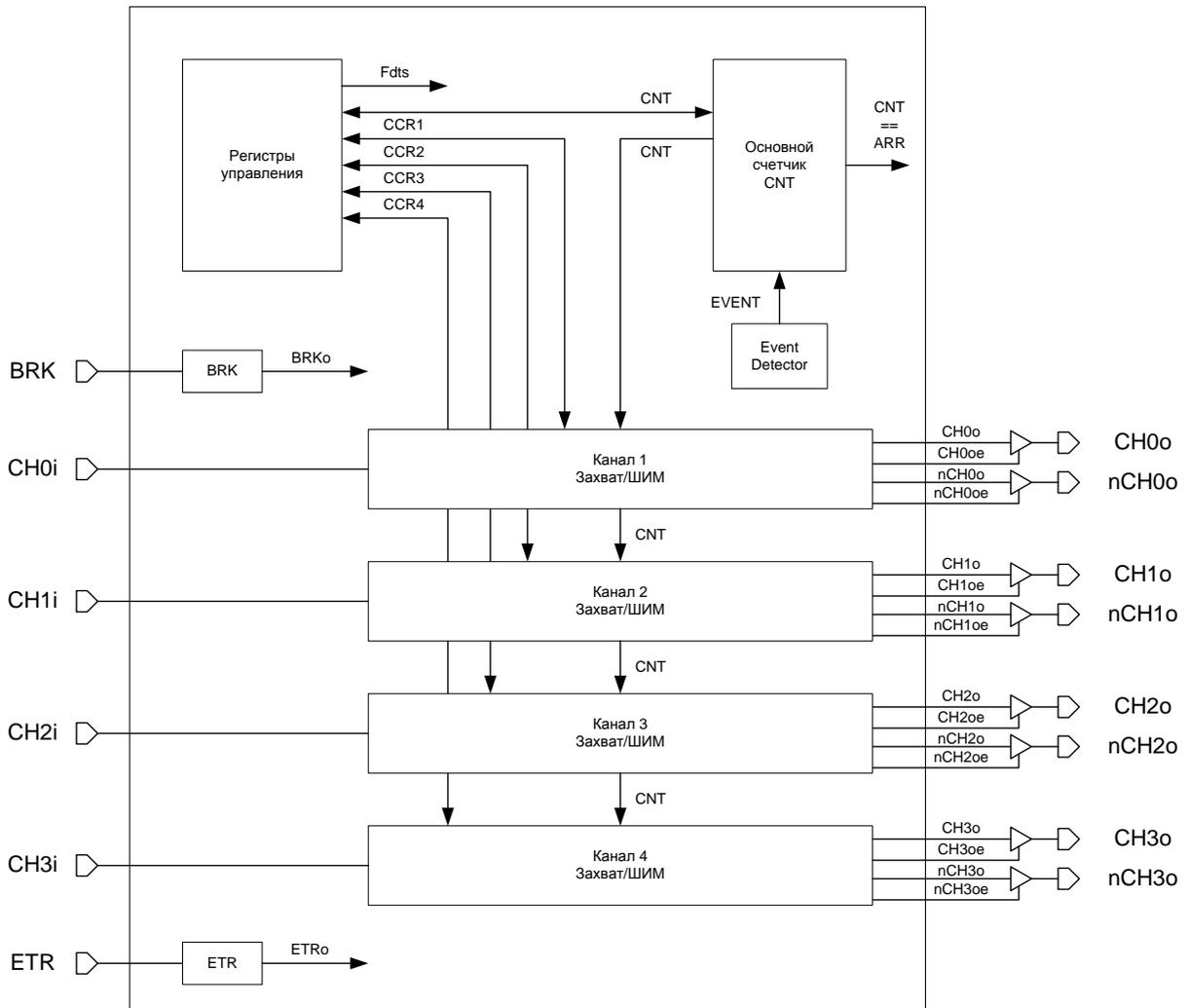


Рисунок 86 – Структурная схема таймера

Таймер содержит основной 32-х битный счетчик CNT, блок регистров управления и четыре канала схем Захвата/ШИМ.

Таймер позволяет работать в режимах:

- таймер;
- расширенный таймер, с объединением нескольких таймеров;
- схема захвата;
- схема ШИМ.

7.17.2 Инициализация таймера

Перед началом работы с таймерами в первую очередь должны быть включены тактовые сигналы. Параметры задаются в блоке контроллера тактовых чистот (CLKCNTR).

Для обращения к регистрам блока необходимо в регистре PER0_CLK блока CLKCNTR установить бит разрешения тактирования интерфейса управления блока – бит MDR_TMRn_EN. В регистре TIMn_CLK блока CLKCNTR необходимо установить бит EN_CLK, чтобы разрешить тактирование определенного таймера, и задать коэффициент деления DIV тактовой частоты MAX_CLK для каждого таймера.

После подачи тактовых сигналов на блок таймера можно приступить к работе с ним.

7.17.3 Режим таймера

Таймеры построены на базе 32-битного счетчика, объединенного с 16-битным предварительным делителем. Скорость счета таймера зависит от значения, находящегося в регистре делителя.

Счетчик может считать вверх, вниз или сначала вверх и затем вниз.

Базовый блок таймера включает в себя:

- основной счетчик таймера (TIMx_CNT);
- делитель частоты при счете основного счетчика (TIMx_PSC);
- основание счета основного счетчика (TIMx_ARR).
- Сигналом для изменения CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо события на линиях TxCHO данного счетчика.

Чтобы запустить работу основного счетчика необходимо задать:

- начальное значение основного счетчика таймера – TIMx_CNT;
- значение предварительного делителя счетчика – TIMx_PSG, при этом основной счетчик будет считать на частоте $CLK = TIMx_CLK / (PSG + 1)$;
- значение основания счета для основного счетчика TIMx_ARR;
- режим работы счетчика TIMx_CNTRL:
 - выбрать источник события переключения счетчика EVENT_SEL;
 - режим счета основного счетчика CNT_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
 - направление счета основного счетчика DIR;
- разрешить работу счетчика CNT_EN;

По событиям совпадения значения основного счетчика со значением нуля или значением основания счета генерируется прерывание и запроса DMA, которые могут быть замаскированы.

7.17.4 Режимы счета

Счет вверх: CNT_MODE = 00, DIR = 0

```
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера
```

```
//Настраиваем работу основного счетчика
```

```
TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика
```

```
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
```

```
TIMx->TIMx_ARR = 0x00000013; //Основание счета
```

```
//Разрешение работы таймера.
```

```
TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK.
```

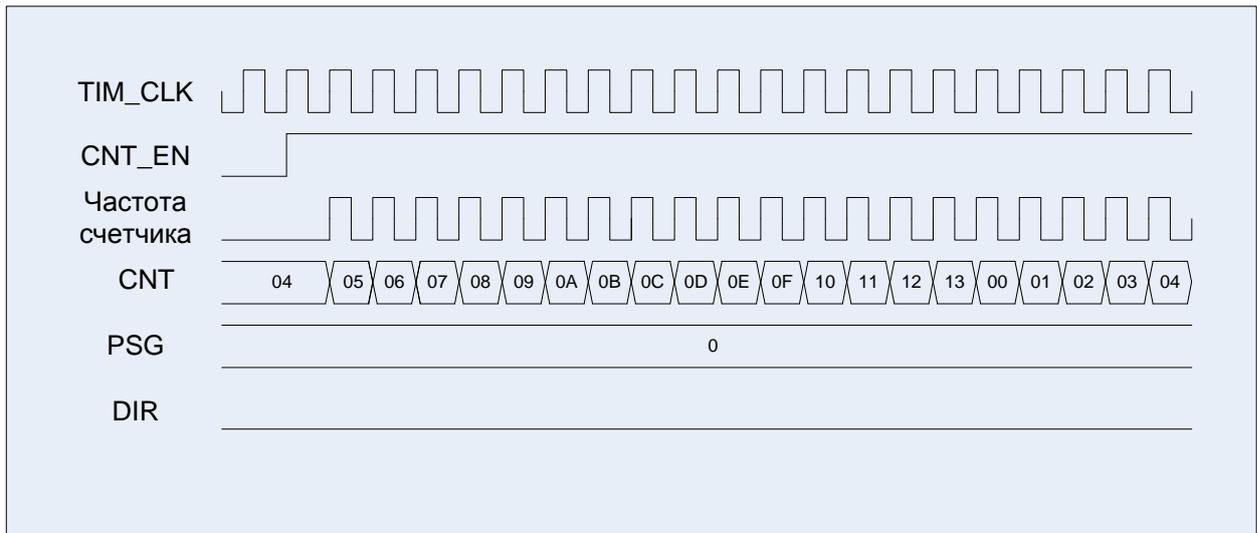


Рисунок 87 – Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

```

Счет вниз: CNT_MODE = 00, DIR = 1
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x00000013; //Основание счета

//Разрешение работы таймера.
TIMx->TIMx_CNTRL = 0x00000009; //Счет вниз по TIM_CLK.
    
```

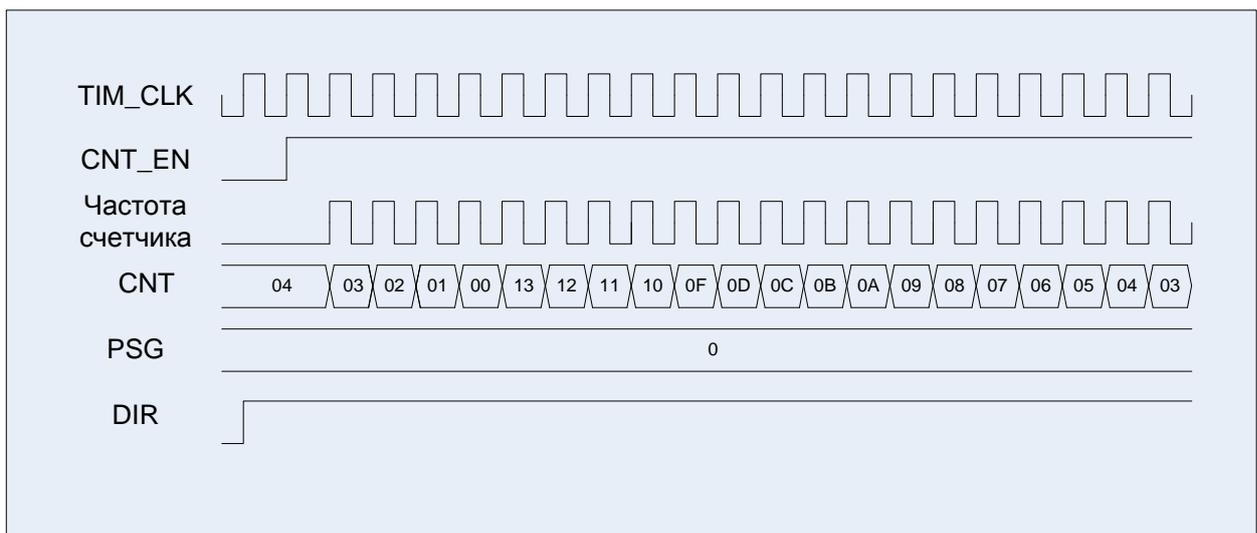


Рисунок 88 – Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

Счет вверх/вниз: CNT_MODE = 01, DIR = 0

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000013; //Основание счета

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000041; //Счет вверх/вниз по TIM_CLK.



Рисунок 89 – Диаграммы работы таймера, счет вверх/вниз, сначала вверх

Счет вверх/вниз: CNT_MODE = 01, DIR = 1

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000013; //Основание счета

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000049; //Счет вверх/вниз по TIM_CLK.

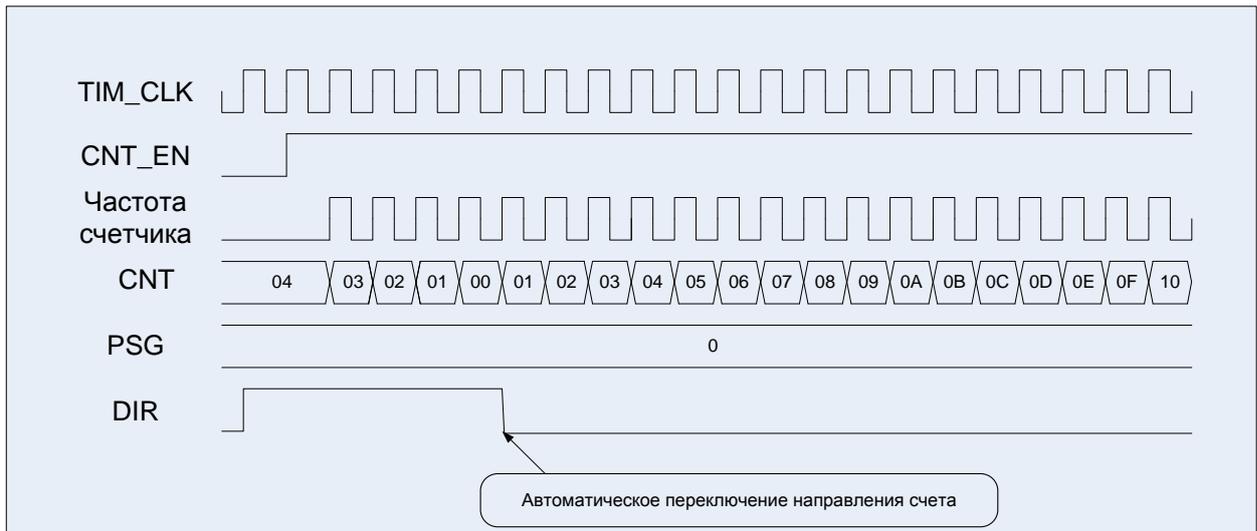


Рисунок 90 – Диаграммы работы таймера, счет вверх/вниз, сначала вниз

7.17.5 Источник событий для счета

Внутренний тактовый сигнал (TIM_CLK).

События в других счетчиках (CNT==ARR в таймере X).

Внешний тактовый сигнал режим 1: События на линиях TxCHO данного счетчика.

Внешний тактовый сигнал режим 2: События на линиях TxCHO данного счетчика.

Внешний тактовый сигнал режим 3: События на входе ETR данного счетчика.

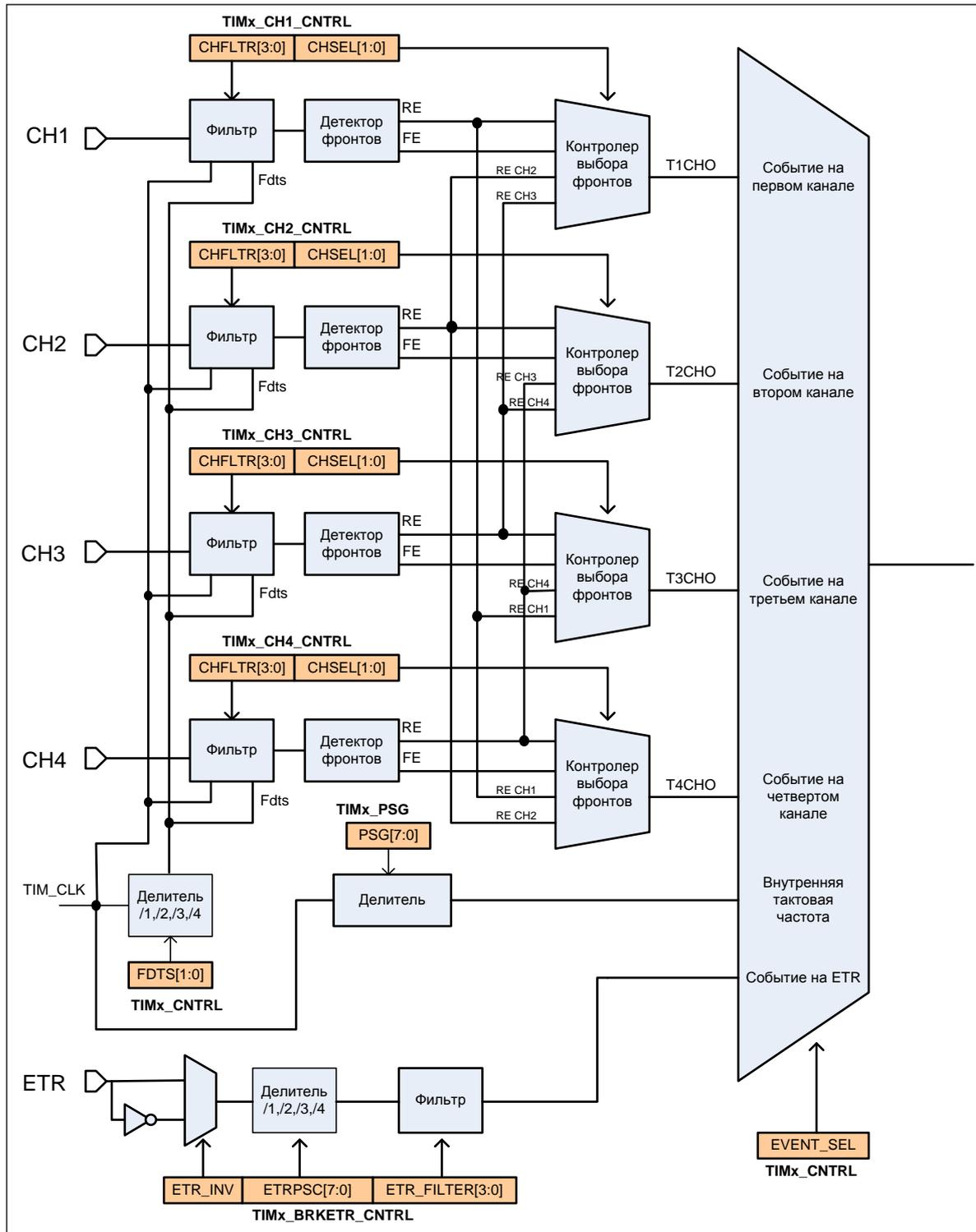


Рисунок 91 – Структурная схема формирования события для счета

7.17.5.1 Внутренний тактовый сигнал (TIM_CLK)

Этот режим выбирается, когда CNT_MODE = 0x, EVENT_SEL = 0000. Для запуска этого режима необходимо задать начальное значение основного счетчика, значение предварительного делителя основного счетчика, основание счета для основного счетчика и задать режим работы в регистре TIMx_CNTRL. Значения регистров TIMx_CNT, TIMx_PSG и TIMx_ARR можно изменять даже во время работы счетчика, при этом их значения вступают в силу по CNT = ARR или CNT = 0, в зависимости от направления счета. Значение регистра основание счета (TIMx_ARR) может вступить в силу мгновенно после записи его в регистр при

условии установленного поля $ARRB_EN = 1$ (регистр $TIMx_CNTRL$). Если значение предварительного делителя основного счетчика не равно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM_CLK до тех пор, пока не достигнет значения, находящегося в регистре делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера измениться на 1 и снова начинается счет. Поле DIR определяет, в какую сторону будет меняться значение счетчика: $DIR = 0$ – счетчик считает вверх (Рисунок 92), $DIR = 1$ – счетчик считает вниз (Рисунок 93). Если $CNT_MODE = 00$, то направление счета определяется полем DIR , если $CNT_MODE = 01$, счетчик считает вверх/вниз с автоматическим изменением DIR (Рисунок 94).

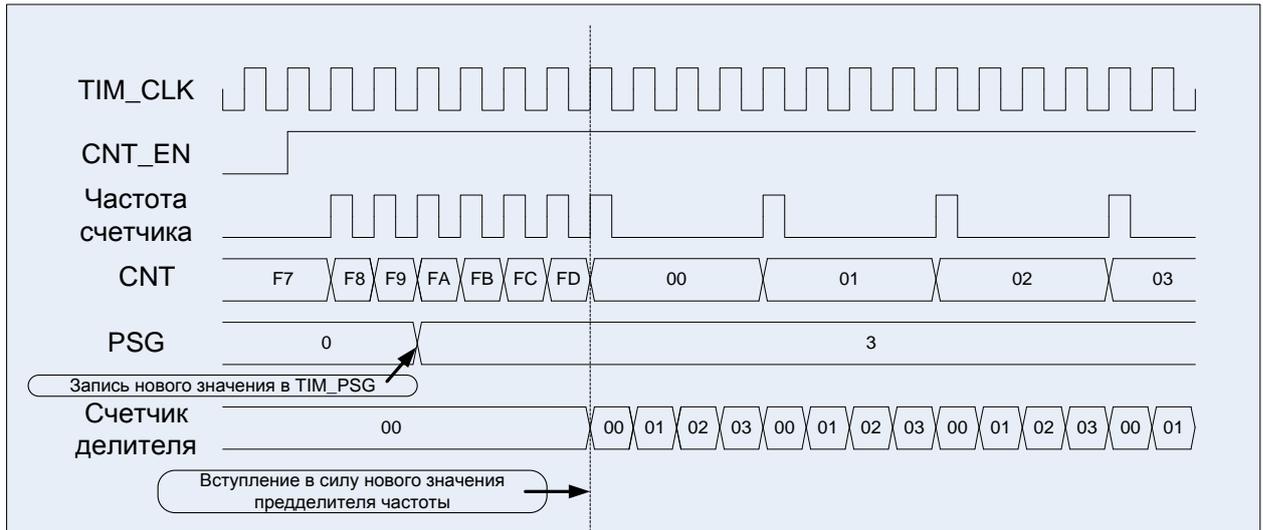


Рисунок 92 – Диаграммы работы счетчика: счет вверх ($CNT_MODE = 00$, $EVENT_SEL = 0000$, $DIR = 0$)

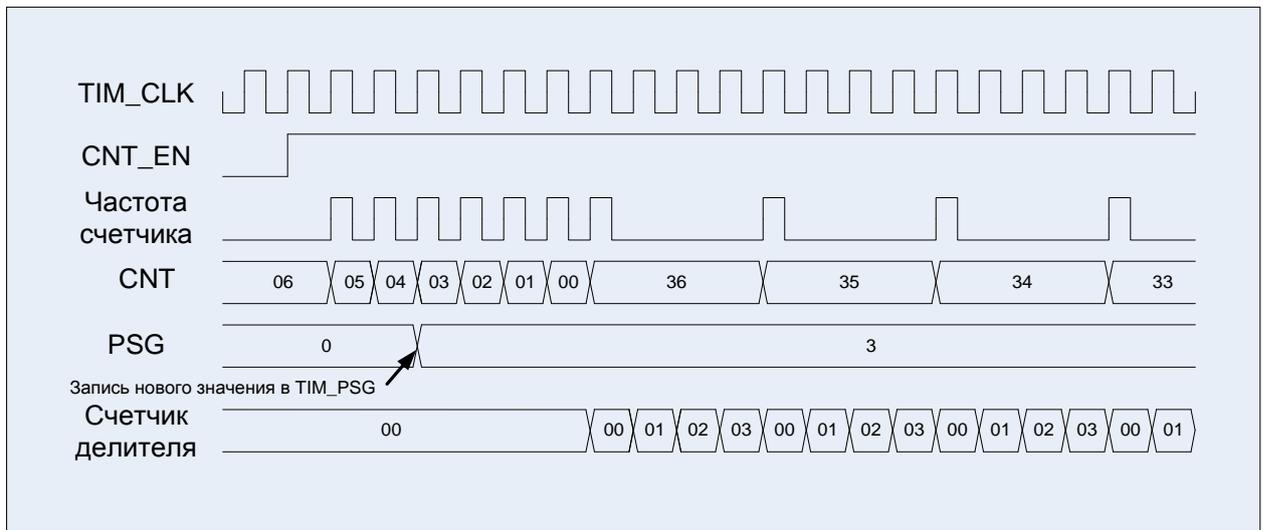


Рисунок 93 – Диаграммы работы счетчика: счет вниз ($CNT_MODE = 00$, $EVENT_SEL = 0000$, $DIR = 1$)

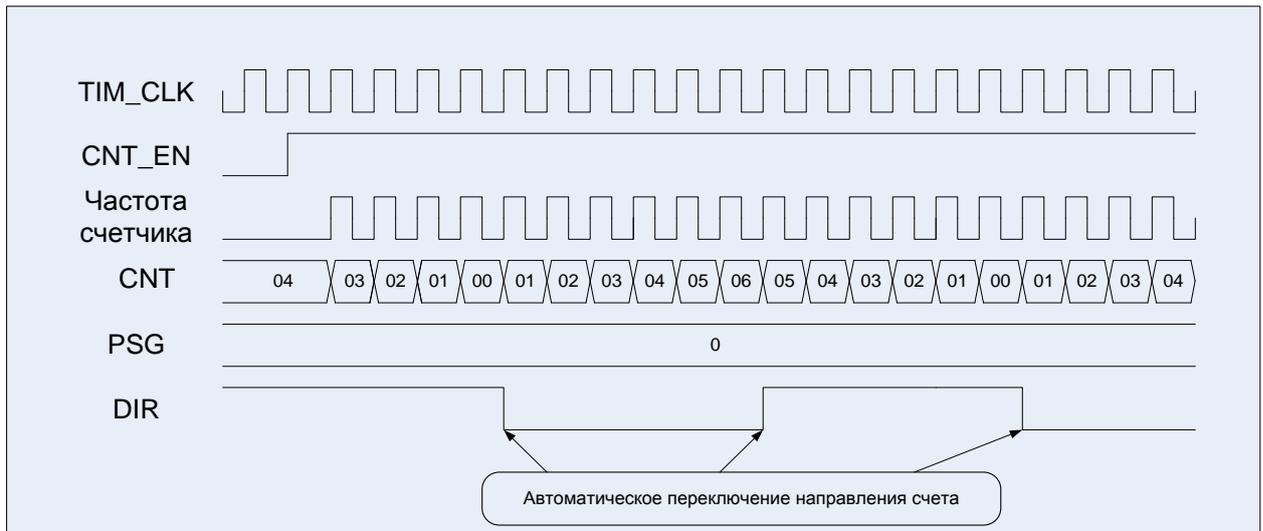


Рисунок 94 – Диаграммы работы счетчика: счет вниз/вверх (CNT_MODE = 01, EVENT_SEL = 0000, DIR = 1)

7.17.5.2 События в других счетчиках (CNT==ARR в таймере X)

Каждый из блоков таймеров полностью независим друг от друга, но у них предусмотрена возможность синхронизированной друг с другом работы. Это позволяет создавать более сложные массивы таймеров, которые работают полностью автономно и не требуют написания какого-либо кода программы для выполнения сложных временных функций.

У каждого таймера имеются входы запуска от других таймеров, а также внешние входы, связанные с выводами блоков захвата/сравнения.

У каждого из блоков таймеров имеется выход запуска, который соединен с входами других таймеров. Синхронизация таймеров возможна в нескольких различных режимах. Ниже показан пример каскадного соединения счетчиков (Рисунок 95).

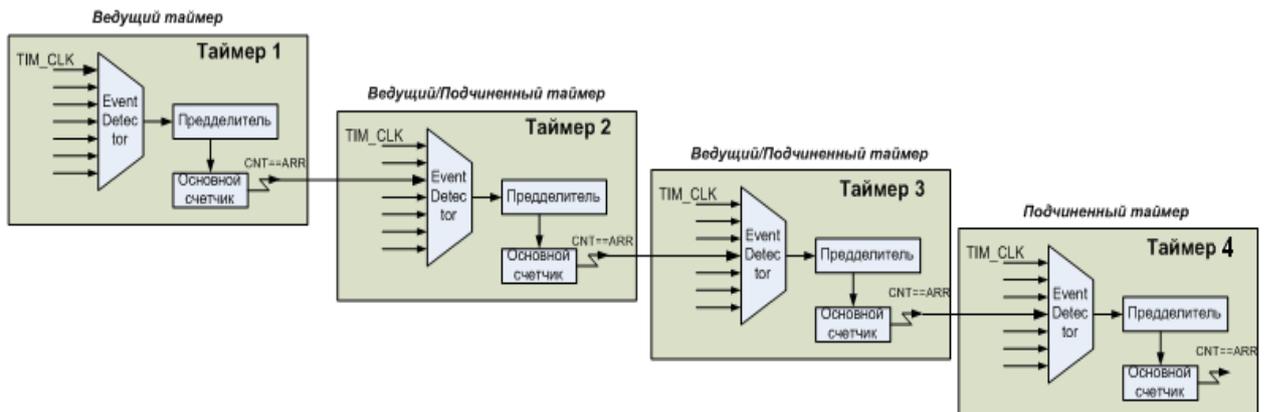
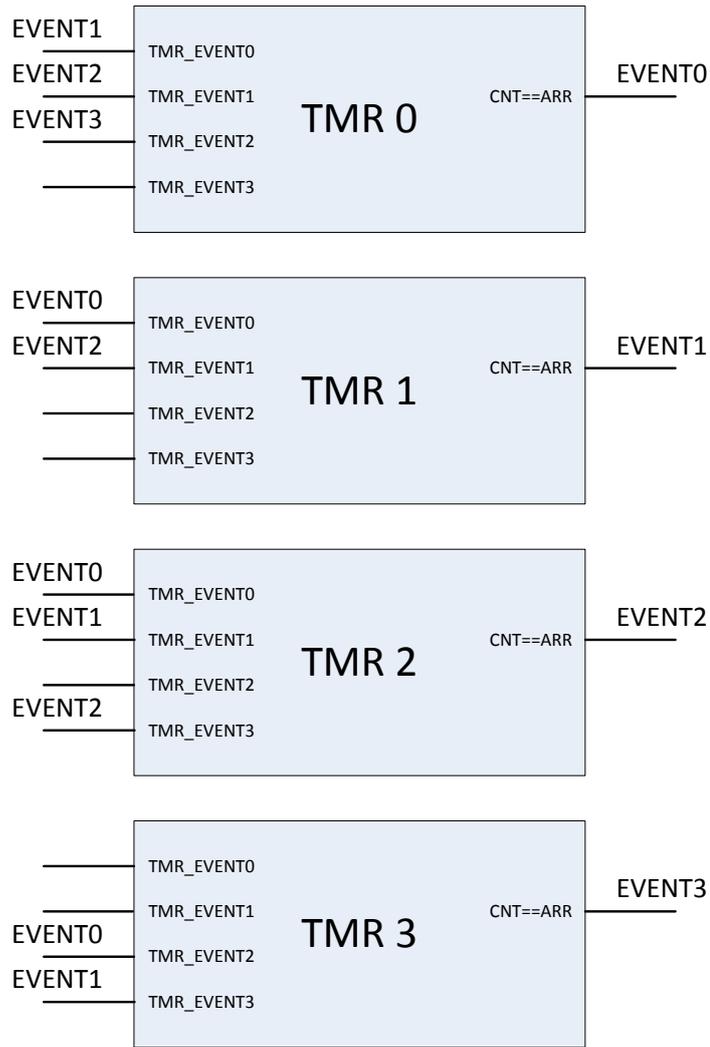


Рисунок 95 – Пример каскадного соединения таймеров



Примечание – Необозначенные входы EVENT не подключены и событий от них не возникает.

Рисунок 96 – Варианты каскадного объединения таймеров

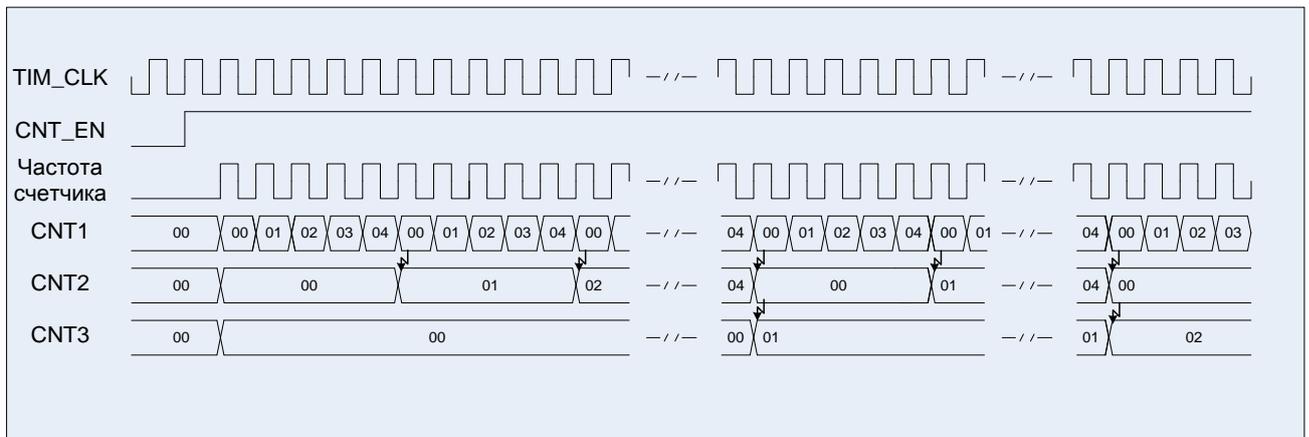


Рисунок 97 – Диаграммы работы трех таймеров в каскаде
 DIR_1, DIR_2, DIR_3 = 0; EVENT_SEL_1 = 0000, EVENT_SEL_2 = 0001, EVENT_SEL_3 = 0010;
 CNT_MODE_1, CNT_MODE_2, CNT_MODE_3 = 00

7.17.5.3 Внешний тактовый сигнал режим 1. События на линиях TxCHO данного счетчика

Этот режим выбирается, когда EVENT_SEL = 01xx в регистре TIMx_CNTRL. Счетчик может считать по положительному фронту или по отрицательному фронту на выбранном входе или по положительному фронту на других каналах (см. рис.). На входе сигнала стоит фильтр, с помощью которого можно контролировать длительность сигнала, для фильтрации можно использовать как сигнал TIM_CLK, при этом может быть идентифицированная длительность 1, 2, 4, 8 TIM_CLK, также можно при фильтровании использовать производную от TIM_CLK частоту FDTS. Частота семплирования данных задается в регистре TIMx_CNTRL в поле FDTS.

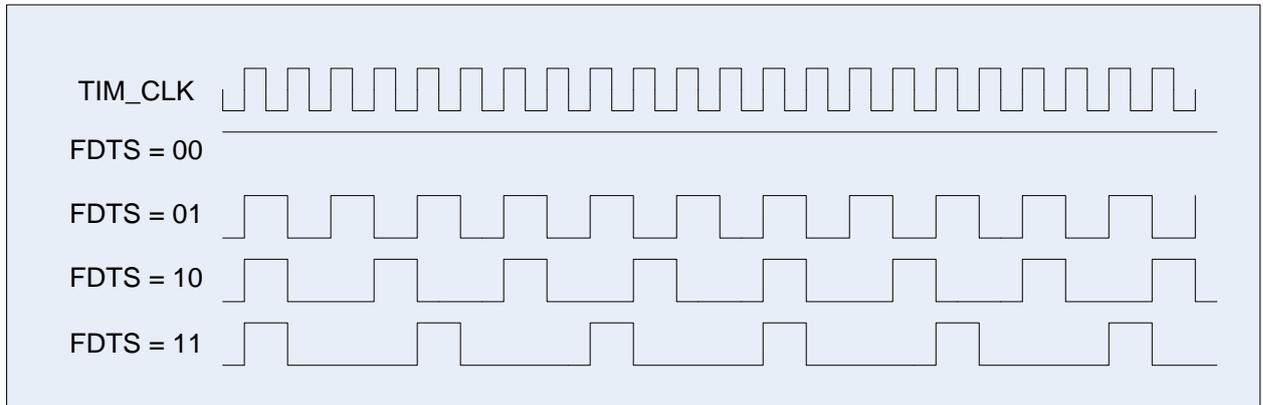


Рисунок 98 – Диаграммы возможных частот семплирования данных (FDTS)

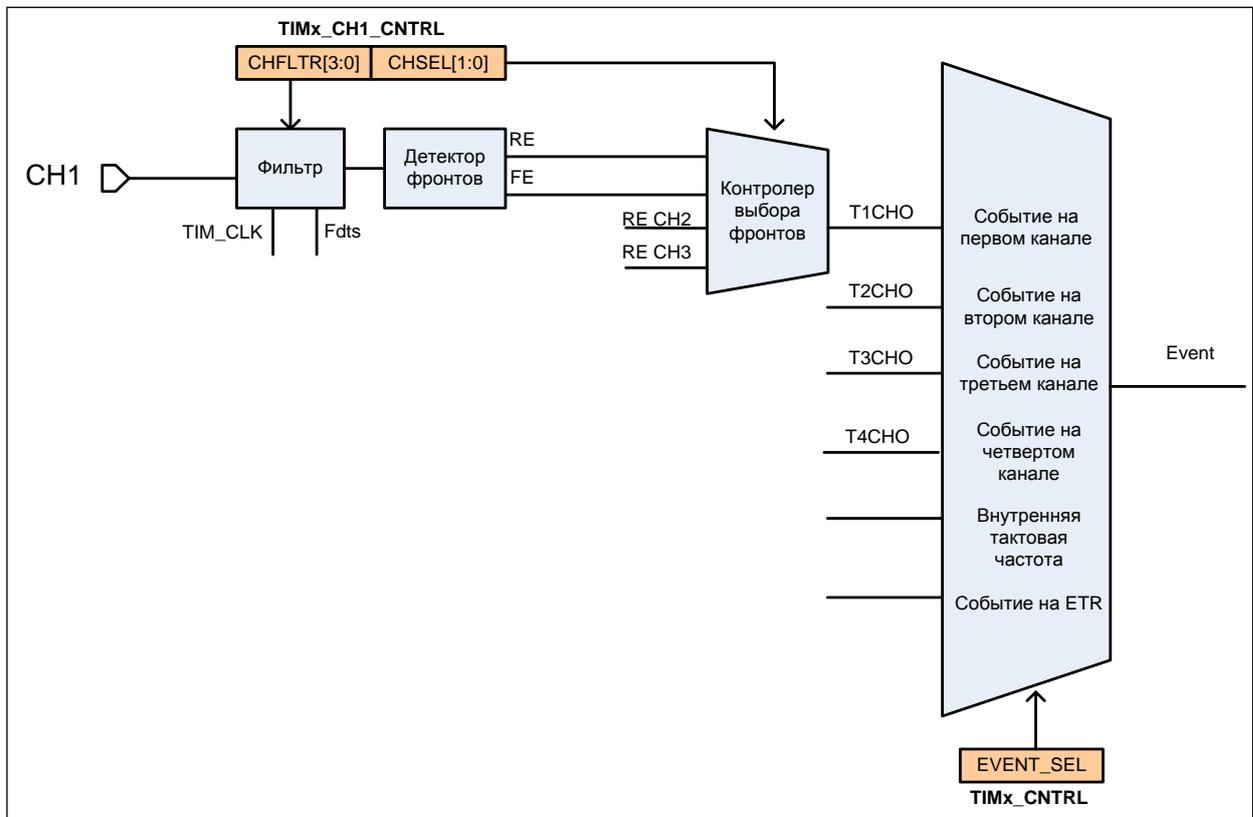


Рисунок 99 – Тактирование с входа первого канала

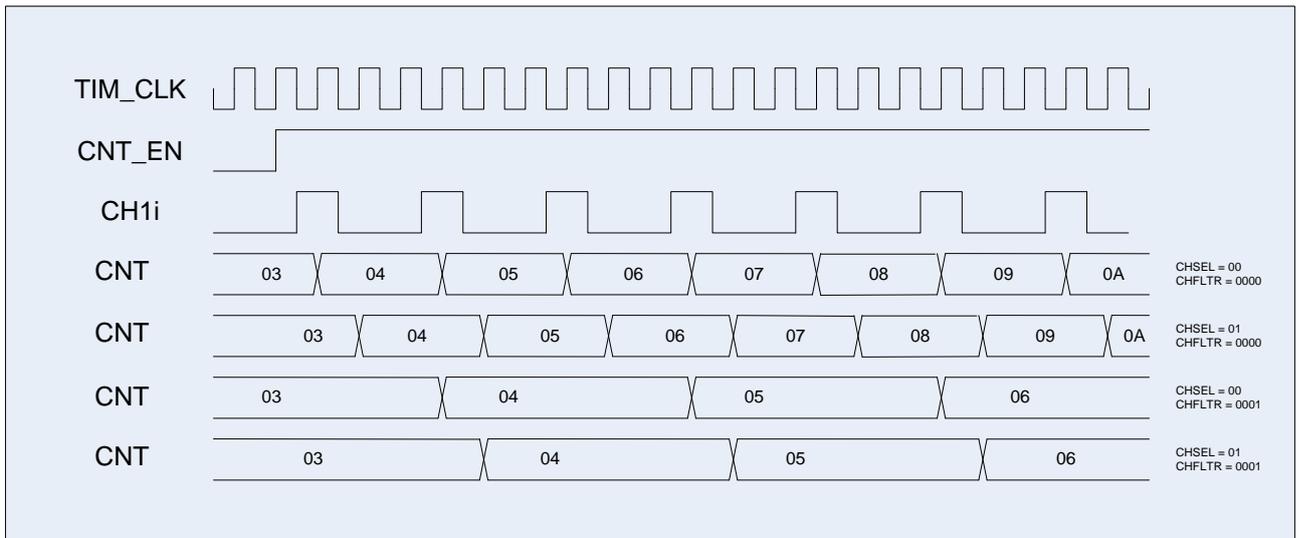


Рисунок 100 – Диаграмма внешнего тактирования с разными вариантами фильтра

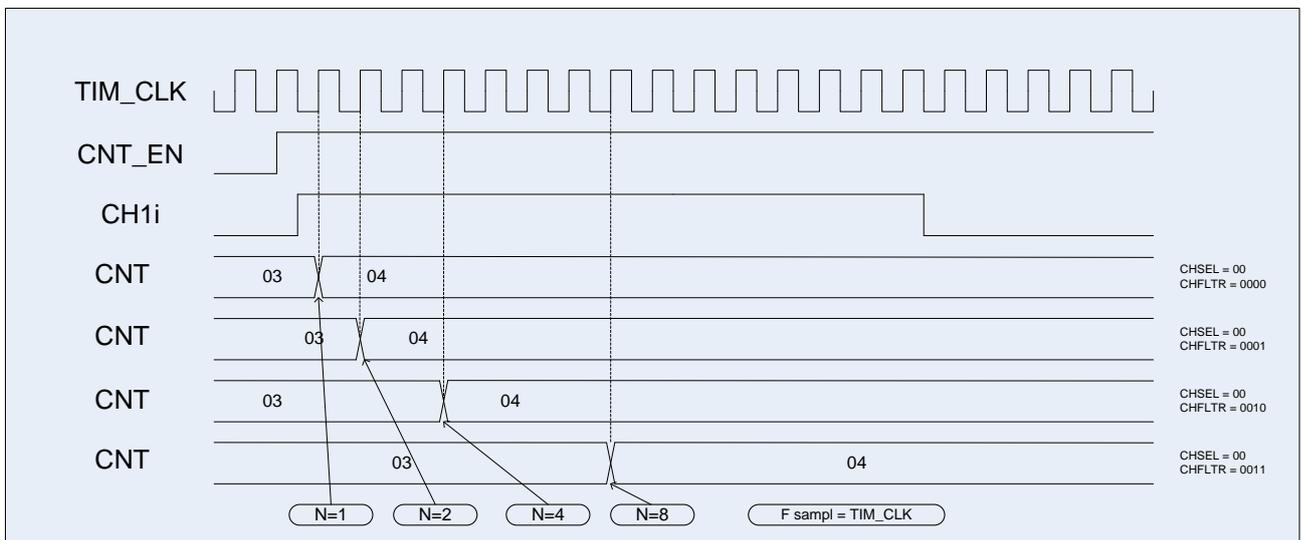


Рисунок 101 – Диаграмма внешнего тактирования с разными вариантами фильтра

7.17.5.4 Внешний тактовый сигнал режим 2. События на входе ETR данного счетчика

Этот режим выбирается, когда EVENT_SEL = 1000 в регистре TIMx_CNTRL. В регистре TIMx_BRKETR_CNTRL можно настроить коэффициент деления 2, 4 или 8 (ETRPSC) данного входа тактовой частоты, а также использовать инверсию входа.

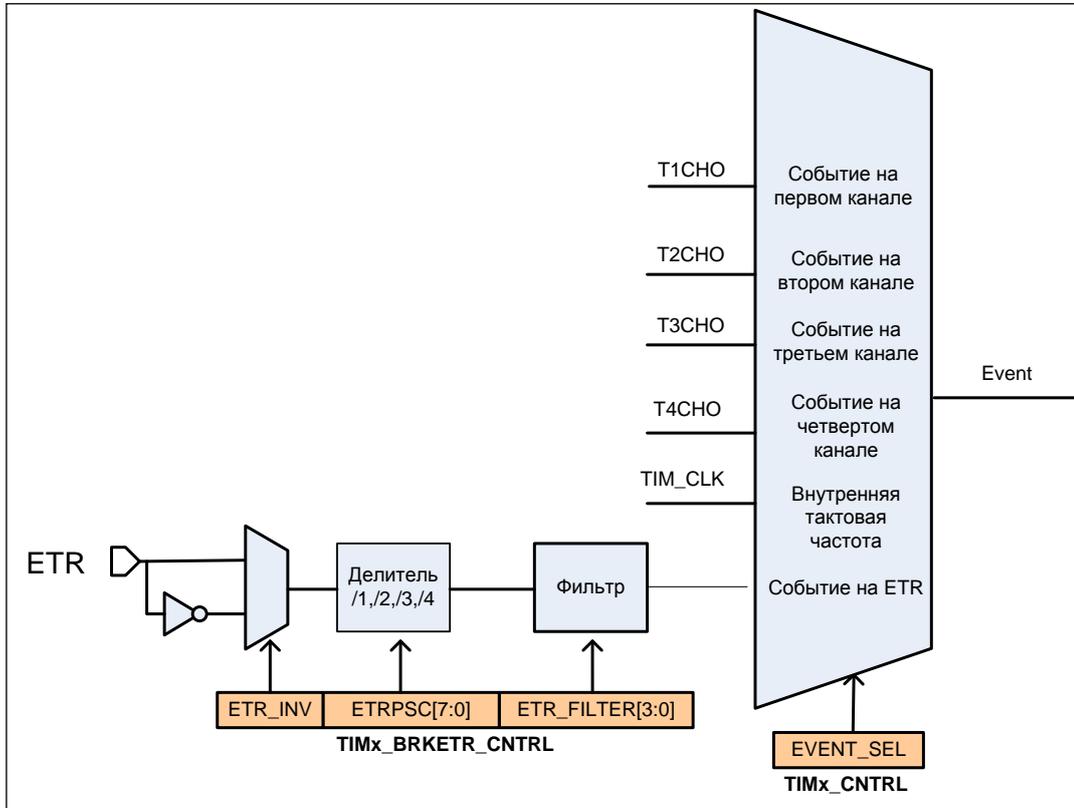


Рисунок 102 – Схема тактирования сигналом с входа ETR

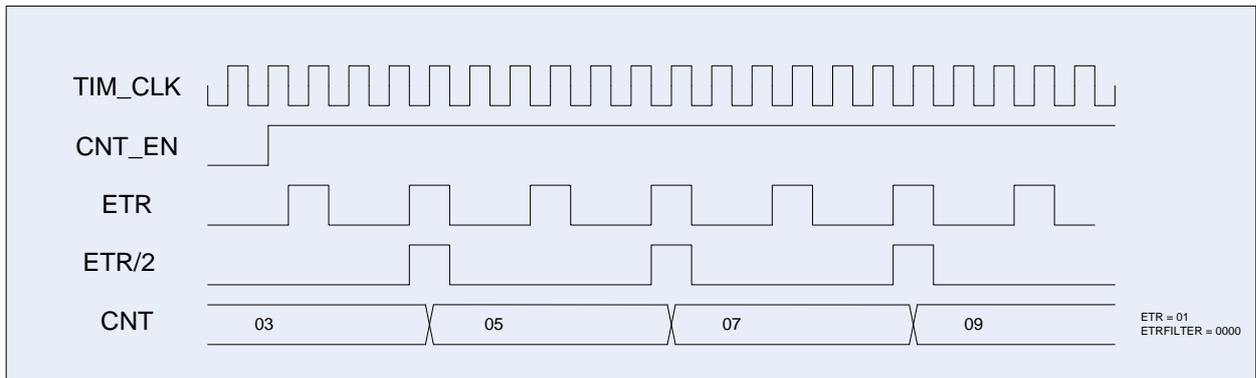


Рисунок 103 – Диаграмма тактирования сигналом с входа ETR

7.17.6 Режим захвата

Структурная схема блока захвата представлена на рисунке ниже (Рисунок 104).

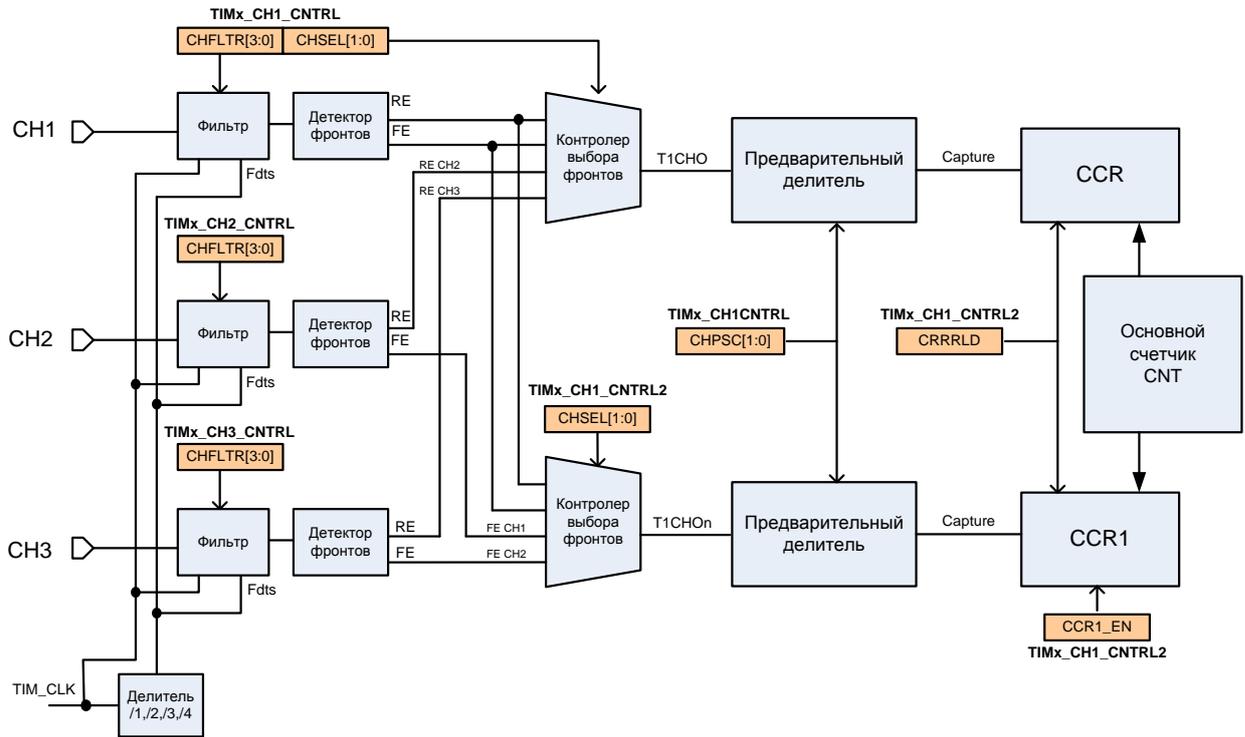


Рисунок 104 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 1 в поле CAPnPWM. Для регистрации событий по линии CHx используется схема регистрации событий. Входной сигнал фиксируется в таймере с частотой Fdts, или TIM_CLK. Также вход может быть настроен на прием импульсов заданной длины за счет конфигурирования блока FILTER. На выходе блока Фильтр вырабатывается сигнал положительного перепада и отрицательного перепада. На блоке MUX производится выбор используемого для Захвата сигнала, между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. После блока MUX предварительный делитель может быть использован для фиксации каждого события, каждого второго, каждого четвертого и каждого восьмого события. Выход предварительного делителя является сигналом Capture для регистра CCR, и Capture1 для регистра CCR1 при этом в регистры CCR и CCR1 записывается текущее значение основного счетчика CNT.

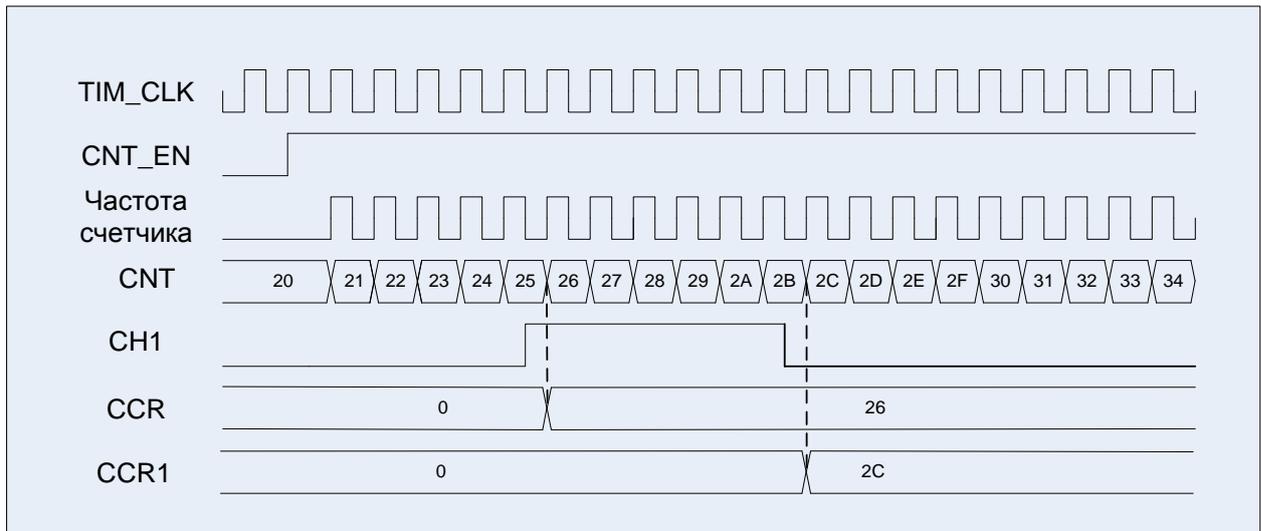


Рисунок 105 – Диаграмма захвата события с входа первого канала

На рисунке показан пример захвата значения основного счетчика в регистр CCR по положительному фронту на входе канала, а в регистр CCR1 по отрицательному фронту на входе канала. В регистре TIMx_IE можно разрешить выработку прерываний по событию захвата на определенном канале, а в регистре TIMx_DMA_RE можно разрешить формирование запросов DMA.

7.17.7 Режим ШИМ

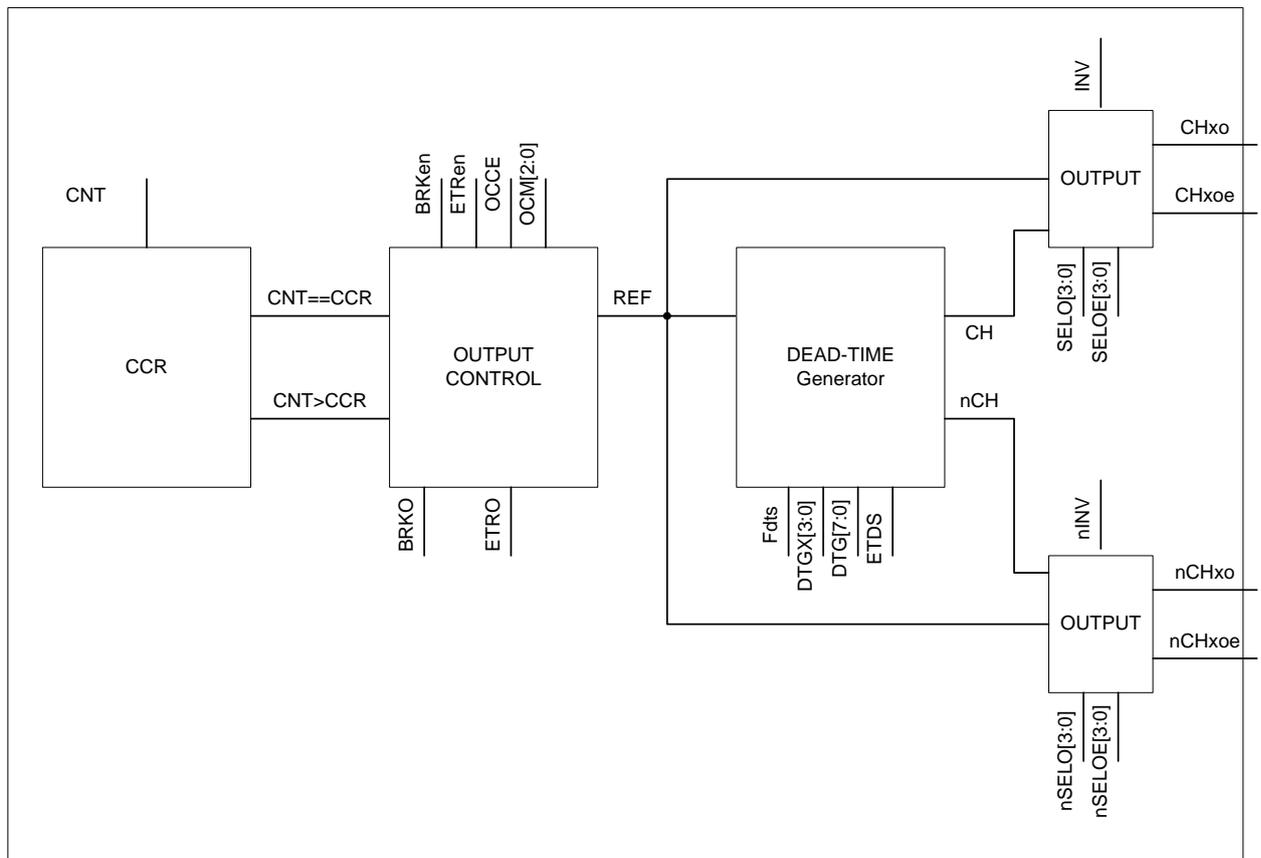


Рисунок 106 – Структурная схема блока сравнения

Для включения режима сравнения для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 0 в поле CAPnPWM. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в

регистре CCR и основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. Полученный сигнал может без изменения выдаваться на выходы CHxO и nCHxO. Либо с применением схемы DEADTIME Generator формируются управляющие сигналы с мертвой зоной. У каждого канала есть два выхода: прямой и инверсный. Для каждого выхода формируется как сигнал для выдачи, так и сигнал разрешения выдачи, т.е. если выход канала должен всегда выдавать тот или иной уровень, то на выводе разрешения выдачи CHxOE (для прямого) и на CHxNOE (для инверсного) должны формироваться "1". Если канал работает на вход (например, режим захвата), то там всегда должен быть "0" для прямого канала. Сигналы OE работают по тем же принципам, что и просто выходные уровни, но у них есть собственные сигналы разрешения вывода SELOE и nSELOE, в которых можно выбрать постоянный уровень, либо формируемый на основании REF.

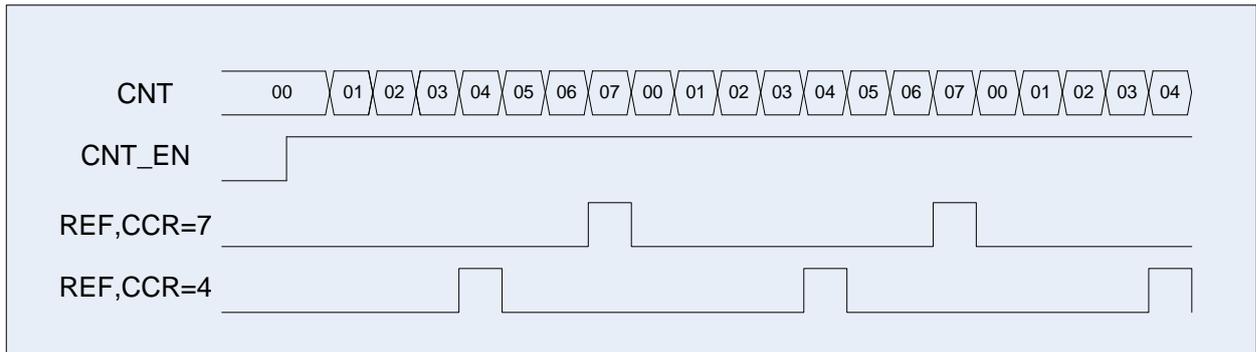


Рисунок 107 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

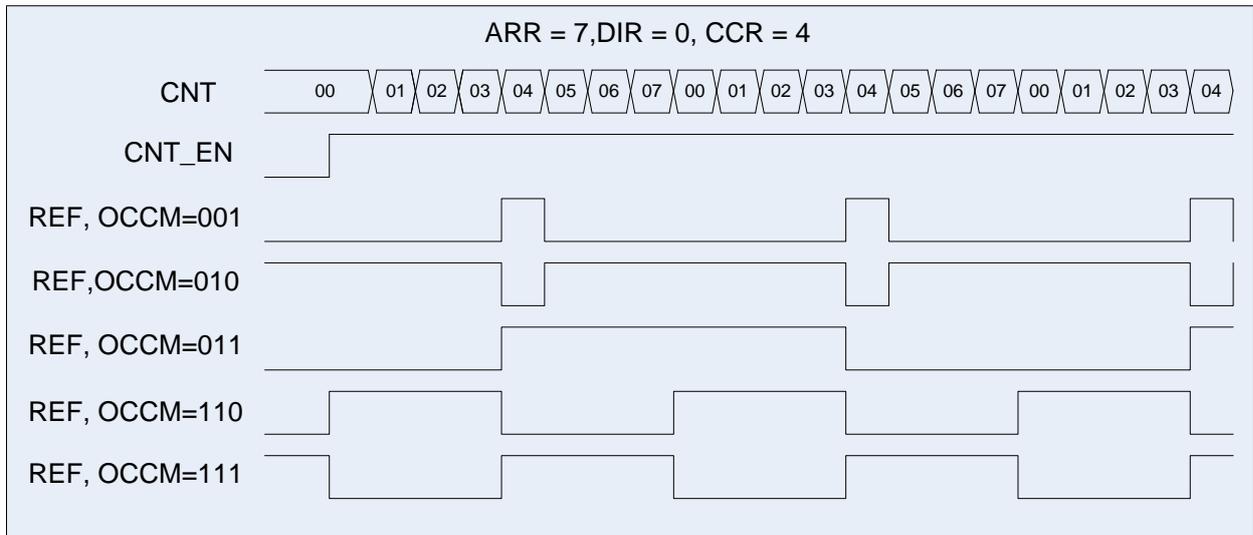


Рисунок 108 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

Сигнал REF может быть очищен с использованием внешнего сигнала с входа ETR или внешнего триггерированного по PCLK сигнала с входа BRK.

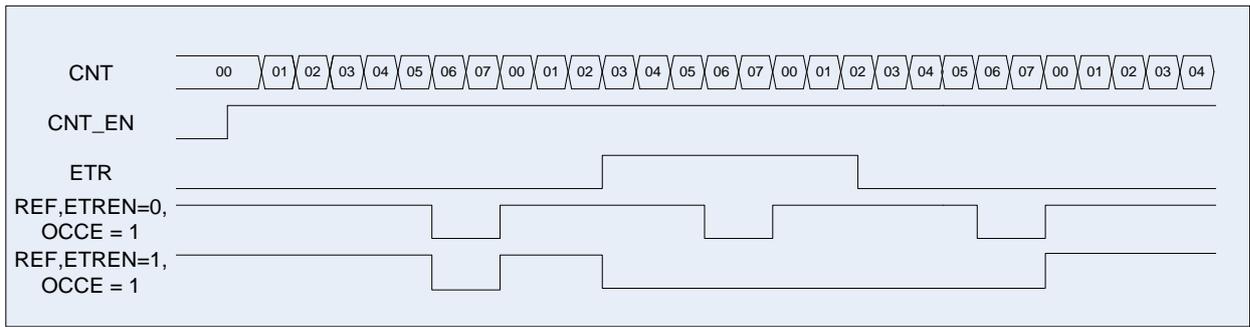


Рисунок 109 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 0

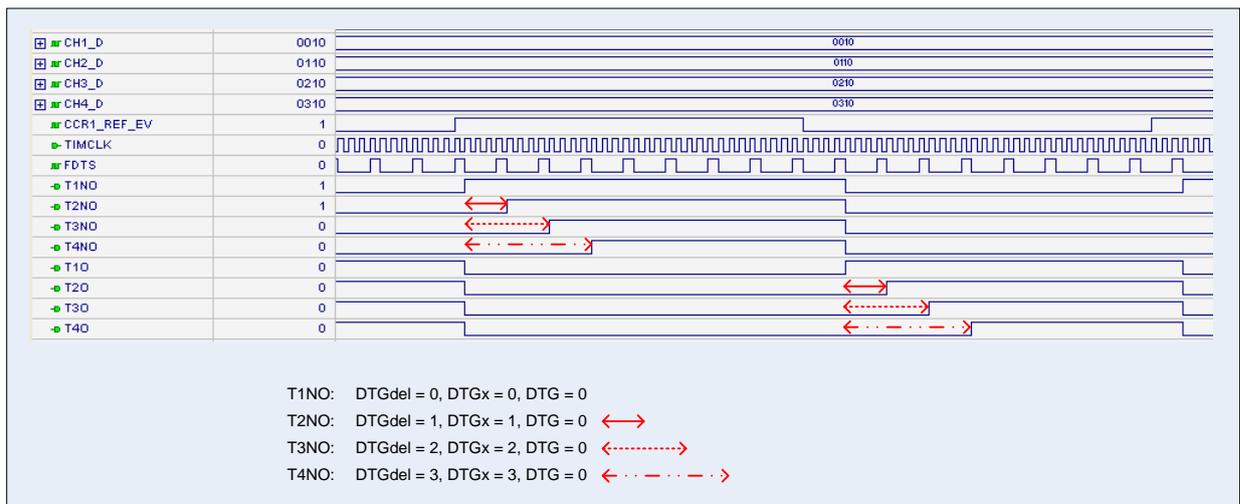


Рисунок 110 – Диаграмма работы схемы DTG

Если CCR1_EN = 1, тогда значение основного счетчика CNT сравнивается со значениями регистров CCR и CCR1, и в зависимости от запрограммированного формата выработки сигнала REF (регистры управления каналами таймера TIMx_CHy_CNTRL поле OCCM) будет формироваться сигнал соответствующей формы.

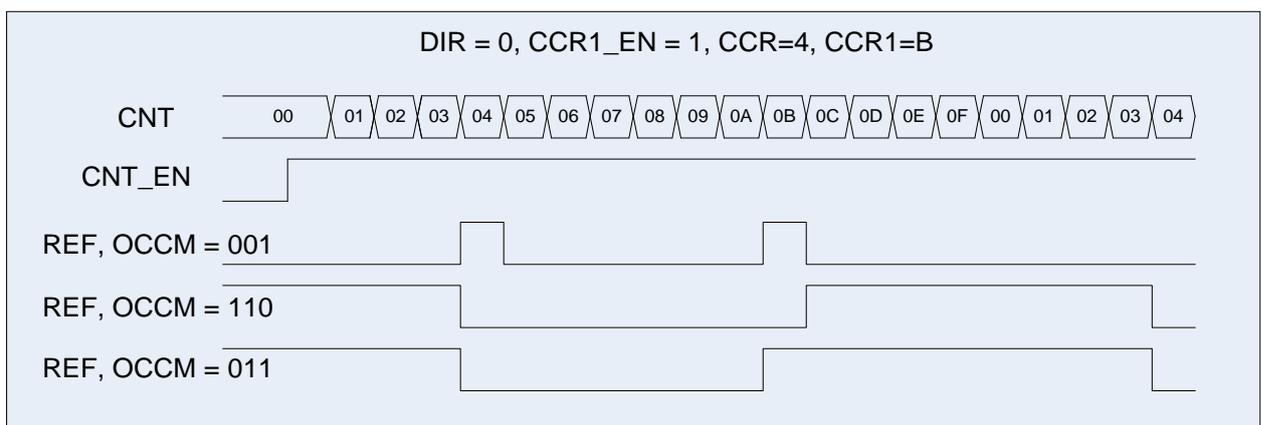


Рисунок 111 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 1

При записи новых значений CCR и CCR1, если установлен бит CRRRLD, то регистры CCR1 и CCR получают новые значения только при CNT = 0, иначе запись осуществляется немедленно. Факт окончания записи обозначается взведением флага WR_CMPL.

7.17.8 Примеры

7.17.8.1 Обычный счетчик

RST_CLK->PER_CLOCK = 0xFFFFFFFF;

RST_CLK->TIM_CLOCK = 0x07000000;

TIMx->TIMx_CNTRL = 0x00000000;

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x0000000F; //Основание счета

TIMx->TIMx_IE = 0x00000002; //Разрешение генерировать прерывание при CNT = ARR

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.

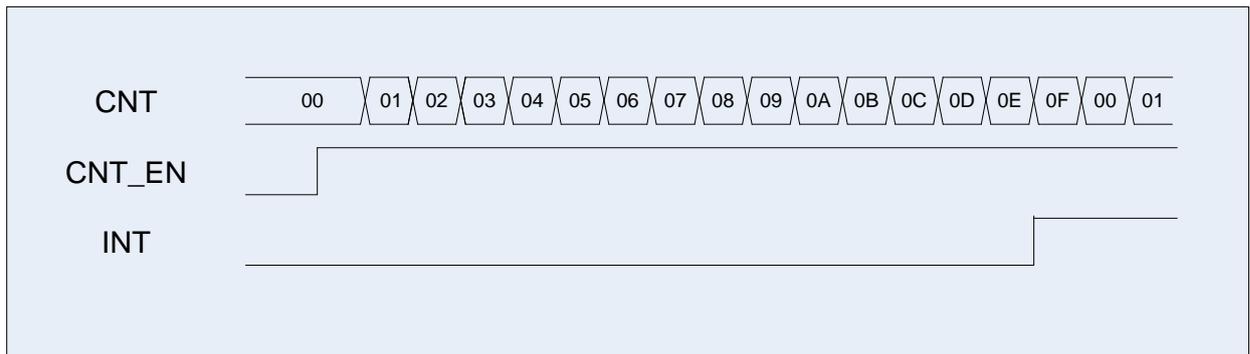


Рисунок 112 – Диаграмма работы счётчика

7.17.8.2 Режим захвата

RST_CLK->PER_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров

RST_CLK->TIM_CLOCK = 0x07000000; //Включение тактовой частоты таймеров

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x000000FF; //Основание счета

TIMx->TIMx_IE = 0x00001E00; //Разрешение генерировать прерывание

//по переднему фронту на выходе CAP по всем каналам

//Режим работы каналов - захват

TIMx->TIMx_CHy_CNTRL[0] = 0x00008000;

TIMx->TIMx_CHy_CNTRL[1] = 0x00008002;

TIMx->TIMx_CHy_CNTRL[2] = 0x00008001;

TIMx->TIMx_CHy_CNTRL[3] = 0x00008003;

//Режим работы выхода канала – канал на выход не работает

TIMx->TIMx_CHy_CNTRL 1[0]= 0x00000000;

TIMx->TIMx_CHy_CNTRL 1[1]= 0x00000000;

TIMx->TIMx_CHy_CNTRL 1[2]= 0x00000000;

TIMx->TIMx_CHy_CNTRL 1[3]= 0x00000000;

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.

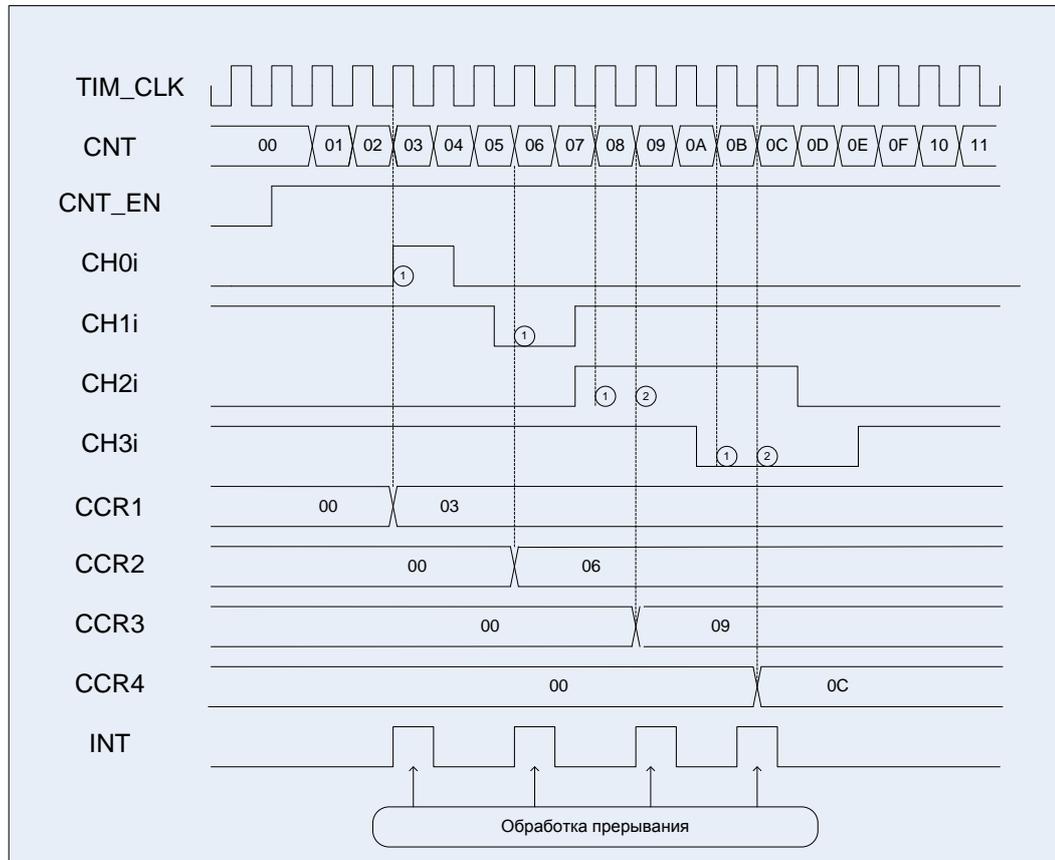


Рисунок 113 – Диаграммы примера работы в режиме захвата

7.17.8.3 Режим ШИМ

RST_CLK->PER_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров

RST_CLK->TIM_CLOCK = 0x07000000; //Включение тактовой частоты таймеров

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000010; //Основание счета

TIMx->TIMx_IE = 0x000001E0; //Разрешение генерировать прерывание

//по переднему фронту на выходе REF по всем каналам

//Режим работы каналов - ШИМ

TIMx->TIMx_CHy_CNTRL[0] = 0x00000200;

TIMx->TIMx_CHy_CNTRL[1] = 0x00000200;

TIMx->TIMx_CHy_CNTRL[2] = 0x00000400;

TIMx->TIMx_CHy_CNTRL[3] = 0x00000600;

//Режим работы выхода канала – канал на выход не работает

TIMx->TIMx_CHy_CNTRL1[0]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[1]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[2]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[3]= 0x00000099;

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK.

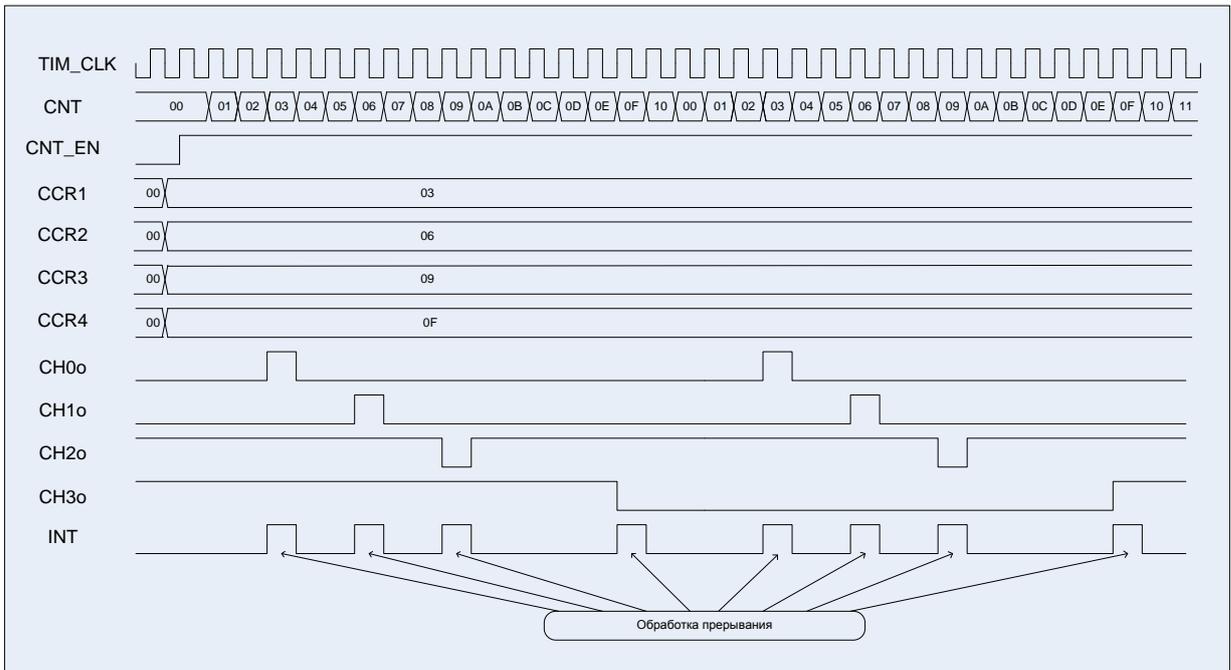


Рисунок 114 – Диаграмма работы в режиме ШИМ

7.17.9 Описание регистров

Таблица 107 – Описание регистров контроллера таймеров общего назначения

Базовый адрес	Название	Состояние после сброса	Описание
0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000			
Смещение			
0x0000_0000	CNT		Основной счетчик таймера
0x0000_0004	PSG		Делитель частоты при счете основного счетчика
0x0000_0008	ARR		Основание счета основного счетчика
0x0000_000C	CNTRL		Регистр управления основного счетчика
0x0000_0010	CCR1		Регистр сравнения, захвата для 1 канала таймера
0x0000_0014	CCR2		Регистр сравнения, захвата для 2 канала таймера
0x0000_0018	CCR3		Регистр сравнения, захвата для 3 канала таймера
0x0000_001C	CCR4		Регистр сравнения, захвата для 4 канала таймера
0x0000_0020	CH1_CNTRL		Регистр управления для 1 канала таймера
0x0000_0024	CH2_CNTRL		Регистр управления для 2 канала таймера
0x0000_0028	CH3_CNTRL		Регистр управления для 3 канала таймера
0x0000_002C	CH4_CNTRL		Регистр управления для 4 канала таймера
0x0000_0030	CH1_CNTRL1		Регистр управления 1 для 1 канала таймера
0x0000_0034	CH2_CNTRL1		Регистр управления 1 для 2 канала таймера
0x0000_0038	CH3_CNTRL1		Регистр управления 1 для 3 канала таймера
0x0000_003C	CH4_CNTRL1		Регистр управления 1 для 4 канала таймера
0x0000_0040	CH1_DTG		Регистр управления DTG для 1 канала таймера
0x0000_0044	CH2_DTG		Регистр управления DTG для 2 канала таймера
0x0000_0048	CH3_DTG		Регистр управления DTG для 3 канала таймера
0x0000_004C	CH4_DTG		Регистр управления DTG для 4 канала таймера
0x0000_0050	BRKETR_CNTRL		Регистр управления входом BRK и ETR
0x0000_0054	STATUS		Регистр статуса таймера
0x0000_0058	IE		Регистр разрешения прерывания таймера
0x0000_005C	DMA_RE		Регистр разрешения запросов DMA от прерываний таймера
0x0000_0060	CH1_CNTRL2		Регистр управления 2 для 1 канала таймера
0x0000_0064	CH2_CNTRL2		Регистр управления 2 для 2 канала таймера
0x0000_0068	CH3_CNTRL2		Регистр управления 2 для 3 канала таймера
0x0000_006C	CH4_CNTRL2		Регистр управления 2 для 4 канала таймера
0x0000_0070	CCR11		Регистр сравнения, захвата 1 для 1 канала таймера
0x0000_0074	CCR21		Регистр сравнения, захвата 1 для 2 канала таймера
0x0000_0078	CCR31		Регистр сравнения, захвата 1 для 3 канала таймера
0x0000_007C	CCR41		Регистр сравнения, захвата 1 для 4 канала таймера
0x0000_0080	DMA_RE1		Регистр разрешения запросов DMA от прерываний канала 1 таймера
0x0000_0084	DMA_RE2		Регистр разрешения запросов DMA от прерываний канала 2 таймера
0x0000_0088	DMA_RE3		Регистр разрешения запросов DMA от прерываний канала 3 таймера
0x0000_008C	DMA_RE4		Регистр разрешения запросов DMA от прерываний канала 4 таймера

7.17.9.1 CNT

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0000												
------------	--	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

CNT[15:0]															

Бит	Имя	Значение	Описание
31...0	CNT[31:0]	0x00000000	Значение основного счетчика таймера

7.17.9.2 PSG

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0004												
------------	--	---------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

PSG[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

PSG[15:0]															

Бит	Имя	Значение	Описание
31...0	PSG[31:0]	0x00000000	Значение предварительного делителя счетчика Основной счетчик считает на частоте CLK = TIM_CLK/(PSG+1)

7.17.9.3 TIMx_ARR

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0008												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															

Бит	Имя	Значение	Описание
31...0	ARR[31:0]	0x00000000	Основание счета для основного счетчика CNT = [0...ARR]

7.17.9.4 CNTRL

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_000C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				EVENT_SEL[3:0]				CNT_MODE[1:0]		FDTs[1:0]		DIR	WRCMPL	ARRBEN	CNT

Бит	Имя	Значение	Описание
31..11	-		Зарезервировано
11..8	EVENT_SEL[3:0]	0x0	Биты выбора источника событий 0000 – всегда “0” 0001 – событие на TMR_EVENT0 0010 – событие на TMR_EVENT1 0011 – событие на TMR_EVENT2 (описание TMR_EVENTx в рисунке каскадного объединения) 0100 – событие на первом канале 0101 – событие на втором канале 0110 – событие на третьем канале 0111 – событие на четвертом канале 1000 – событие переднего фронта ETR 1001 - событие заднего фронта ETR 1010 - событие на TMR_EVENT3

Бит	Имя	Значение	Описание
7..6	CNT_MODE[1:0]	00	Режим счета основного счетчика 00 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT_SEL = 0000 01 – счетчик вверх/вниз с автоматическим изменением DIR при EVENT_SEL = 0000 10 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT_SEL != 0000 11 – счетчик вверх/вниз с автоматическим изменением DIR при EVENT_SEL != 0000
5..4	FDTTS[1:0]	00	Частота семплирования данных FDTTS 00 – каждый TIM_CLK 01 – каждый второй TIM_CLK 10 – каждый третий TIM_CLK 11 – каждый четвертый TIM_CLK
3	DIR	0	Направление счета основного счетчика 0 – вверх, от 0 до ARR 1 – вниз, от ARR до 0
2	WR_CMPL	0	Окончание записи, при задании нового значения регистров CNT, PSG и ARR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
1	ARRB_EN	0	Разрешение мгновенного обновления ARR 0 – ARR будет перезаписан в момент записи в ARR 1 – ARR будет перезаписан при завершении счета CNT
0	CNT_EN	0	Разрешение работы таймера 0 – таймер отключен 1 – таймер включен

7.17.9.5 CCRx

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0010 0x0000_0014 0x0000_0018 0x0000_001C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR[15:0]															

Бит	Имя	Значение	Описание
31...0	CCR[31:0]	0x00000000	Значение CCR, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

7.17.9.6 CCRx1

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0070 0x0000_0074 0x0000_0078 0x0000_007C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															

Бит	Имя	Значение	Описание
31...0	CCR1[31:0]	0x00000000	Значение CCR1, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

7.17.9.7 CHx_CNTRL

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0020 0x0000_0024 0x0000_0028 0x0000_002C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPnPWM	WRMPL	ETREN	BRKEN	OCCM[2:0]			OCCE	CHPSC[1:0]		CHSEL[1:0]		CHFLTR[3:0]			

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15	CAPnPWM	0	Режим работы канала Захват или ШИМ 1 – канал работает в режиме Захват 0 – канал работает в режиме ШИМ
14	WRMPL	0	Флаг окончания записи, при задании нового значения регистра CCR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
13	ETREN	0	Разрешения сброса по выводу ETR 0 – запрещен сброс 1 – разрешен
12	BRKEN	0	Разрешение сброса по выводу BRK 0 – запрещен сброс 1 – разрешен
11...9	OCCM[2:0]	000	<p>Формат выработки сигнала REF в режиме ШИМ</p> <p>Если CCR1_EN = 0:</p> <p>000 – всегда 0 001 – 1, если CNT = CCR; 010 – 0, если CNT = CCR; 011 – переключение REF, если CNT = CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR = 0 (счет вверх), CNT < CCR, иначе 0; 0, если DIR = 1 (счет вниз), CNT > CCR, иначе 1; 111 – 0, если DIR = 0 (счет вверх), CNT < CCR, иначе 1; 1, если DIR = 1 (счет вниз), CNT > CCR, иначе 0.</p> <p>Если CCR1_EN = 1:</p> <p>000 – всегда 0; 001 – 1, если CNT = CCR или CNT = CCR1 010 – 0, если CNT = CCR или CNT = CCR1; 011 – переключение REF, если CNT = CCR или CNT = CCR1; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR = 0 (счет вверх), CCR < CNT < CCR1, иначе 0; 0, если DIR = 1 (счет вниз), CCR < CNT < CCR1, иначе 1; 111 – 0, если DIR = 0 (счет вверх), CCR < CNT < CCR1, иначе 1; 1, если DIR = 1 (счет вниз), CCR < CNT < CCR1, иначе 0.</p> <p>При условии что CCR < CCR1.</p>

8	OCCE	0	Разрешение работы ETR 0 – запрет ETR 1 – разрешение ETR
7...6	CHPSC[1:0]	00	Предварительный делитель входного канала 00 – нет деления 01 – /2 10 – /4 11 – /8
5...4	CHSEL[1:0]	00	Выбор события по входному каналу CHx _i для фиксации значения основного счетчика (регистр MDR_TIMERx->CNT) в регистр CCR: 00 – положительный фронт на входном канале CHx _i 01 – отрицательный фронт на входном канале CHx _i 10 – положительный фронт от других каналов Для первого канала от 2 канала Для второго канала от 3 канала Для третьего канала от 4 канала Для четвертого канала от 1 канала 11 – положительный фронт от других каналов Для первого канала от 3 канала Для второго канала от 4 канала Для третьего канала от 1 канала Для четвертого канала от 2 канала
3...0	CHFLTR[3:0]	0000	Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTS/2 0101 – в 8 триггерах на частоте FDTS/2 0110 – в 6 триггерах на частоте FDTS/4 0111 – в 8 триггерах на частоте FDTS/4 1000 – в 6 триггерах на частоте FDTS/8 1001 – в 8 триггерах на частоте FDTS/8 1010 – в 5 триггерах на частоте FDTS/16 1011 – в 6 триггерах на частоте FDTS/16 1100 – в 8 триггерах на частоте FDTS/16 1101 – в 5 триггерах на частоте FDTS/32 1110 – в 6 триггерах на частоте FDTS/32 1111 – в 8 триггерах на частоте FDTS/32

7.17.9.8 CHx_CNTRL1

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0030 0x0000_0034 0x0000_0038 0x0000_003C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			NINV	NSELO[1:0]		NSELOE[1:0]		-			INV	SELO [1:0]		SELOE [1:0]	

Бит	Имя	Значение	Описание
31..13	-		Зарезервировано
12	NINV	0	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
11..10	NSELO[1:0]	00	Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 – на выход выдается сигнал с DTG.
9...8	NSELOE[1:0]	00	Режим работы канала на выход 00 – всегда на ОЕ выдается 0, канал на выход не работает 01 – всегда на ОЕ выдается 1, канал всегда работает на выход 10 – на ОЕ выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на ОЕ выдается сигнал с DTG, при CHn = 0 вход, при CHn = 1 выход
7...5	-		Зарезервировано
4	INV	0	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
3...2	SELO[1:0]	00	Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 – на выход выдается сигнал с DTG.
1...0	SELOE[1:0]	00	Режим работы канала на выход 00 – всегда на ОЕ выдается 0, канал на выход не работает 01 – всегда на ОЕ выдается 1, канал всегда работает на выход 10 – на ОЕ выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на ОЕ выдается сигнал с DTG, при CH = 0 вход, при CH = 1 выход

7.17.9.10 CHx_DTG

Base ADDR=		0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000				Offset=		0x0000_0040 0x0000_0044 0x0000_0048 0x0000_004C									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTG[7:0]											EDTS	DTGx[3:0]			

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15...8	DTGx[7:0]		Основной делитель частоты Задержка DTGdel = DTGx*(DTG+1).
7...5	-		Зарезервировано
4	EDTS		Частота работы DTG 0 – TIM_CLK 1 – FDS
3...0	DTG[3:0]		Предварительный делитель частоты DTG

7.17.9.11 BRKETR_CNTRL

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_0050												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ETR_FILTER[3:0]				ETR_PSC[1:0]		ETRINV	BRKINV

Бит	Имя	Значение	Описание
31..8	-		Зарезервировано
7...4	ETR FILTER[3:0]		Цифровой фильтр на входе ETR. Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTs/2 0101 – в 8 триггерах на частоте FDTs/2 0110 – в 6 триггерах на частоте FDTs/4 0111 – в 8 триггерах на частоте FDTs/4 1000 – в 6 триггерах на частоте FDTs/8 1001 – в 8 триггерах на частоте FDTs/8 1010 – в 5 триггерах на частоте FDTs/16 1011 – в 6 триггерах на частоте FDTs/16 1100 – в 8 триггерах на частоте FDTs/16 1101 – в 5 триггерах на частоте FDTs/32 1110 – в 6 триггерах на частоте FDTs/32 1111 – в 8 триггерах на частоте FDTs/32
3...2	ETRPSC[1:0]		Асинхронный предделитель внешней частоты 00 – без деления 01 – /2 10 – /4 11 – /8
1	ETR INV		Инверсия входа ETR 0 – без инверсии 1 – инверсия
0	BRK INV		Инверсия входа BRK 0 – без инверсии 1 – инверсия

7.17.9.12 STATUS

Base ADDR=		0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000				Offset=		0x0000_0054							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															CCRCAP1 EVENT[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1EVENT[2:0]			CCRREFEVENT[3:0]				CCRCAP1EVENT[3:0]			BRKEVENT	ETRFEEVE NT	ETRREEVE NT	CNTARR EVENT	CNTZERO EVENT	

Бит	Имя	Значение	Описание
31..17	-		Зарезервировано
16..13	CCRCAP1EVENT[3:0]		Событие настроенного фронта на входе CHx _i канала таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREFEVENT[3:0]		Событие переднего фронта на выходе REF каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAP1EVENT[3:0]		Событие настроенного фронта на входе CHx _i канала таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
4	BRKEVENT		Триггерированное по PCLK состояние входа BRK, 0 – BRK == 0 1 – BRK == 1 Сбрасывается записью 0, при условии наличия 0 на входе BRK
3	ETRFEEVENT		Событие заднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.

2	ETREEVENT		<p>Событие переднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.</p>
1	CNTARREVENT		<p>Событие совпадения CNT с ARR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT и ARR не изменили состояния, то флаг повторно не взводится.</p>
0	CNTZEROEVENT		<p>Событие совпадения CNT с нулем 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT не изменил состояния, то флаг повторно не взводится.</p>

7.17.9.13 IE

Base ADDR=		0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000				Offset=		0x0000_0058							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															CCRCAP1 EVENTIE [3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1EVENTIE[2:0]			CCRREFEVENTIE[3:0]				CCRCAP1EVENTIE[3:0]			BRKEVENTIE	ETRFEEVENTIE	ETRREEVENTIE	CNTARR EVENTIE	CNTZERO EVENTIE	

Бит	Имя	Значение	Описание
31..17	-		Зарезервировано
16..13	CCRCAP1 EVENTIE [3:0]		Флаг разрешения прерывания по событию настроенного фронта на входе CHx канала таймера (фиксация значения основного счетчика таймера в регистре CCR1) 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREF EVENTIE[3:0]		Флаг разрешения прерывания по событию переднего фронта на выходе REF каналов таймера 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAP EVENTIE [3:0]		Флаг разрешения прерывания по событию настроенного фронта на входе CHx канала таймера (фиксация значения основного счетчика таймера в регистре CCR) 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENTIE		Флаг разрешения по триггерированному по PCLK состоянию входа BRK, 0 – нет прерывания 1 – прерывание разрешено
3	ETRFEEVENTIE		Флаг разрешения прерывания по заднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
2	ETRRE EVENTIE		Флаг разрешения прерывания по переднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
1	CNTARR EVENTIE		Флаг разрешения прерывания по событию совпадения CNT и ARR 0 – нет прерывания 1 – прерывание разрешено

Бит	Имя	Значение	Описание
0	CNTZERO EVENTIE		Флаг разрешения прерывания по событию совпадения CNT и нуля 0 – нет прерывания 1 – прерывание разрешено

7.17.9.14 DMA_REx

Base ADDR=	0x4008_A000 0x4008_B000 0x4008_C000 0x4008_D000	Offset=	0x0000_005C 0x0000_0080 0x0000_0084 0x0000_0088 0x0000_008C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															CCRCAP1 EVENTRE [3]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1 EVENTRE[2:0]			CCRREF EVENTRE[3:0]			CCRCAP EVENTRE[3:0]					BRK EVENTRE	ETRFE EVENTRE	ETRRE EVENTRE	CNTARR EVENTRE	CNTZERO EVENTRE

Бит	Имя	Значение	Описание
31..17	-		Зарезервировано
16..13	CCRCAP1 EVENTRE [3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP1 каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREF EVENTRE[3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе REF каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAP EVENTRE [3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
4	BRKEVENTRE		Флаг разрешения по пересинхронизированному по PCLK состоянию входа BRK, 0 – нет запроса DMA 1 – запрос DMA разрешен

3	ETRFEEVENTRE		Флаг разрешения запроса DMA по заднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен
2	ETRREEVENTRE		Флаг разрешения запроса DMA по переднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен
1	CNTARREVENTRE		Флаг разрешения запроса DMA по событию совпадения CNT и ARR 0 – нет запроса DMA 1 – запрос DMA разрешен
0	CNTZEROEVENTRE		Флаг разрешения запроса DMA по событию совпадения CNT и нуля 0 – нет запроса DMA 1 – запрос DMA разрешен

7.18 Контроллер CAN

В микроконтроллере реализован контроллер интерфейса CAN, который является полнофункциональным CAN-узлом, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости не более 1 Мбит/сек.

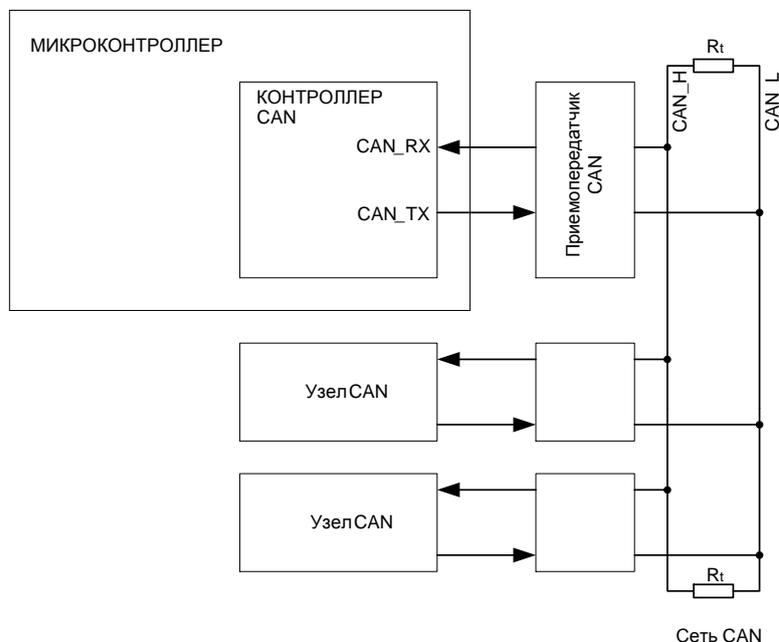


Рисунок 115 – Структурная блок-схема организации сети CAN

Интерфейс CAN позволяет обмениваться сообщениями в сети равноправных устройств. При передаче сообщения в сети CAN все узлы сети получают это сообщение. В сообщении передается уникальный идентификатор узла и данные. Все сообщения в протоколе CAN довольно короткие и могут содержать не более восьми байтов данных. При возникновении коллизий (одновременная передача сообщений различными узлами) при передаче идентификатора, происходит арбитраж, и узел с большим номером идентификатора уступает сеть узлу с меньшим номером идентификатора.

Особенности:

- поддержка CAN протокола версии CAN2.0 A и B;
- скорость передачи до 1 Мбит/с;
- 32 буфера приема/передачи;
- поддержка приоритетов сообщений;
- 32 фильтра приема;
- маскирование прерываний.

7.18.1 Режимы работы

CAN-контроллер поддерживает несколько режимов работы: нормальный режим для приема и передачи пакетов сообщений, режим работы только на прием, режим самотестирования и режим инициализации для задания параметров связи.

Режим нормальной передачи (регистр CAN_STATUS: ROM = 0, STM = 0)

Выводы CAN_TX и CAN_RX подключены к шине.

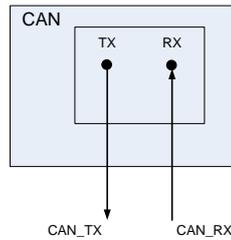


Рисунок 116 – Режим нормальной передачи

В этом режиме можно установить флаги разрешения приема своих пакетов и флаги разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP).

Режим работы только на прием – ReceiveOnlyMode (регистр CAN_STATUS: ROM = 1, STM = 0)

Контроллер CAN интерфейса принимает, но не посылает никакой информации, т.е. линия TX всегда в «1», но внутри контроллера все управляющие сигналы проходят.

Режим самотестирования – Self Test Mode (регистр CAN_STATUS: STM = 1, ROM = 0)

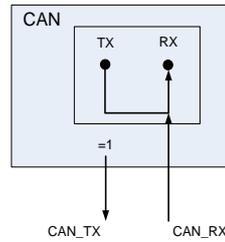


Рисунок 117 – Режим работы только на прием – ReceiveOnlyMode

Выводы CAN_TX и CAN_RX отключены, вся передаваемая информация видна только внутри контроллера.

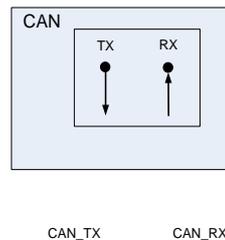


Рисунок 118 – Режим самотестирования – SelfTestMode

Для успешного приема своих сообщений необходимо установить флаги разрешения приема своих пакетов и разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP). В этом режиме передаваемые сообщения сразу же принимаются в приемный буфер. Режим самотестирования полезен в период отладки кода программы.

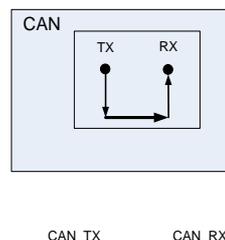


Рисунок 119 – Режим инициализации для задания параметров связи

Еще одна важная функция CAN-контроллера – фильтрация получаемых сообщений. Поскольку CAN является широкополосной шиной, каждое переданное сообщение принимается всеми узлами шины. В CAN-шине любой разумной степени сложности передается достаточно большое число сообщений. Задачей каждого подключенного к CAN-узлу ЦПУ является реагирование на CAN-сообщения. Таким образом, чтобы избавить CAN-контроллер от проблемы приема в буфер нежелательных сообщений, необходима их фильтрация. У CAN-контроллера микроконтроллеров 1986VE8x имеется 32 регистра фильтров и 32 регистра масок, которые можно использовать для блокировки всех CAN-сообщений, кроме избранных сообщений или групп сообщений.

7.18.2 Типы пакетов сообщений

Информация на шине представлена в виде фиксированных сообщений различной, но ограниченной длины. Когда шина свободна, любой подключенный узел может начать передавать новое сообщение. При передаче информации с помощью протокола CAN используется четыре типа пакетов:

- **пакет удаленного запроса данных** передается узлом, чтобы запросить передачу пакета данных с тем же самым идентификатором;
- **пакет ошибки** передается любым узлом при обнаружении ошибочного состояния на шине. Пакет ошибки передается сразу же после обнаружения ошибки и накладывается на передаваемый пакет так, чтобы испортить его окончательно. Таким образом, если один из узлов обнаружил ошибку, он усиливает ошибку для того, чтобы ее обнаружили и другие узлы;
- **пакет перегрузки** используется для обеспечения дополнительной задержки между предшествующим и последующим кадрами данных или кадрами удаленного запроса данных. Он передается в редких случаях, подробнее можно прочесть в стандарте ISO 11898-1. Контроллер CAN интерфейса отправляет пакет перегрузки в соответствии со стандартом;
- основными пакетами на шине CAN являются **пакеты данных**. Пакет данных передает данные от передатчика приемнику. Пакеты могут быть стандартными и расширенными. Отличие пакетов заключается в размере полей идентификатора. Пакеты с 11-разрядным идентификатором называются стандартными пакетами. Пакеты, содержащие 29-разрядные идентификаторы, называются расширенными пакетами. При передаче идентификационной информации происходит автоматический арбитраж на шине CAN таким образом, чтобы пакет с меньшим значением поля ID остался на шине. На шине не допускается наличие двух или более узлов с одним и тем же идентификатором. Размер передаваемых данных кодируется в поле DLC и может составлять от 0 до 8 байт. После передачи поля данных контроллер автоматически передает рассчитанное значение CRC. Если хотя бы один из узлов принял пакет, он выставляет АСК подтверждение на шине, если хотя бы один из узлов обнаружит ошибку, на шину будет выставлен пакет ошибки. Таким образом, обеспечивается гарантированность доставки сообщений.

Пакеты данных и пакеты удаленного запроса данных отделяются от предшествующих пакетов межкадровым пространством.

7.18.3 Структура пакета данных (DataFrame)

Пакет данных состоит из 7 различных полей:

- "начало пакета" (SOF-startofframe);
- "поле арбитража" (arbitrationfield);
- "поле контроля" (controlfield);
- "поле данных" (datafield);
- "поле CRC" (CRCfield);
- "поле подтверждения" (ACKfield);
- "конец пакета" (endofframe).

7.18.3.3 Идентификатор

Идентификатор – стандартный формат. Длина идентификатора – 11 бит и соответствует StandartID в расширенном формате. Эти биты передаются в порядке от Bit28 до Bit18. Самый младший бит – Bit18. 7 старших бит (Bit28 – Bit 22) не должны быть все единичными битами.

Идентификатор – расширенный формат. В отличие от стандартного идентификатора, расширенный идентификатор состоит из 29 бит. Его формат содержит две секции:

- **StandartID – 11 бит;**
- **ExtendedID – 18 бит.**

Standart ID состоит из 11 бит. Эта секция передается в порядке от Bit28 до Bit18. Это эквивалентно формату стандартного идентификатора. **Standart ID** определяет базовый приоритет расширенного пакета.

Extended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В стандартном пакете идентификатор сопровождается RTR битом.

7.18.3.4 Бит RTR

Бит запроса удаленной передачи. В пакетах данных RTR бит должен быть передан нулевым уровнем. Внутри пакета удаленного запроса данных RTR бит должен быть единичным. В расширенном пакете сначала передается **StandartID**, с последующими битами IDE и SRR. Extended ID передается после SRR бита.

7.18.3.5 Бит SRR (расширенный формат)

Заменитель бита удаленного запроса. SRR – единичный бит. Он передается в расширенных пакетах в позиции RTR бита. Таким образом, он заменяет RTR – бит стандартного пакета.

Следовательно, при одновременной передаче стандартного пакета и расширенного пакета, **StandartID** которого совпадает с идентификатором стандартного пакета, стандартный пакет преобладает над расширенным пакетом.

7.18.3.6 Бит IDE (расширенный формат)

Бит IDE – бит расширения идентификатора

Бит IDE принадлежит:

- полю арбитража для расширенного формата;
- полю управления для стандартного формата.

Бит IDE в стандартном формате передается нулевым уровнем, в расширенном формате Бит IDE – единичный уровень.

7.18.3.7 Поле управления (Controlfield)

Поле управления состоит из шести бит. Формат поля управления отличается для стандартного и расширенного формата.

Пакеты в стандартном формате включают: код длины данных (DLC), бит IDE, который передается нулевым уровнем (см. выше), и зарезервированный бит r0.

Пакеты в расширенном формате включают код длины данных и два зарезервированных бита r1 и r0. Зарезервированные биты должны быть посланы нулевым уровнем, но приемники принимают единичные и нулевые уровни биты во всех комбинациях.

7.18.3.8 Код длины данных (Datalengthcode)

Число байт в поле данных обозначается кодом длины данных. Этот код длины данных, размером 4 бита, передается внутри поля управления. Допустимое число байт данных: {0,1, ..., 7,8}. Другие величины использоваться не могут.

7.18.3.9 Поле данных (Datafield)

Поле данных состоит из данных, которые будут переданы внутри пакета данных. Оно может содержать от 0 до 8 байт, каждый содержит 8 бит, которые передаются, начиная со старшего значащего бита.

7.18.3.10 Поле CRC (CRCfield)

Содержит последовательность CRC и CRC – разделитель. При вычислении 15 битного CRC кода используется последовательность бит, состоящая из полей: "начало пакета", "поле арбитража", "управляющее поле", "поле данных" (если есть). Последовательность CRC сопровождается разделителем CRC, который состоит из одного единичного бита.

7.18.3.11 Поле подтверждения (ACKfield)

Поле подтверждения имеет длину два бита и содержит: "область подтверждения" и разделитель подтверждения. В поле подтверждения передающий узел посылает два бита с единичным уровнем. Приемник, который получил сообщение правильно (CRC соответствует), сообщает об этом передатчику, посылая бит с нулевым уровнем в течение приема поля "область подтверждения".

7.18.3.12 Конец пакета (End of frame)

Каждый пакет данных и пакет удаленного запроса данных ограничен последовательностью флагов, состоящей из семи единичных бит.

7.18.4 Структура пакета удаленного запроса данных (Remoteframe)

Узел, действующий как приемник некоторых данных, может инициировать передачу соответствующих данных исходными узлами, посылая пакет удаленного запроса данных. Пакет удаленного запроса данных существует и в стандартном формате, и в расширенном формате. В обоих случаях он состоит из шести битовых полей:

- "начало пакета" (Start of frame);
- "поле арбитража" (Arbitration field);
- "управляющее поле" (Control field);
- "поле CRC" (CRC-field);
- "поле подтверждения" (ACK field);
- "конец пакета" (End of frame).

В отличие от обычного пакета данных, RTR бит пакета удаленного запроса данных - единичный. В этом пакете отсутствует поле данных. При этом значение кода длины данных может принимать любое значение в пределах допустимого диапазона [0,8]. Значение кода длины данных соответствует коду длины данных кадра данных. RTR бит указывает, является ли переданный кадр кадром данных.

7.18.5 Арбитраж на шине

Арбитраж сообщений гарантирует, что наиболее важное сообщение захватит шину и будет передано без задержки. Затем будут переданы приостановленные сообщения согласно их приоритетам (сообщение с наименьшим идентификатором передается первым).

Если планируется передача сообщения, и шина свободна, то сообщение будет передано и сможет быть принято любым заинтересованным в нем узлом. Если передача сообщения запланирована, а шина активна, то прежде чем приступить к передаче сообщения, необходимо дождаться освобождения шины. Если запланирована передача нескольких сообщений, то при освобождении шины они начнут передаваться одновременно, синхронизируясь по признаку начала пакета. В этом случае на шине начнется процесс арбитража, задача которого – определить, какое именно из сообщений захватит шину и будет передано.

Арбитраж сообщений на шине CAN осуществляется методом, который называется «неразрушающий побитовый арбитраж».

На рисунке изображены три сообщения, ожидающие передачи. После освобождения шины и синхронизации пакетов сообщений по старт-биту на шину начинают выдаваться все три идентификатора. При передаче первых двух бит все три узла выставляют на шину одинаковые логические уровни и соответственно считывают те же значения, поэтому они все продолжают передачу. Однако при передаче третьего бита узлы А и С выставляют на шину доминантный бит, а узел В выставляет рецессивный бит, но при этом считывает с шины доминантный. В результате узел освобождает шину и начинает следить за ее состоянием. Узлы А и С продолжают передачу, пока ситуация не повторится; теперь узел С выдает рецессивный бит, а узел А – доминантный. При этом узел С прекращает передачу и начинает следить за состоянием шины. С этого момента шина захватывается узлом А. После передачи сообщения узлом А узлы В и С начинают передачу, причем узел С захватит шину и передает свое сообщение. Если бы узлу А снова надо было передавать сообщение, он снова захватил бы шину. Таким образом, первым на шине CAN передается сообщение с наименьшим идентификатором.

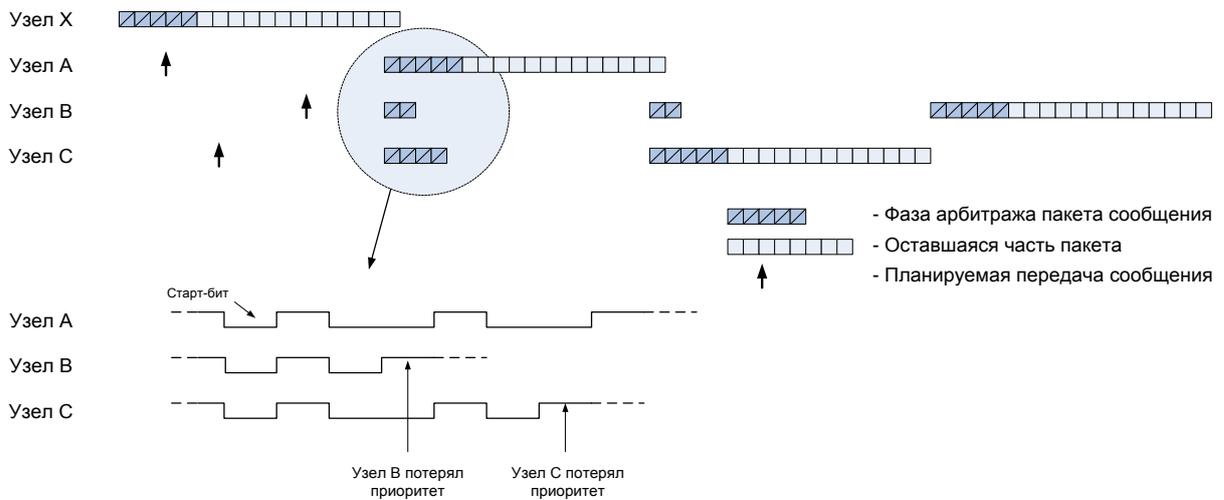


Рисунок 123 – Арбитраж на шине CAN

В случае «проигрыша» арбитража в регистре статуса контроллера CAN будет установлен флаг ID_LOWER.

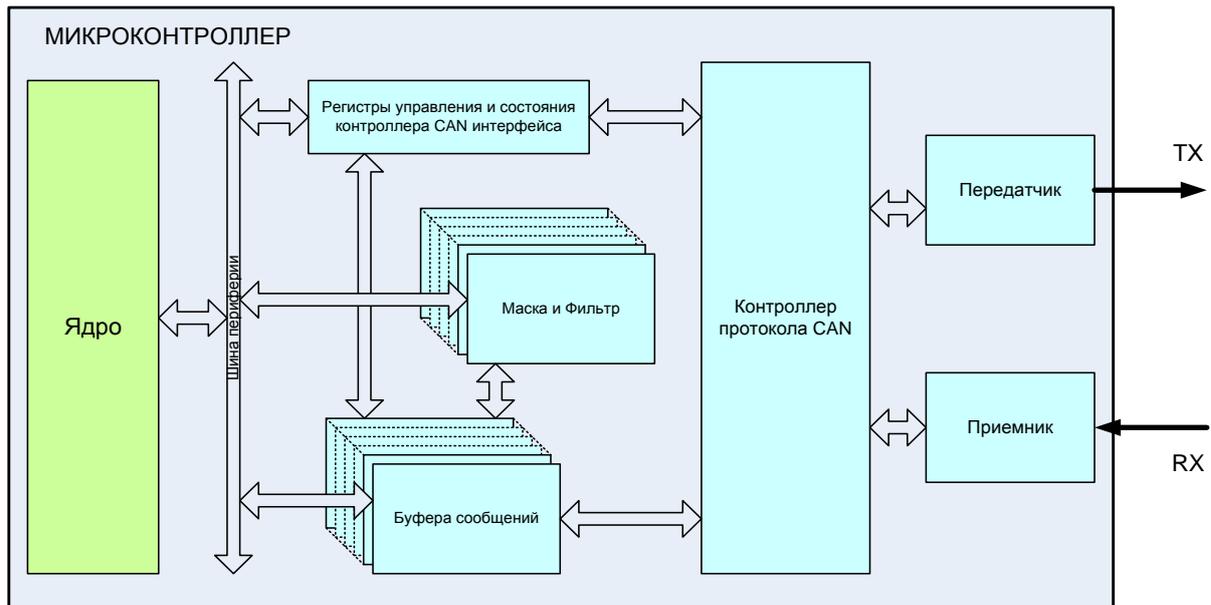


Рисунок 124 – Структурная блок-схема контроллера CAN

7.18.6 Инициализация

Перед началом работы с контроллерами CAN в первую очередь должны быть заданы параметры тактового сигнала.

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 0 для CAN1, бит 1 для CAN2 регистра PER_CLOCK). В регистре CAN_CLOCK установить бит CANyCLKEN, чтобы разрешить тактовую частоту для определенного контроллера CAN, задать коэффициент деления тактовой частоты HCLK для каждого CAN контроллера.

После подачи тактового сигнала на блок таймера можно приступить к работе с ним.

Для работы контроллера шины CAN он должен быть настроен на соответствующую скорость шины CAN. Для этого должны быть заданы соответствующим образом поля SB, SJW, SEG2, SEG1, PSEG и BRP в регистре CAN_BITTMNG. После этого должны быть заданы работающие буфера сообщений путем задания бит EN (разрешение работы) RXTXn (1 – прием, 0 – передача) в регистре BUF_xx_CON. После этого должен быть выдан общий сигнал разрешения работы контроллера через задание бита CANEN в регистре CONTROL. После этого контроллер CAN начинает работу.

7.18.7 Передача сообщений

Для передачи сообщения необходимо, в разрешенный для работы и конфигурируемый на передачу буфер, записать сообщение для передачи (задать значения регистрам CAN_BUF[x].ID, CAN_BUF[x].DLC, CAN_BUF[x].DATAL и CAN_BUF[x].DATAH), после чего установить бит TX_REQ. После установки этого бита сообщение будет поставлено в очередь на отправку. После отправки сообщения бит TX_REQ будет автоматически сброшен. Если в нескольких буферах есть сообщения на отправку, то порядок отправки определяется по полю PRIOR_0. Если у сообщения бит PRIOR_0 выставлен в ноль, то оно отправляется в первую очередь. Если есть несколько сообщений с одинаковым приоритетом, то порядок отправки определяется порядковым номером буфера, буфер с меньшим порядковым номером имеет больший приоритет. Значение полей ID для выбора порядка отправки в рамках контроллера CAN (одного узла) значения не имеет. По ID выбирается приоритет между различными узлами.

7.18.8 Передача сообщений по RemoteTransmitRequest (RTR)

Для автоматической отправки сообщения по запросу Remote Transmit Request необходимо задать режим маскирования для данного буфера таким образом, чтобы он принимал только сообщения от устройства, которое может выслать RTR запрос. При необходимости в регистре INT_TX настроить генерацию прерывания передачи для соответствующего буфера. В регистре управления этим буфером (BUFF_CON[x]) проверить, что флаг TX_REQ = 0, задать приоритет отправляемого сообщения PRIOR_0, установить разрешение ответа при приеме RTR в буфер (RTR_EN = 1), задать RX_TX = 0 для разрешения отправки сообщения и задать EN = 1 для разрешения работы буфера. В регистре идентификации задать необходимые SID и EID, в регистре BUF_xx_DLC указать формат пакета (расширенный или стандартный) и указать длину передаваемых данных в поле DLC. В регистрах данных CAN_BUF[x].DATAL и CAN_BUF[x].DATAH задать необходимые для отправки данные. Далее можно переходить к выполнению остальной части программы с отправкой CAN сообщений. Отправка сообщения буфером будет произведена по RTR запросу, удовлетворяющему механизму фильтрации для принимаемых сообщений, который выбран для данного буфера.

7.18.9 Прием сообщений

Для приема сообщений необходимо иметь свободные и разрешенные для работы буфера, сконфигурированные на прием сообщений. При этом, если по шине CAN будут передаваться сообщения от других узлов, они будут сохраняться в этих буферах.

7.18.10 Автоматическая фильтрация принимаемых сообщений

Для уменьшения затрат процессорного ядра на обработку принимаемых сообщений, контроллер CAN интерфейса может автоматически фильтровать принимаемые сообщения. Для каждого буфера могут быть заданы маска (CAN_BUF_FILTER[x].MASK) и фильтр (CAN_BUF_FILTER[x].FILTER) таким образом, что в этот буфер будут приниматься только те сообщения, для которых выполняется условие:

$$ID \& CAN_BUF_FILTER[x].MASK == CAN_BUF_FILTER[x].FILTER$$

Если принимаемое сообщение не может быть помещено ни в один из буферов, то оно будет проигнорировано. Если сообщение может быть принято более чем одним буфером, то оно будет помещено в буфер с меньшим порядковым номером. При инициализации после включения питания или сброса CAN_BUF_FILTER[x].MASK и CAN_BUF_FILTER[x].FILTER для всех буферов имеют произвольное значение, таким образом, необходимо перед началом работы их проинициализировать. Для приема всех сообщений без фильтрации необходимо задать им нулевое значение. Специального бита для включения или выключения фильтрации нет.

7.18.11 Задание скорости передачи и момента сэмплирования

Все узлы шины CAN должны работать на одной скорости. Протокол CAN использует кодирование без возврата в ноль (NRZ). Также при передаче не передаются тактовые сигналы. Таким образом, приемники должны засинхронизироваться с тактовым сигналом передатчика. Поскольку все узлы имеют свои индивидуальные тактовые генераторы, все приемники имеют специальный блок синхронизации DPLL.

Максимальная скорость передачи CAN 1 Мбит/сек. Время битового интервала Nominal Bit Time определяется как:

$$T_{BIT} = 1 / \text{Скорость передачи}$$

Блок DPLL разбивает битовый интервал на интервалы Time Quanta (TQ). Битовый интервал состоит из четырех частей:

- SynchronizationSegment (Sync_Seg);
- PropagationTimeSegment (PSEG);
- Phase Buffer Segment 1 (SEG1);
- Phase Buffer Segment 2 (SEG2).

По определению Nominal Bit Time программируется длительностью от 8 до 25 TQ. В этом случае:

$$\text{Nominal Bit Time} = TQ * (\text{Sync_Seg} + \text{PSEG} + \text{SEG1} + \text{SEG2})$$

Время TQ фиксировано и определяется периодом генератора и программируемым прескалером BRP со значением от 1 до 65536:

$$TQ (\mu s) = ((BRP+1)) / CANCLK0 (MHz)$$

или

$$TQ (\mu s) = ((BRP+1)) * T_{clk} (\mu s)$$

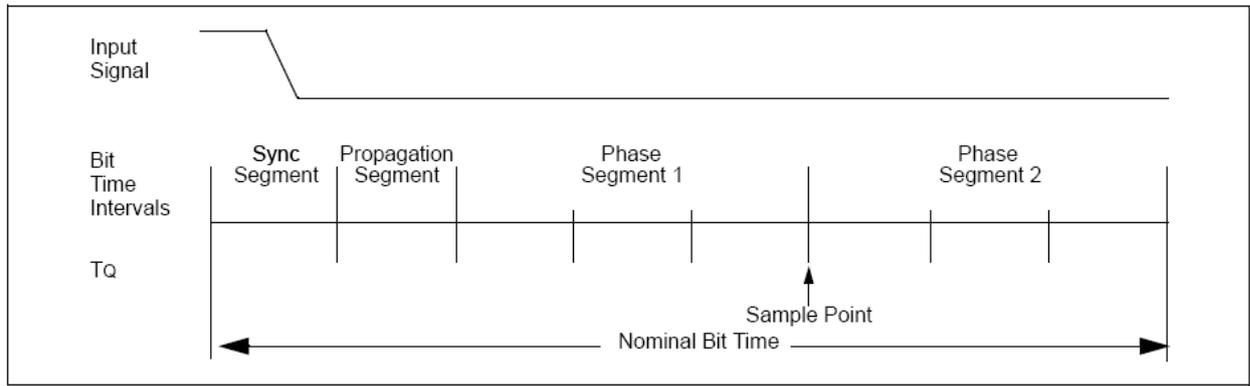


Рисунок 125 – Структура битового интервала

7.18.11.1 SynchronizationSegment

Эта часть битового интервала, в которой должно происходить переключение сигнала. Длительность этого интервала 1 TQ. Если переключение происходит в этой области, то приемник засинхронизирован с передатчиком.

7.18.11.2 PropagationTimeSegment

Эта часть предназначена, чтобы компенсировать физические задержки времени распространения сигнала в шине и внутренние задержки в узлах. Длительность этого интервала может быть запрограммирована от 1 до 8 TQ

7.18.11.3 PhaseBufferSegments

Эти интервалы предназначены для более точной установки точки сэмплирования, которая располагается между ними. Длительности этих интервалов могут быть запрограммированы между 1 и 8 TQ.

7.18.12 Перезапись принятых сообщений

В буфере может быть включено разрешение перезаписи принятого сообщения. Если принимаемое сообщение не может быть сохранено в свободный буфер, то оно может быть сохранено в буфер с ранее полученным сообщением, если для него выставлен бит OVER_EN. При этом выставляется флаг OVER_WR. Таким образом, если у буфера разрешена перезапись принятых сообщений, после прочтения сообщения необходимо проверить флаг OVER_WR. Если он выставлен в 1, то необходимо сбросить OVER_WR (не сбрасывая флаг RX_FULL), затем еще раз прочесть сообщение, после чего снова проверить флаг OVER_WR и, если он не выставлен повторно, то сбросить флаг RX_FULL. Считанное значение считать корректным.

Прибегать к помощи механизма перезаписи принятых сообщений можно только в случае, когда допустима потеря сообщений, работа с перезаписью сообщений не гарантирует прием всех сообщений, а только позволяет принять сообщение корректно, так как момент чтения сообщения может совпасть с моментом сохранения нового сообщения. При этом первая часть считанного процессорным ядром сообщения будет от первого сообщения, вторая от второго. Если же между сбросом флага OVER_WR, чтением сообщения, и при следующей проверке OVER_WR он оказался не выставлен, это означает, что в момент чтения сообщения из буфера в него не сохранялось новое сообщение.

7.18.13 Синхронизация

При обнаружении фронта принимаемого сигнала этот момент принимается как граница между битовыми интервалами; в зависимости от того, на какой интервал приходится фронт, DPLL выполняет различного рода действия по подсинхронизации данных.

7.18.13.1 HardSynchronization

Жесткая синхронизация выполняется однократно во время начала приема сообщения. Независимо от того, в каком состоянии находился DPLL при возникновении фронта, он переводится в Sync_Seg.

7.18.13.2 Resynchronization

Если фронт принимаемого сигнала отклоняется от Sync_Seg, длительность Phase Segment 1 может быть увеличена, а Phase Segment 2 уменьшена, чтобы в следующий раз фронт прошел в нужном месте. Величина изменения Phase Segment 1 и Phase Segment 2 варьируется в зависимости от значения отклонения фронта, но не превышает значения Synchronization Jump Width (SJW).

7.18.14 Обработка ошибок

В спецификации протокола CAN определено пять методов ограничения распространения ошибок, реализованных на аппаратном уровне. При обнаружении любой ошибки передающее устройство повторяет посылку пакета, поэтому ядру не нужно вмешиваться до тех пор, пока не возникнет грубая ошибка. Предусмотрено три метода обнаружения ошибок на уровне пакетов (контроль формата, CRC и подтверждение) и два метода на уровне бит (контроль бит и битстаффинг). Для реализации этих методов используется несколько полей, добавляемых к основному сообщению. При приеме осуществляется проверка, все ли поля присутствуют в сообщении. Если нет, то сообщение игнорируется, генерируется кадр ошибки и в регистре статуса контроллера STATUS устанавливается флаг ошибки формата пакета FRAME_ERR.

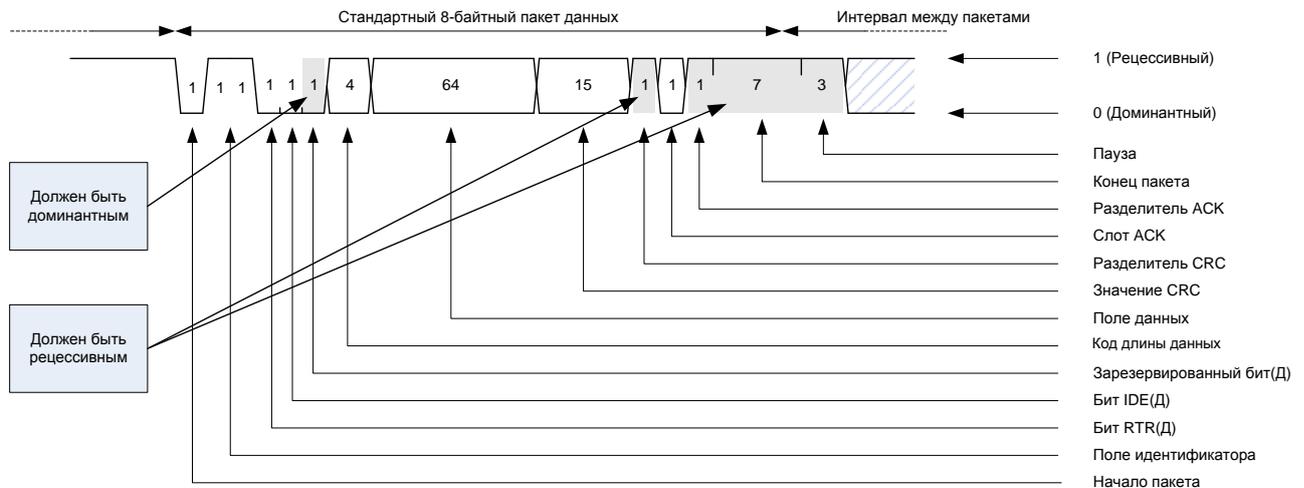


Рисунок 126 – Контроль формата пакета

Каждое сообщение должно подтверждаться вставкой доминантного бита в поле подтверждения. Если подтверждения нет, передающий узел будет передавать сообщение до тех пор, пока не получит подтверждение, при этом в регистре статуса контроллера STATUS будет установлен флаг ошибки подтверждения ACK_ERR.

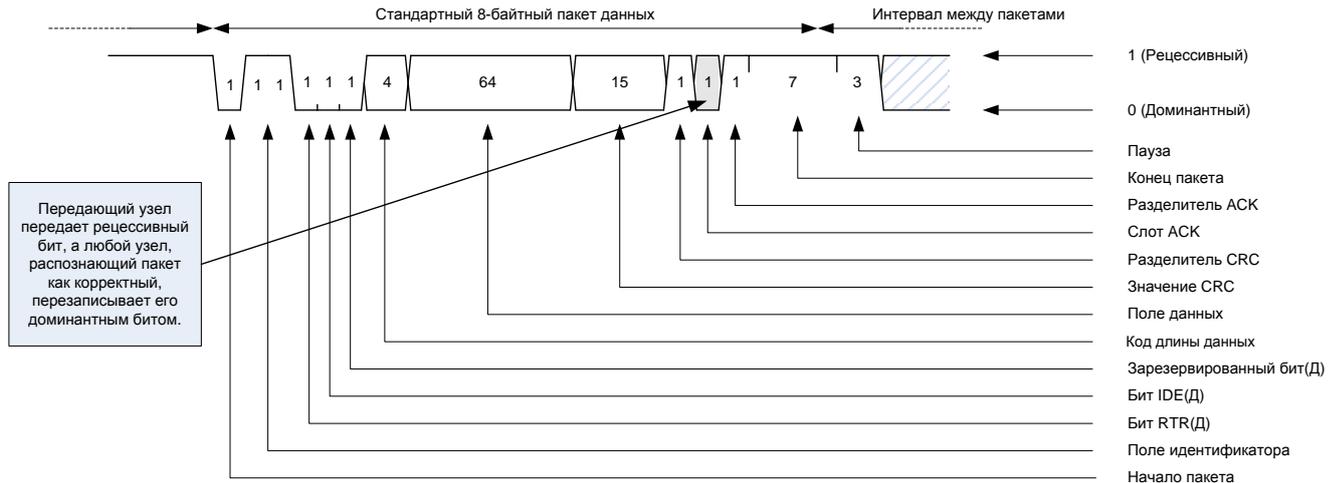


Рисунок 127 – Контроль подтверждения

Пакет сообщения CAN содержит 15-битное значение CRC, которое автоматически генерируется передатчиком и проверяется приемником. С помощью этого кода можно обнаружить и исправить ошибку в 4-х битах сообщения от начала кадра до начала поля CRC. Если CRC неверен и сообщение игнорируется, то передается кадр ошибки и в регистре статуса контроллера STATUS будет установлен флаг ошибки контрольной суммы пакета CRC_ERR.

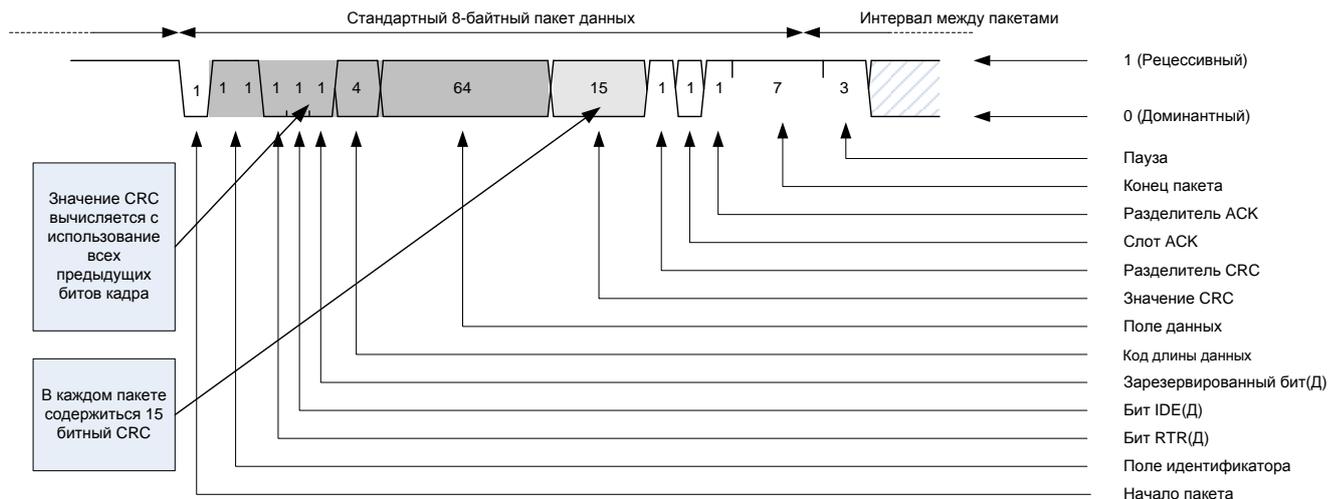


Рисунок 128 – Контроль CRC

После того, как узел выиграет арбитраж, он начинает передачу своего сообщения по шине. Как и во время арбитража, CAN-контроллер считывает обратно каждый бит, выдаваемый им на шину. Поскольку узел уже выиграл арбитраж, больше никто не должен передавать данные на шине, поэтому значение каждого выданного на шину бита должно соответствовать значению, считанному обратно с шины. Если считано неверное значение, передатчик генерирует кадр ошибки, в регистре статуса контроллера STATUS устанавливает флаг ошибки передаваемых бит пакета BIT_ERR и сообщение снова ставит в очередь. Это сообщение будет послано в следующем слоте сообщений, однако при этом оно должно пройти через процесс арбитража с другими запланированными сообщениями.

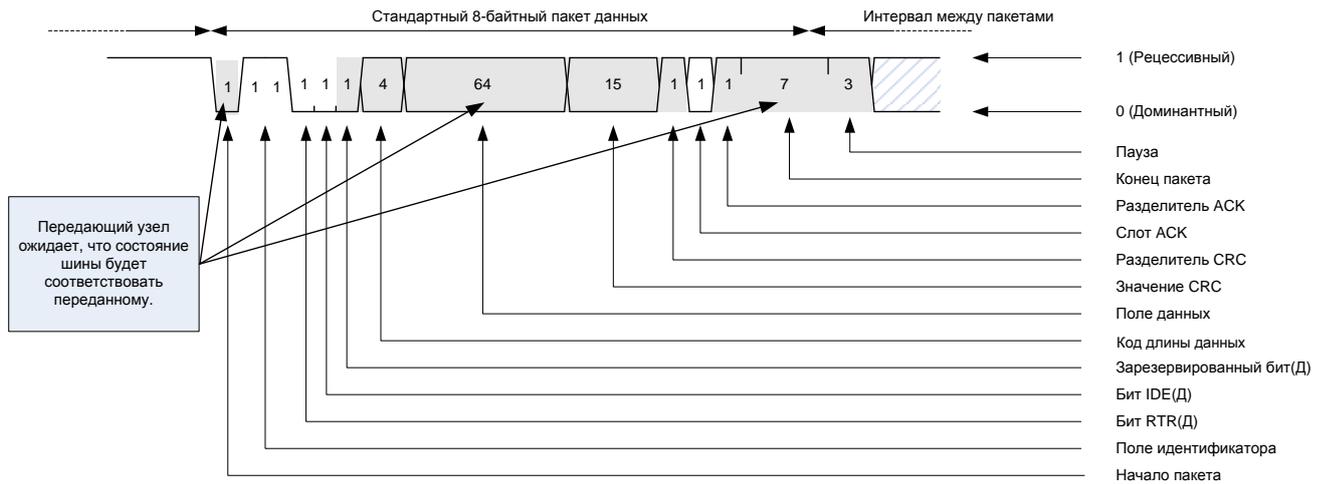


Рисунок 129 – Контроль передаваемых бит

На уровне бит в протоколе CAN реализован также метод вставки бита (битстаффинг). После каждой последовательности из пяти доминантных бит вставляется рецессивный бит; если рецессивный бит не обнаружен, в регистре статуса устанавливается флаг ошибки вставленных бит пакета BIT_STUF_ERR. Этот метод позволяет предотвратить появление на шине постоянных уровней и обеспечивает наличие в потоке бит достаточного количества переходов, используемых для повторной синхронизации. Кадр ошибки в протоколе CAN представляет собой простую последовательность из шести доминантных бит. Это позволяет любому контроллеру CAN формировать на шине сообщение об ошибке сразу после ее обнаружения, не дожидаясь конца сообщения.

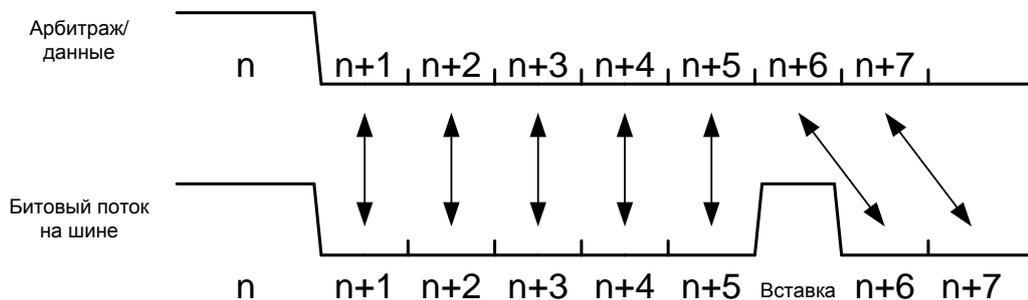


Рисунок 130 – Битстаффинг

В каждом CAN контроллере имеется два счетчика. Этими счетчиками являются счетчик ошибок приема (регистр STATUS, поле RX_ERR_CNT) и счетчик ошибок передачи (регистр STATUS, поле TX_ERR_CNT). Изменение состояния этих счетчиков происходит при приеме или передаче кадра ошибки. Когда любой счетчик достигает значения 128, контроллер CAN переходит в режим «error passive». В этом режиме он продолжает отзываться на кадры ошибки, однако при генерации кадра ошибки он вместо доминантных бит выставляет на шину рецессивные. Если счетчик ошибок передачи достигает значения 255, то контроллер CAN переходит в режим «bus-off» и больше не принимает участия в обмене по шине. Для возобновления обмена необходимо вмешательство процессора, который повторно инициализирует контроллер и подключает его обратно к шине. Текущий статус состояния контроллера можно посмотреть в регистре статуса контроллера STATUS.

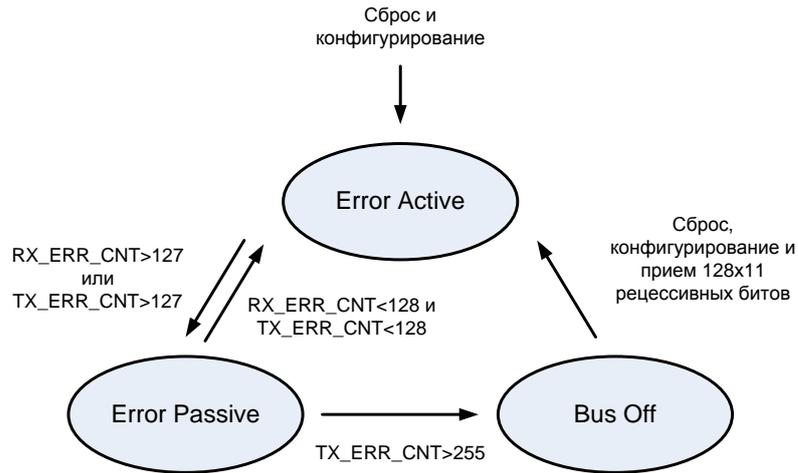


Рисунок 131 – Счетчики ошибок

Контроллер CAN имеет несколько механизмов обнаружения ошибок. Во-первых, из регистра состояния контроллера CAN_STATUS можно считать текущее состояние счетчиков ошибок приема и передачи. Также в этом регистре содержится флаг превышения счетчиками ошибок порогового значения ERROR_OVER. Это значение произвольно и записывается в регистр CAN_OVER. Как и регистры синхронизации, регистр CAN_OVER можно изменять только при нахождении контроллера в состоянии сброса.

7.18.15 Прерывания

В контроллере CAN в качестве источников прерывания выступают буфера сообщений. Генерируемые прерывания делятся на три группы:

- Прерывания передачи (по одному для каждого буфера);
- Прерывания приема (по одному для каждого буфера);
- Прерывания ошибки.

При возникновении какого-либо прерывания и наличии сигналов разрешения этих прерываний, буфер вырабатывает прерывание. Контроллер CAN объединяет прерывания приема, передачи и ошибки в каждом буфере и вырабатывает прерывание, отображаемое в регистре прерываний периферии. Если прерывание разрешено в регистре, процессор выполняет переход на обработчик прерываний. Обработчик прерываний должен выполнить действия по обработке прерывания и снять его выставление. Прерывание передачи/приема для каждого буфера может быть замаскировано путем установления соответствующего бита в регистрах CAN_INT_TX/CAN_INT_RX. Также есть возможность группового маскирования прерываний по приему, по передаче и по ошибке.

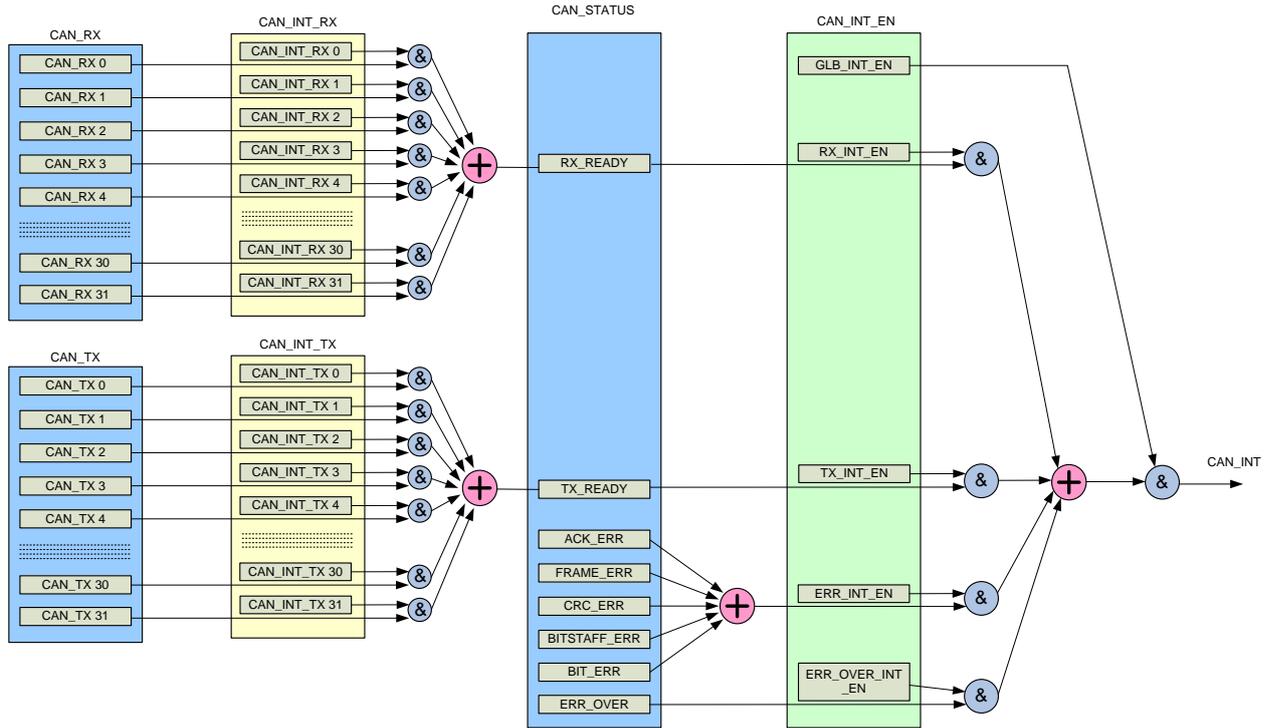


Рисунок 132 – Схема формирования прерывания блока CAN

7.18.16 Описание регистров

Таблица 108 – Описание регистров контроллера CAN

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4009_0000		MDR_CAN0		
	0x0000_0000	CONTROL		Регистр управление контроллером CAN
	0x0000_0004	STATUS		Регистр состояния контроллера CAN
	0x0000_0008	BITTMNG		Регистр задания скорости работы
	0x0000_0010	INT_EN		Регистр разрешения прерываний контроллера
	0x0000_001C	OVER		Регистр границы счетчика ошибок
	0x0000_0020	RXID		Регистр принятого ID сообщения
	0x0000_0024	RXDLC		Регистр принятого DLC сообщения
	0x0000_0028	RXDATAH		Регистр принятых данных
	0x0000_002C	RXDATAH		Регистр принятых данных
	0x0000_0030	TXID		Регистр передаваемого ID сообщения
	0x0000_0034	TXDLC		Регистр передаваемого DLC сообщения
	0x0000_0038	DATAL		Регистр передаваемых данных
	0x0000_003C	DATAH		Регистр передаваемых данных
	0x0000_0040	BUF_CON[0]		Регистр управления буфером 00
		...		
	0x0000_00BC	BUF_CON[31]		Регистр управления буфером 31
	0x0000_00C0	INT_RX		Флаги разрешения прерываний от приемных буферов
	0x0000_00C4	RX		Флаги RX_FULL от приемных буферов
	0x0000_00C8	INT_TX		Флаги разрешения прерываний от передающих буферов
	0x0000_00CC	TX		Флаги ~TX_REQ от передающих буферов
	0x0000_0200	BUF[0].ID		ID сообщения буфера 00
	0x0000_0204	BUF[0].DLC		DLC сообщения буфера 00
	0x0000_0208	BUF[0].DATAL		Данные сообщения буфера 00
	0x0000_020C	BUF[0].DATAH		Данные сообщения буфера 00
	0x0000_0210	BUF[1].ID		ID сообщения буфера 01
		...		
	0x0000_03FC	BUF[31].DATAH		Данные сообщения буфера 31
	0x0000_0400-0x0000_07FF			Зарезервировано
	0x0000_0500	FILTER[0].MASK		Маска для приема сообщения в буфер 00
	0x0000_0504	FILTER[0].FILTER		Фильтр для приема сообщения в буфер 00
	0x0000_0508	FILTER[1].MASK		Маска для приема сообщения в буфер 01
		...		
	0x0000_05FC	FILTER[31].FILTER		Фильтр для приема сообщения в буфер 31

7.18.16.1 CONTROL

Base ADDR= 0x4009_0000				Offset= 0x0000_0000											
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ROP	SAP	STM	ROM
															CANEN

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ROP	0	Прием собственных пакетов (ReceiveOwnPackets): 1 – контроллер принимает собственные пакеты; 0 – контроллер принимает только чужие пакеты
3	SAP	0	Подтверждение собственных пакетов (SendACKOwnPackets): 1 – контроллер подтверждает прием собственных пакетов; 0 – контроллер подтверждает прием только чужих пакетов
2	STM	0	Режим самотестирования (SelfTestMode): 1 – контроллер работает в режиме самотестирования; 0 – контроллер работает в нормальном режиме
1	ROM	0	Режим «Только прием» (ReadOnlyMode): 1 – контроллер работает только на прием; 0 – контроллер работает в нормальном режиме
0	CAN_EN	0	Режим работы контроллера CAN: 1 – разрешение работы; 0 – сброс

7.18.16.2 STATUS

Base ADDR= 0x4009_0000					Offset= 0x0000_0004										
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXERRCNT [7:0]								RXERRCNT [7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TXERRCNT 8	RXERRCNT 8	ERRSTATUS [1:0]		IDLOWER	ACKERR	FRAMEERR	CRCERR	BITSTUFF ERR	BITERR	ERROR OVER	TX READY	RX READY

Бит	Имя	Значение	Описание
31...24	TXERRCNT [7:0]	0x00	Счетчик ошибок передатчика TEC, биты [7:0]: TEC > 127, ERROR PASSIVE
23...16	RXERRCNT [7:0]	0x00	Счетчик ошибок приемника REC, биты [7:0]: REC > 127, ERROR PASSIVE
15...13	-		
12	TXERRCNT8	0	Счетчик ошибок передатчика TEC, бит 8: 0 – TEC менее 255; 1 – TEC более 255
11	RXERRCNT8	0	Счетчик ошибок приемника REC, бит 8: 0 – REC менее 255; 1 – REC более 255
10...9	ERRSTATUS[1:0]	00	Статус состояния контроллера CAN: 00 – ERRORACTIVE, при возникновении ошибки отсылается флаг активной ошибки; 01 – ERRORPASSIVE, при возникновении ошибки отсылается флаг пассивной ошибки; 1x – BUSOFF, ожидается восстановление шины
8	ID LOWER	0	Флаг «проигрыша» арбитража: 0 – при передаче не было проигрыша арбитража; 1 – при передаче был проигран арбитраж
7	ACK ERR	0	Флаг ошибки подтверждения приема: 0 – нет ошибки; 1 – есть ошибка
6	FRAME ERR	0	Флаг ошибки формата пакета: 0 – нет ошибки; 1 – есть ошибка
5	CRC ERR	0	Флаг ошибки контрольной суммы пакета: 0 – нет ошибки; 1 – есть ошибка
4	BIT STUFF ERR	0	Флаг ошибки вставленных бит пакета: 0 – нет ошибки; 1 – есть ошибка
3	BIT ERR	0	Флаг ошибки передаваемых бит пакета: 0 – нет ошибки; 1 – есть ошибка
2	ERROR OVER	0	Флаг превышения TEC и REC уровня, заданного ERROR_MAX: 0 – ERROR_MAX < TEC и REC; 1 – ERROR_MAX ≥ TEC или REC
1	TX READY	0	Флаг наличия буферов для отправки: 0 – нет буферов готовых для отправки сообщений; 1 – есть буфер готовый для отправки сообщений
0	RX READY	0	Флаг наличия принятых сообщений: 0 – нет буферов с принятыми сообщениями; 1 – есть буфер с принятым сообщением

7.18.16.3 BITTMNG

Base ADDR=		0x4009_0000				Offset=		0x0000_0008									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
				SB	SJW[1:0]		SEG2[2:0]			SEG1[2:0]			PSEG[2:0]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP[15:0]															

Бит	Имя	Значение	Описание
31...28	-		Зарезервировано
27	SB	0	Сэмплирование: 0 – однократное; 1 – трехкратное с мажоритарным контролем
26...25	SJW [1:0]	00	Значение размера фазы SJW: 11 = Synchronization jump width time = 4 x TQ 10 = Synchronization jump width time = 3 x TQ 01 = Synchronization jump width time = 2 x TQ 00 = Synchronization jump width time = 1 x TQ SJW – это максимальное значение, на которое происходит подстройка приема и передачи при работе на шине CAN. Приемник подстраивается на значение ошибки, но не более чем SJW
24...22	SEG2 [2:0]	000	Значение размера фазы SEG2: 111 = Phase Segment 2 time = 8 x TQ 110 = Phase Segment 2 time = 7 x TQ 101 = Phase Segment 2 time = 6 x TQ 100 = Phase Segment 2 time = 5 x TQ 011 = Phase Segment 2 time = 4 x TQ 010 = Phase Segment 2 time = 3 x TQ 001 = Phase Segment 2 time = 2 x TQ 000 = Phase Segment 2 time = 1 x TQ SEG2 – это время, используемое для сокращения битового интервала при подстройке
21...19	SEG1 [2:0]	000	Значение размера фазы SEG1: 111 = Phase Segment 1 time = 8 x TQ 110 = Phase Segment 1 time = 7 x TQ 101 = Phase Segment 1 time = 6 x TQ 100 = Phase Segment 1 time = 5 x TQ 011 = Phase Segment 1 time = 4 x TQ 010 = Phase Segment 1 time = 3 x TQ 001 = Phase Segment 1 time = 2 x TQ 000 = Phase Segment 1 time = 1 x TQ SEG1 – это время, используемое для увеличения битового интервала при подстройке

Бит	Имя	Значение	Описание
18...16	PSEG[2:0]	000	Значение размера фазы PSEG 111 = Propagation time = 8 x TQ 110 = Propagation time = 7 x TQ 101 = Propagation time = 6 x TQ 100 = Propagation time = 5 x TQ 011 = Propagation time = 4 x TQ 010 = Propagation time = 3 x TQ 001 = Propagation time = 2 x TQ 000 = Propagation time = 1 x TQ PSEG - это время компенсирующее задержку распространения сигналов в шине CAN
15...0	BRP [15:0]	0x0000	Предделитель системной частоты: CLK = CANCLK0/(BRP + 1) TQ (us) = 1/CLK (MHz) = (BRP + 1)/CANCLK0 (MHz)

7.18.16.4 INT_EN

Base ADDR= 0x4009_0000					Offset= 0x0000_0010										
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ERROVER INTN	ERRINTEN	TXINTEN	RXINTEN	GLBINTEN

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ERROVER INTEN	0	Флаг разрешения прерывания по превышению TEC или REC допустимого значения в ERROR_MAX: 0 – запрещено прерывание; 1 – разрешено прерывание
3	ERRINT EN	0	Флаг разрешения прерывания по возникновению ошибки: 0 – запрещено прерывание; 1 – разрешено прерывание
2	TXINT EN	0	Флаг разрешения прерывания по возможности передачи: 0 – запрещено прерывание; 1 – разрешено прерывание
1	RXINT EN	0	Флаг разрешения прерывания по приему сообщений: 0 – запрещено прерывание; 1 – разрешено прерывание
0	GLBINT EN	0	Общий флаг разрешения прерывания блока CAN: 0 – запрещено прерывание; 1 – разрешено прерывание

7.18.16.5 OVER

Base ADDR=		0x4009_0000				Offset=		0x0000_001C								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ERROR_MAX[7:0]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ERRORMAX [7:0]	0x00	Регистр границы счетчика ошибок Допустимое значение счетчиков ошибок TEC и REC, при превышении которого вырабатывается флаг ERROR_OVER

7.18.16.6 BUF_CON[x]

Base ADDR= 0x4009_0000				Offset=				0x0000_0040								
								..								
								0x0000_00BC								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								OVERWR	RXFULL	TXREQ	PRIOR 0	RTREN	OVEREN	RXTXn	EN

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	OVER_WR	0	Флаг перезаписи принятого сообщения: 0 – не было перезаписи; 1 – была перезапись принятого сообщения
6	RX_FULL	0	Флаг готовности приема: 0 – нет принятого сообщения; 1 – принятое сообщение в буфере
5	TX_REQ	0	Запрос на отправку сообщения: 0 – нет запроса или отправлено; 1 – запрос на отправку
4	PRIOR_0	0	Приоритет при отправке: 0 – приоритет; 1 – нет приоритета
3	RTR_EN	0	Режим ответа на RTR: 0 – не отвечать при приеме RTR; 1 – ответить при приеме RTR в буфер
2	OVER_EN	0	Разрешение перезаписи принятого сообщения: 0 – не разрешена перезапись; 1 – разрешена перезапись сообщения
1	RX_TXn	0	Режим работы буфера: 0 – на передачу; 1 – на прием
0	EN	0	Разрешение работы буфера: 0 – отключен; 1 – работает

7.18.16.7 INT_RX

Base ADDR=	0x4009_0000	Offset=	0x0000_00C0												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_INT_RX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_INT_RX[15:0]															

Бит	Имя	Значение	Описание
31...0	CAN_INT_RX[31:0]	0x00000000	Флаги разрешения прерываний от буферов по приему сообщений: 0 – прерывание запрещено 1 – прерывание разрешено CAN_INT_RX[0] – для первого буфера CAN_INT_RX[1] – для второго буфера и так далее

7.18.16.8 RX

Base ADDR=	0x4009_0000	Offset=	0x0000_00C4												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_RX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_RX[15:0]															

Бит	Имя	Значение	Описание
31...0	CAN_RX[31:0]	0x00000000	Флаги RX_FULL разрешенных на прием буферов: CAN_RX[0] – флаг RX_FULL от первого буфера CAN_RX[1] – флаг RX_FULL от второго буфера и так далее

7.18.16.9 INT_TX

Base ADDR=		0x4009_0000				Offset=		0x0000_00C8									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
CAN_INT_TX[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_INT_TX[15:0]															

Бит	Имя	Значение	Описание
31...0	CAN_INT_TX[31:0]	0x00000000	Флаги разрешения прерываний от буферов по передаче сообщений: CAN_INT_TX[0] – для первого буфера CAN_INT_TX[1] – для второго буфера и так далее

7.18.16.10 TX

Base ADDR=		0x4009_0000				Offset=		0x0000_00CC									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
CAN_TX[31:16]																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_TX[15:0]															

Бит	Имя	Значение	Описание
31...0	CAN_TX[31:0]	0x00000000	Флаги ~TX_REQ разрешенных на передачу буферов: CAN_TX[0] – флаг ~TX_REQ от первого буфера CAN_TX[1] – флаг ~TX_REQ от второго буфера и так далее, доступны только на чтение

**7.18.16.11 RXID
TXID
BUF[x].ID
FILTER[x].MASK
FILTER[x].FILTER**

Base ADDR=	0x4009_0000	Offset=	0x0000_0020 0x0000_0030 0x0000_0200 0x0000_0210 .. 0x0000_03F0 0x0000_0500 0x0000_0504 .. 0x0000_05FC												
------------	-------------	---------	--	--	--	--	--	--	--	--	--	--	--	--	--

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SID[10:0]											EID[17:16]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EID[15:0]															

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...18	SID [10:0]	0x000*	Поле SID. Для стандартного и расширенного пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже
17...0	EID [17:0]	0x00000*	Поле EID. Для расширенных пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже

* Регистры BUF[x].ID FILTER[x].MASK FILTER[x].FILTER не инициализируются сбросом.

**7.18.16.12 RXDLC
TXDLC
BUF[x].DLC**

Base ADDR=	0x4009_0000	Offset=	0x0000_0024 0x0000_0034 0x0000_0204 0x0000_0214 .. 0x0000_03F4												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IDE	SRR	R0	R1	RTR							DLC[3:0]	

Бит	Имя	Значение	Описание
31...13	-	0*	Зарезервировано
12	IDE	0*	Поле IDE. Поле, обозначающее формат пакета: 0 – стандартный пакет; 1 – расширенный пакет
11	SRR	0*	Поле SRR, расширенного формата. Всегда должно быть равно “1”
10	R0	0*	Поле R0. Всегда должно быть равно “0”
9	R1	0*	Поле R1, расширенного формата. Всегда должно быть равно “1”
8	RTR	0*	Поле RTR, запроса обратного ответа: 0 – нет запроса; 1 – есть запрос. Если узел получил пакет с запросом обратного ответа, то он должен ответить
7...4	-	0*	Зарезервировано
3...0	DLC[3:0]	0000*	Поле DLC, длина передаваемых данных в пакете: 0000 – нет данных 0001 – 1 байт 0010 – 2 байт 0011 – 3 байт 0100 – 4 байт 0101 – 5 байт 0110 – 6 байт 0111 – 7 байт 1000 – 8 байт 1xxx – 8 байт и недопустимо

* Регистры BUF[x].DLC не инициализируются сбросом.

**7.18.16.13 RXDATAL
TXDATAL
BUF[x].DATAL**

Base ADDR=	0x4009_0000					Offset=	0x0000_0028 0x0000_0038 0x0000_0208 0x0000_0218 .. 0x0000_03F8									
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DB3[7:0]								DB2[7:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DB1[7:0]								DB0[7:0]								

Бит	Имя	Значение	Описание
31...24	DB3[7:0]	0x00*	Поле DB3. Четвертый байт, передаваемый в пакете
23...16	DB2[7:0]	0x00*	Поле DB2. Третий байт, передаваемый в пакете
15...8	DB1[7:0]	0x00*	Поле DB1. Второй байт, передаваемый в пакете
7...0	DB0[7:0]	0x00*	Поле DB0. Первый байт, передаваемый в пакете

* Регистры BUF[x].DATAL не инициализируются сбросом.

**7.18.16.14 RXDATAH
TXDATAH
BUF[x].DATAH**

Base ADDR=	0x4009_0000	Offset=	0x0000_002C 0x0000_003C 0x0000_020C 0x0000_021C .. 0x0000_03FC												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							

Бит	Имя	Значение	Описание
31...24	DB7[7:0]	0x00*	Поле DB7. Четвертый байт, передаваемый в пакете
23...16	DB6[7:0]	0x00*	Поле DB5. Третий байт, передаваемый в пакете
15...8	DB5[7:0]	0x00*	Поле DB4. Второй байт, передаваемый в пакете
7...0	DB4[7:0]	0x00*	Поле DB3. Первый байт, передаваемый в пакете

* Регистры BUF[x].DATAH не инициализируются сбросом.

7.18.16.15 BUF[x]

Base ADDR=	0x4009_0000	Offset=	0x0000_0200																
			0x0000_0210																
			..																
			0x0000_03F0																
REG Name:																			
127																			112
DATAH[31:0]																			
111																			96
95																			80
DATAI[31:0]																			
79																			64
63																			48
DLC[31:0]																			
47																			32
31																			16
ID[31:0]																			
15																			0

Бит	Имя	Значение	Описание
127...96	DATAH[31:0]		Данные сообщения
95...64	DATAI[31:0]		Данные сообщения
63...32	DLC[31:0]		DLC сообщения
31...0	ID[31:0]		ID сообщения

7.18.16.16 FILTER[x]

Base ADDR=	0x4009_0000	Offset=	0x0000_0500 0x0000_0508 .. 0x0000_05F8												
REG Name:															
63															48
FILTER[31:0]															
47															32
31															16
MASK[31:0]															
15															0

Бит	Имя	Значение	Описание
63...32	FILTER[31:0]		Фильтр для приёма сообщения
31...0	MASK[31:0]		Маска для приёма сообщения

7.19 Контроллер SSP

Модуль порта синхронной последовательной связи (SSP – Synchronous Serial Port) выполняет функции интерфейса последовательной синхронной связи в режиме ведущего и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из протоколов:

- интерфейс SPI фирмы Motorola;
- интерфейс SSI фирмы Texas Instruments;
- интерфейс Microwire фирмы National Semiconductor.

Как в ведущем, так и в ведомом режиме работы модуль SSP обеспечивает:

- преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 16-разрядных ячеек данных) из параллельного в последовательный формат;
- преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 16-разрядных ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута.

Основные сведения о модуле представлены в следующих разделах:

- характеристики интерфейса SPI;
- характеристики интерфейса Microwire;
- характеристики интерфейса SSI.

Основные характеристики модуля SSP:

- функционирование как в ведущем, так и в ведомом режиме;
- программное управление скоростью обмена;
- состоит из независимых буферов приема и передачи (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
- поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке .

7.19.1 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- режим функционирования периферийного устройства – ведущее или ведомое;
- разрешение или запрещение функционирования;
- формат информационного кадра;
- скорость передачи данных;
- фаза и полярность тактового сигнала;
- размер блока данных – от 4 до 16 бит;
- маскирование прерываний.

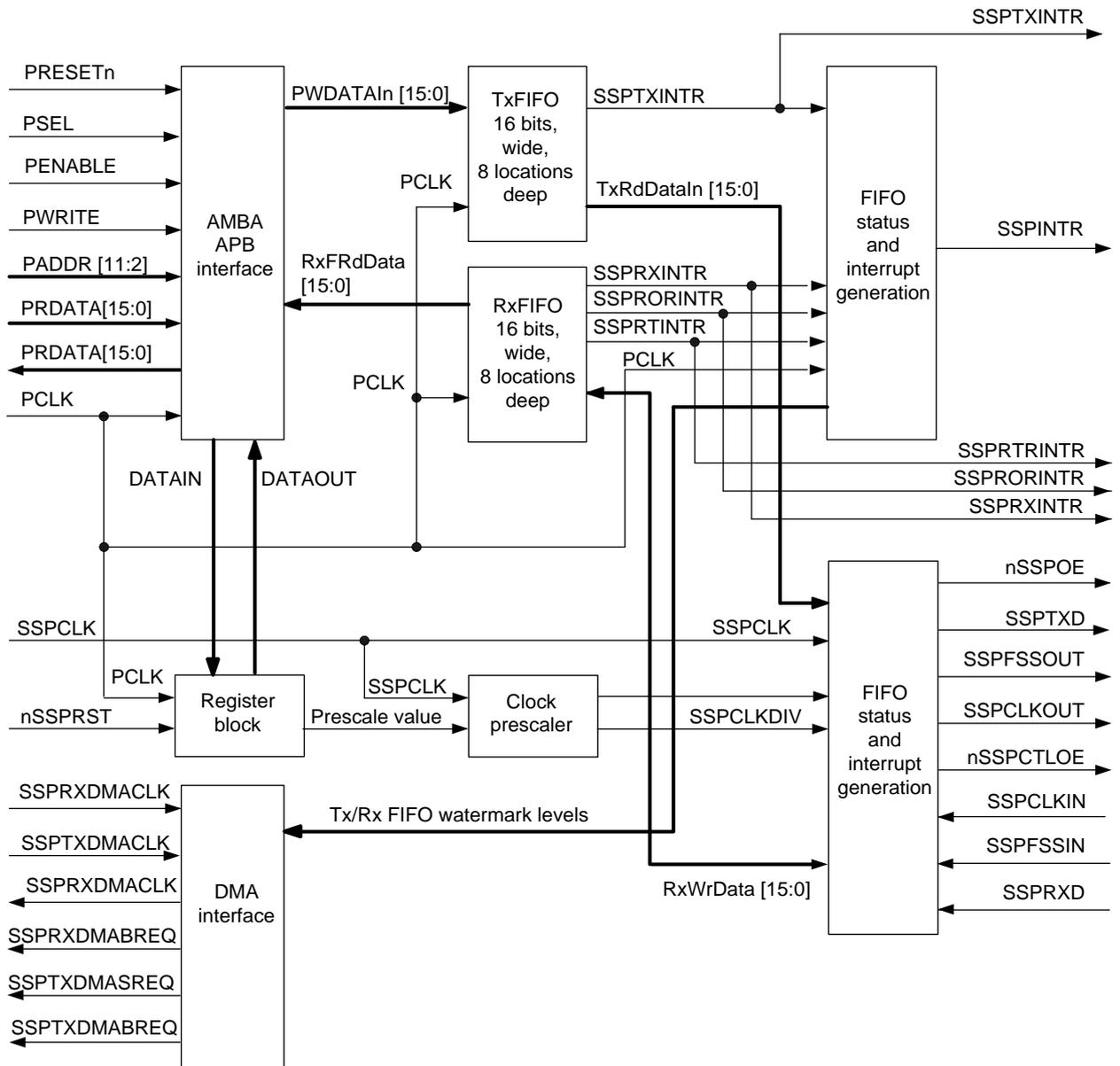


Рисунок 133 – Структурная схема модуля SSP

7.19.1.1 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- программное задание фазы и полярности тактового сигнала.

7.19.1.2 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

7.19.1.3 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

7.19.2 Общий обзор модуля SSP

Модуль SSP представляет собой интерфейс синхронного последовательного обмена данными, способный функционировать в качестве ведущего или ведомого устройства и поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Модуль выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму;
- центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии;
- прием и передача данных буферизуются с помощью буферов FIFO, обеспечивающих хранение до восьми слов данных шириной 16 бит, независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSP_TXD и принимаются с линии SSP_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSP_CLK из сигнала, поступающего на линию SSPCLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSPCLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления CR0 и CR1.

Модуль формирует четыре независимо маскируемых прерывания:

- SSPTXINTR – запрос на обслуживание буфера передатчика;
- SSPRXINTR – запрос на обслуживание буфера приемника;
- SSPRORINTR – переполнение приемного буфера FIFO;
- SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO.

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний, который идет на контроллер NVIC.

Модуль также формирует сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В зависимости от режима работы модуля сигнал SSPFSSOUT используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

7.19.2.1 Блок формирования тактового сигнала

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP_CLK с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Путем записи значения в регистр SSPCPSR можно задать коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2. Так как младший значащий разряд коэффициента деления не используется, то исключается возможность деления частоты на нечетный коэффициент деления. Это, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровней).

Сформированный описанным образом сигнал далее поступает на второй делитель частоты, с выхода которого и снимается тактовый сигнал обмена данными SSP_CLK. Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256, путем записи соответствующего значения в регистр управления SSPCR0.

7.19.2.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO – «первый вошел, первый вышел». Данные от центрального процессора сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

7.19.2.3 Буфер FIFO приемника

Буфер приемника имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO – «первый вошел, первый вышел». Принятые от периферийного устройства данные сохраняются в этом буфере блоком приема данных до тех пор, пока не будут считаны центральным процессором.

7.19.2.4 Блок приема и передачи данных

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP_CLK для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает данные из буфера FIFO передатчика и производит их преобразование из параллельной формы в последовательную. Далее поток последовательных данных и элементов кадровой синхронизации, тактированный сигналом SSP_CLK, передаётся по линии SSP_TXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSP_RXD, из последовательной в параллельную форму. Затем загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSP_CLK.

При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную, после чего выдает поток последовательных данных и элементов кадровой синхронизации в линию SSP_TXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSP_RXD синхронно с сигналом SSP_CLK, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

7.19.2.5 Блок формирования прерываний

Модуль SSP генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания подаётся на контроллер прерываний NVIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

7.19.2.6 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

7.19.2.7 Конфигурирование приемопередатчика

После сброса работа блоков приемопередатчика запрещается до выполнения процедуры задания конфигурации.

Для этого необходимо выбрать ведущий или ведомый режим работы устройства, а также используемый протокол передачи данных (SPI фирмы Motorola, SSI фирмы Texas Instruments, либо Microwave фирмы National Semiconductor), после чего записать необходимую информацию в регистры управления CR0 и CR1.

Кроме того, для установки требуемой скорости передачи данных необходимо выбрать параметры блока формирования тактового сигнала с учетом значения частоты сигнала SSPCLK и записать соответствующую информацию в регистр PSR.

7.19.2.8 Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSP_TXD и SSP_RXD.

7.19.2.9 Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов CPU_CLK и SSPCLK. Частота SSPCLK должна быть меньше или равна частоте CPU_CLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала CPU_CLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

$$FSSPCLK \leq FPCLK$$

В режиме ведомого устройства сигнал SSP_CLK от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того, чтобы обнаружить фронт сигнала SSP_CLK, необходимо три такта сигнала SSP_CLK. Сигнал SSP_TXD имеет меньшее время установки по отношению к заднему фронту SSP_CLK, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP_RXD по отношению к сигналу SSP_CLK должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSP_CLK.

Выбор частоты тактового сигнала SSPCLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSP_CLK в режиме ведомого устройства равно 12, в режиме ведущего – двум.

Так, в режиме ведущего устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3,6864 МГц. В этом случае в регистр CPSR должно быть записано значение 2, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSPCLK с частотой не менее 22,12 МГц. При этом в регистр CPSR должно быть записано значение 12, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой SSPCLKOUT составляет $254 * 256$.

Минимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$\begin{cases} FSSPCLK_{min} \geq 2 \times FSSPCLKOUT_{max} \text{ (master mode)} \\ FSSPCLK_{min} \geq 12 \times FSSPCLKIN_{max} \text{ (slave mode)} \end{cases}$$

Аналогично, максимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$\begin{cases} FSSPCLK_{max} \leq 254 \times 256 \times FSSPCLKOUT_{min} \text{ (master mode)} \\ FSSPCLK_{max} \leq 254 \times 256 \times FSSPCLKIN_{min} \text{ (slave mode)}. \end{cases}$$

7.19.2.10 Программирование регистра управления CR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний, задается совместно - значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра SSPCR0 и значением поля CPSDVR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра SSPCPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS, регистра SSPCR0.

Для протокола SPI фирмы Motorola также задаются полярность и фаза сигнала (биты SPH и SPO).

7.19.2.11 Программирование регистра управления CR1

Регистр SSPCR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика;
- включения режима проверки канала по шлейфу;
- разрешения или запрещения работы модуля.

Выбор ведущего режима осуществляется путем записи 0 в поле MS, регистра SSPCR1 (это значение устанавливается после сброса автоматически).

Запись 1 в поле MS переводит приемопередатчик в режим ведомого устройства. В этом режиме разрешение или запрещение формирования сигнала передатчика SSP_TXD осуществляется путем установки бита SOD (slave mode SSP_TXD output disable – запрет линии SSP_TXD для ведомого режима) регистра CR1. Указанная функция полезна при подключении к одной линии нескольких подчиненных устройств.

Для того чтобы разрешить функционирование приемопередатчика, необходимо установить в 1 бит SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта).

7.19.2.12 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота этого сигнала делится на четный коэффициент CPSDVR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр CPSR. Сформированный сигнал, далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирования через CR0.

Частота выходного тактового сигнала обмена данными SSP_CLK определяется следующим соотношением:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)} .$$

Например, в случае, если частота сигнала SSPCLK составляет 3.6864 МГц, а значение CPSDVR = 2, частота сигнала SSP_CLK лежит в интервале от 7.2 кГц до 1.8432 МГц.

7.19.2.13 Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Можно выбрать три базовых структуры построения кадра:

- SSI фирмы Texas Instruments;
- SPI фирмы Motorola;
- Microwire фирмы National Semiconductor.

Во всех трех режимах построения кадра тактовый сигнал SSP_CLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSP_CLK в неактивное состояние, используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSP_FSS имеет активный низкий уровень и поддерживается в низком уровне, в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments, перед началом каждого информационного кадра на выходе SSP_FSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSP_CLK, а считывает данные из линии, по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

7.19.2.14 Формат синхронного обмена SSI фирмы Texas Instruments

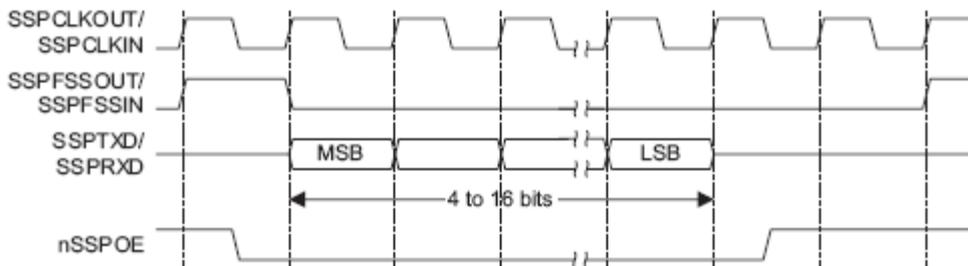


Рисунок 134 – Формат синхронного обмена протокола SSI (единичный обмен)

В данном режиме, при неактивном приемопередатчике SSP сигналы SSP_CLK и SSP_FSS переводятся в низкий логический уровень, а линия передачи данных SSP_TXD поддерживается в третьем состоянии.

После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSP_FSS переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSP_CLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP_CLK, старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP_TXD и т.д.

В режиме приема данных как модуль SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSP_CLK. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSP_CLK.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунках: Рисунок 134 – передача единичного информационного кадра, Рисунок 135 – передача последовательности кадров.

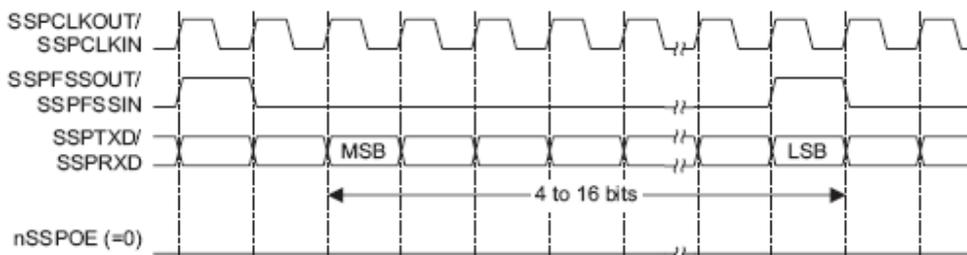


Рисунок 135 – Формат синхронного обмена протокола SSI (непрерывный обмен)

7.19.2.15 Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSP_FSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI, является возможность выбора состояния и фазы сигнала SSP_CLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления SSPSCR0.

Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен 0, то в режиме ожидания линия SSP_CLK переводится в низкий логический уровень. В противном случае, при отсутствии обмена данными, линия SSP_CLK переводится в высокий логический уровень.

Выбор фазы тактового сигнала – бит SPH

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае, если бит SPH установлен в 0, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала, в противном случае – после второго.

7.19.2.16 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=0 показаны на рисунках: Рисунок 136 – одиночный обмен, Рисунок 137 – непрерывный обмен.

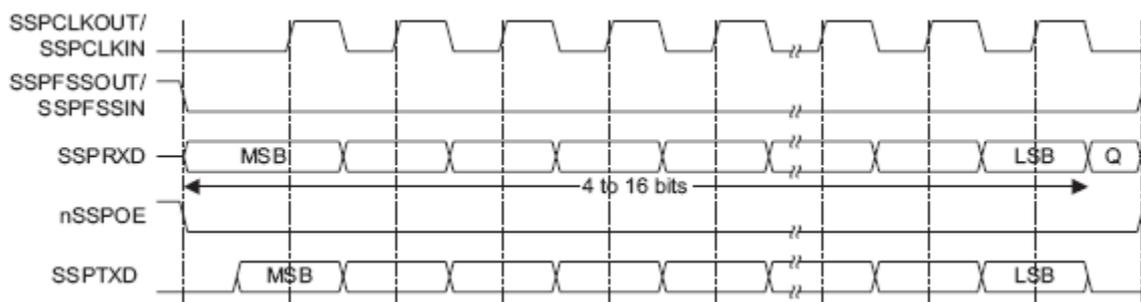


Рисунок 136 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиночный обмен)

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

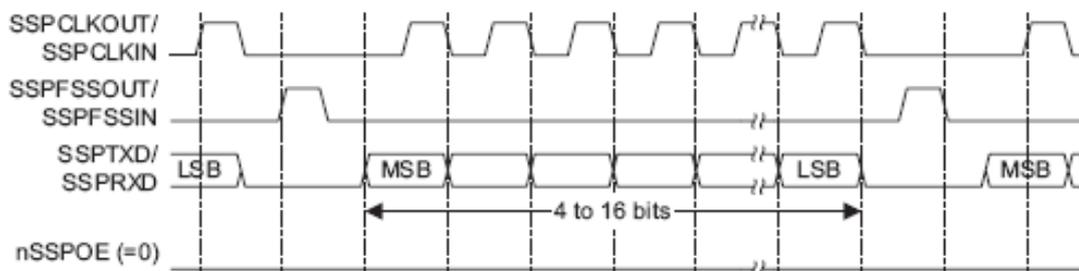


Рисунок 137 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK на линии SSP_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта, сигнал SSP_CLK переводится в высокий логический уровень.

Далее, данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных, после приема его последнего бита, линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных, на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита блока данных, линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

7.19.2.17 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показывает Рисунок 138 – одиночный и непрерывный обмен.

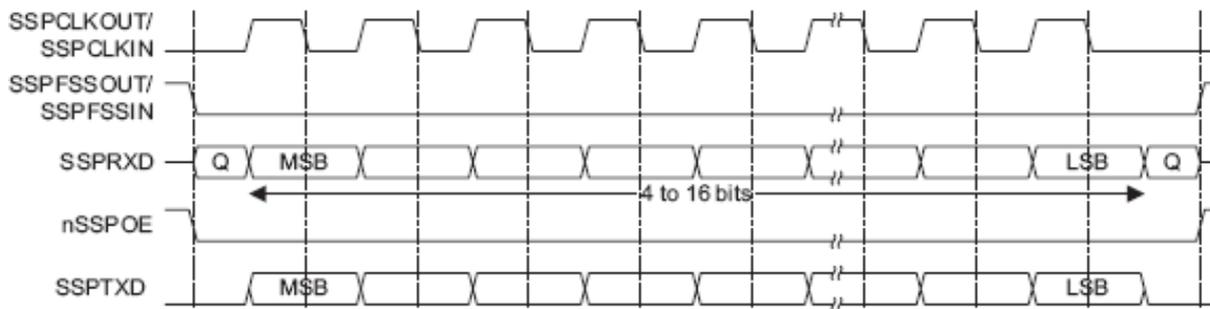


Рисунок 138 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK, на линиях обмена как ведущего, так и ведомого устройств будут сформированы значения первых бит передаваемых данных. В это же время, включается линия SSP_CLK и на ней формируется передний фронт сигнала.

Далее, данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных, после приема его последнего бита, линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных, линия SSP_FSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

7.19.2.18 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=0 показаны на рисунках: Рисунок 139 – одиночный обмен и Рисунок 140 – непрерывный обмен.

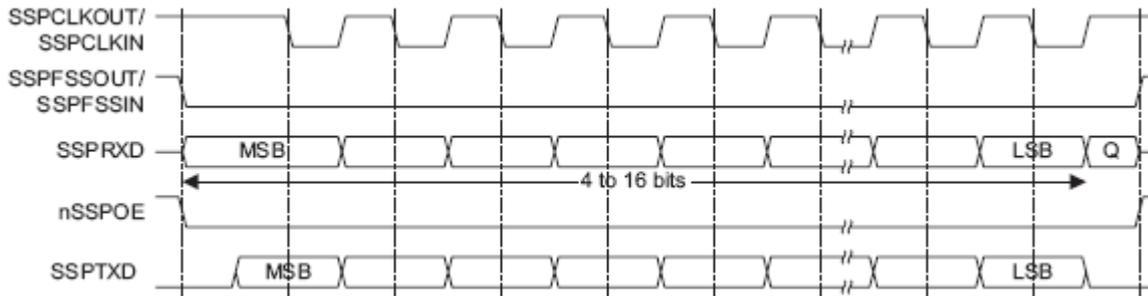


Рисунок 139 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (одиночный обмен)

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

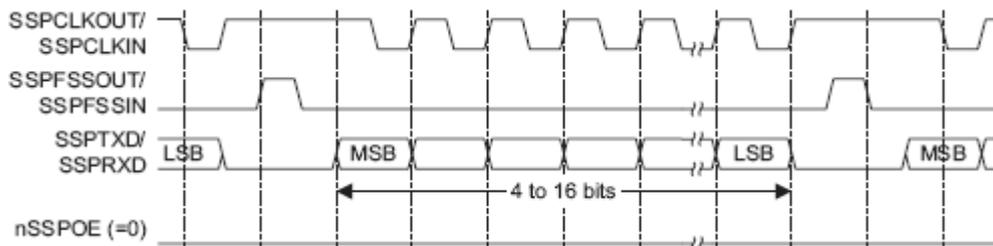


Рисунок 140 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет высокий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK, на линии SSP_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта, сигнал SSP_CLK переводится в низкий логический уровень.

Далее, данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных, после приема его последнего бита, линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных, на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита блока данных, линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

7.19.2.19 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=1 показывает Рисунок 141 – одиночный и непрерывный обмен.

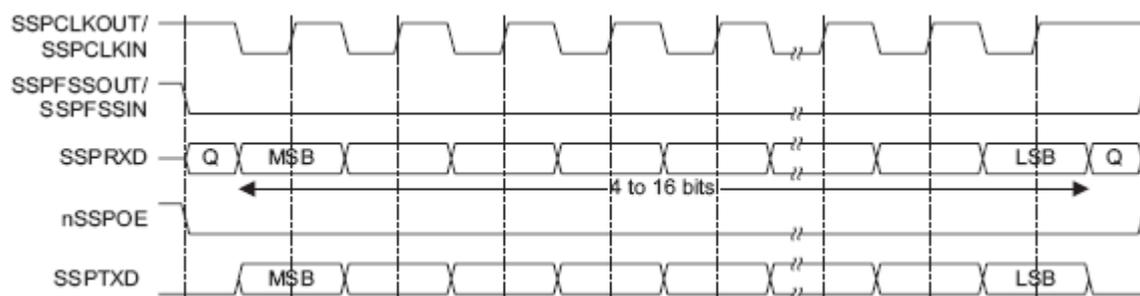


Рисунок 141 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет высокий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутака сигнала SSP_CLK, на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSP_CLK и на ней формируется передний фронт сигнала.

Далее, данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных, после приема его последнего бита, линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных, линия SSP_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

7.19.2.20 Формат синхронного обмена Microwire фирмы NationalSemiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунках: Рисунок 142 – одиночный обмен, Рисунок 143 – непрерывный обмен.

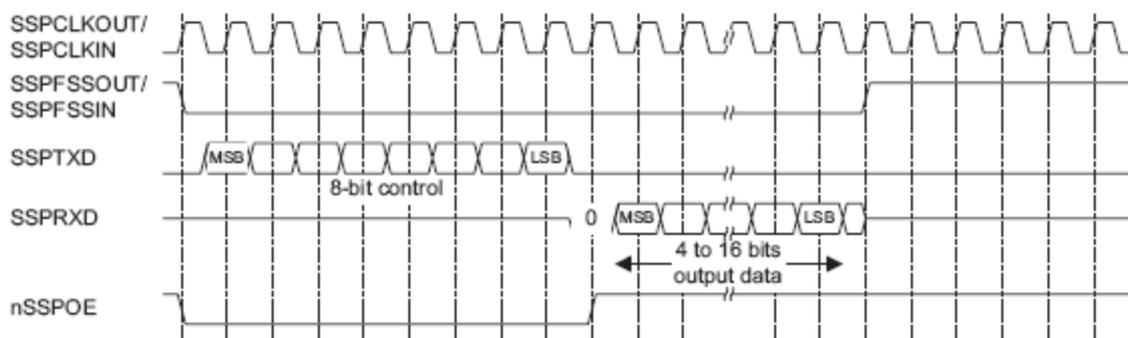


Рисунок 142 – Формат синхронного обмена протокола Microwire (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной 8-ми битной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSP_FSS данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSP_TXD. Линия SSP_FSS остается в низком логическом уровне в течение всей передачи кадра. Линия SSP_RXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSP_CLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSP_RXD по заднему фронту сигнала SSP_CLK. Ведущее устройство, в свою очередь, регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSP_FSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

Примечание – Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSP_CLK после приема последнего бита слова данных, либо после перевода линии SSP_FSS в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается так же, как и одиночный обмен. Однако линия SSP_FSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника, после регистрации младшего разряда очередного слова по заднему фронту сигнала SSP_CLK.

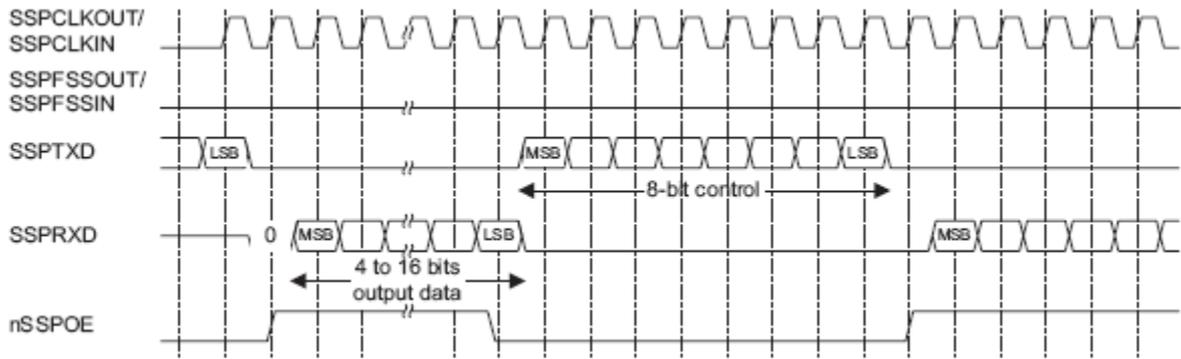


Рисунок 143 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

Требования к временным параметрам сигнала SSP_FSS относительно тактового сигнала SSP_CLK в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSP_CLK после установки сигнала SSP_FSS в низкий логический уровень. Ведущие устройства, формирующие сигнал SSP_CLK, должны гарантировать достаточное время установки и удержания сигнала SSP_FSS, по отношению к переднему фронту сигнала SSP_CLK.

Данные требования иллюстрирует Рисунок 144. По отношению к переднему фронту сигнала SSP_CLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSP_FSS должно быть как минимум в два раза больше периода SSP_CLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSP_CLK, должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

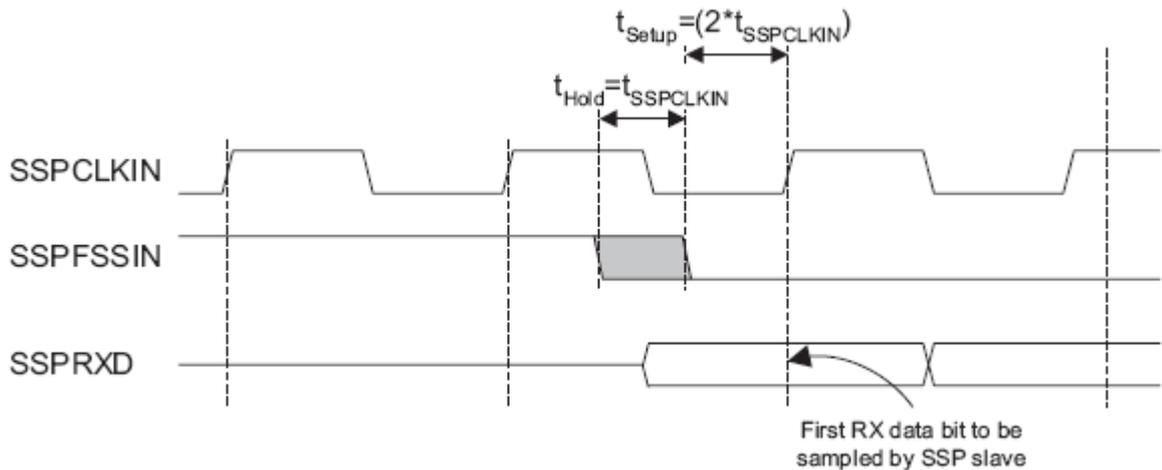


Рисунок 144 – Формат Microwire, требования к времени установки и удержания сигнала

7.19.2.21 Примеры конфигурации модуля в ведущем и ведомом режимах

На рисунках ниже (Рисунок 145, Рисунок 146 и Рисунок 147) показаны варианты подключения модуля SSP к периферийным устройствам, работающим в ведущем или ведомом режиме.

Примечание – Модуль SSP не поддерживает динамическое изменение режима «ведущий – ведомый». Каждый приёмопередатчик должен быть изначально сконфигурирован в одном из этих режимов.

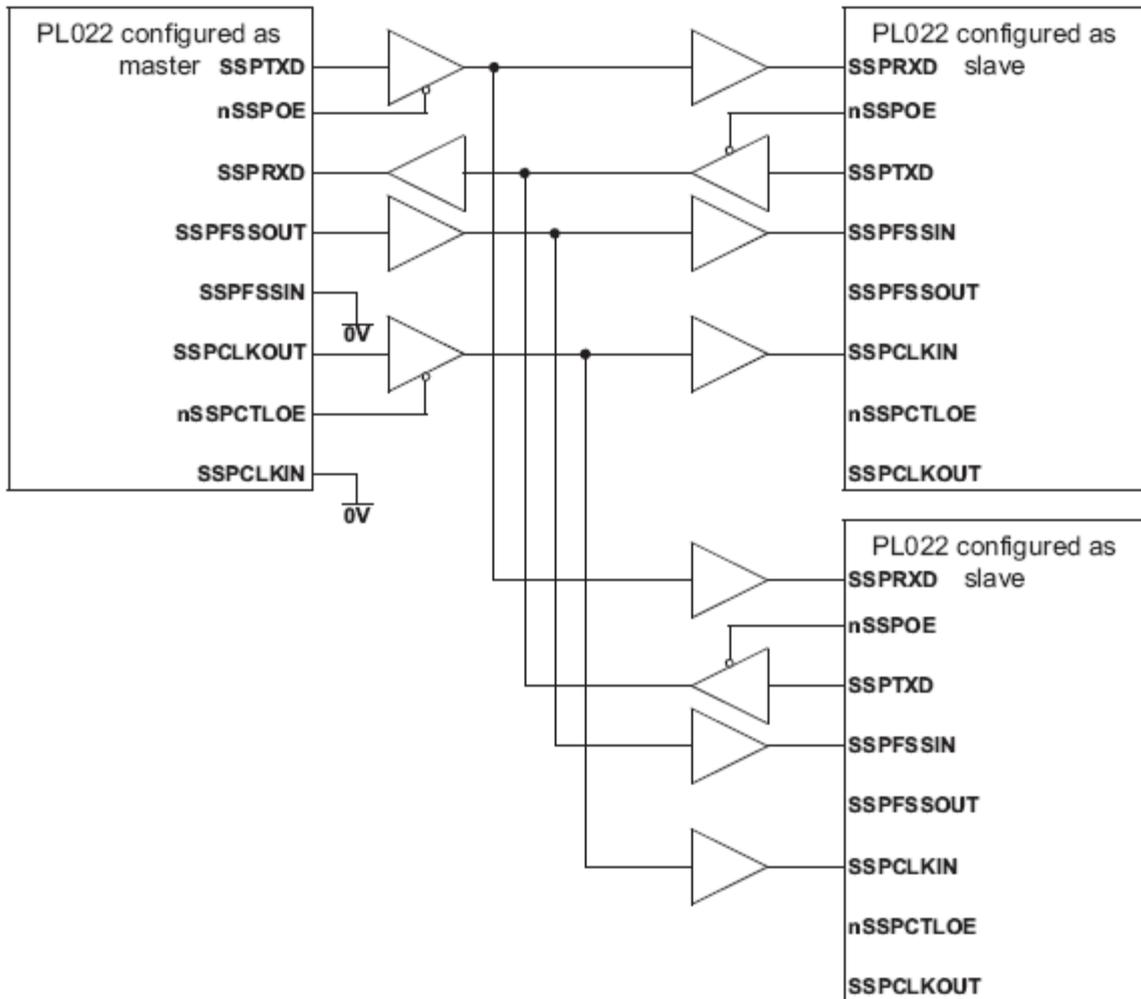


Рисунок 145 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 145 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные по кругу в адрес двух ведомых по линии SSP_TXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSP_TXD на вход SSP_RXD ведущего.

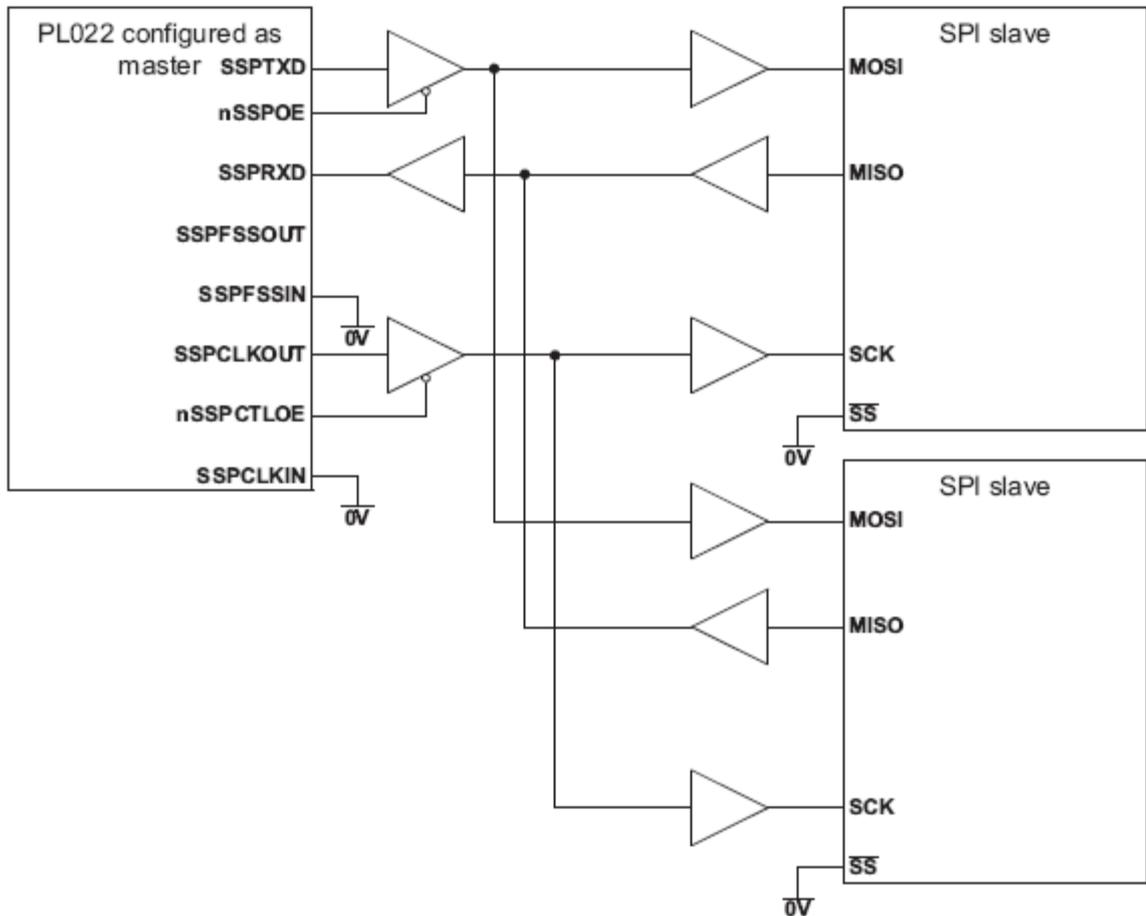


Рисунок 146 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 146 показывает подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые, путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых по кругу по линии SSP_TXD. Ответная передача данных на входную линию SSP_RXD ведущего устройства, одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

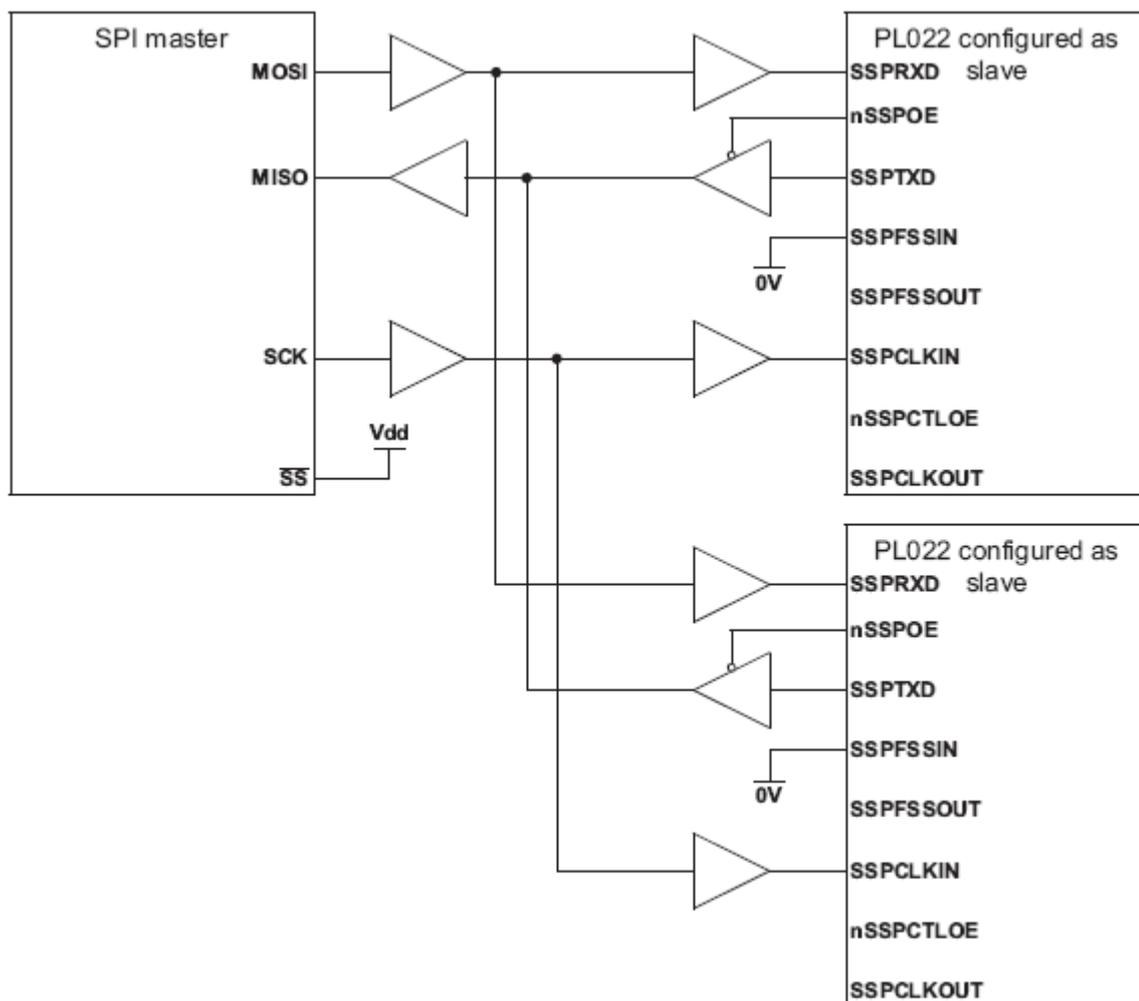


Рисунок 147 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 147 показывает ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI по кругу в адрес двух ведомых модулей.

Для ответной передачи данных, один из ведомых модулей переводит линию SSP_TXD в активное состояние, разрешая, таким образом, прохождение сигнала от своей линии SSP_TXD на вход SSP_RXD ведущего.

7.19.3 Интерфейс прямого доступа к памяти

Модуль SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA - DMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

- SSPRXDMASREQ – запрос передачи отдельного символа, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ;
- SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более символов;
- SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

- SSPTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит по меньшей мере одну свободную ячейку;
- SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее символов;
- SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимоисключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, формируются как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае, если количество данных в буфере приема меньше порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание – Для оставшихся трех символов контроллер SSP не инициирует процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

В таблице приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 109 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/2	4	4

Рисунок 148 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

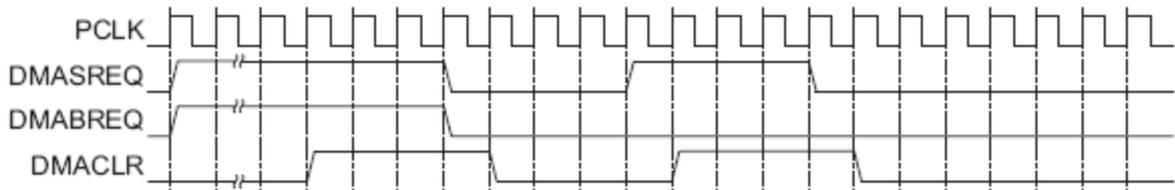


Рисунок 148 – Временные диаграммы обмена в режиме DMA

7.19.4 Прерывания

В модуле предусмотрено четыре маскируемые линии запроса на прерывание с выводом на один общий сигнал, представляющий собой их объединение по схеме ИЛИ.

Сигналы запроса на прерывание:

- SSPRXINTR – запрос на обслуживание буфера FIFO приемника;
- SSPTXINTR – запрос на обслуживание буфера FIFO передатчика;
- SSPRORINTR – переполнение буфера FIFO приемника;
- SSPRTINTR – таймаут приемника.

Сигнал SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка бита в 1 разрешает соответствующее прерывание, а в 0 – запрещает. Общий сигнал запроса прерывания SSPINTR маскируется средствами NVIC ядра контроллера.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPRIS, либо из маскированного регистра прерываний SSPMIS.

7.19.4.1 SSPRXINTR

Прерывание по заполнению буфера FIFO приемника. Формируется в случае, если буфер приемника содержит четыре или более несчитанных слов данных.

7.19.4.2 SSPTXINTR

Прерывание по заполнению буфера FIFO передатчика. Формируется в случае, если буфер передатчика содержит четыре или менее корректных слов данных.

Состояние прерывания не зависит от значения сигнала разрешения работы модуля SSP. Это позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов. Во-первых, можно записать данные в буфер заблаговременно, перед активизацией передатчика и разрешения прерываний. Во-вторых, можно предварительно разрешить работу модуля и формирование прерываний и заполнять буфер передатчика в ходе работы процедуры обслуживания прерываний.

7.19.4.3 SSPRORINTR

Прерывание по переполнению буфера FIFO приемника формируется в случае, если буфер уже заполнен и блоком приемника осуществлена попытка записать в него еще одно слово. При этом принятое слово данных регистрируется в регистре сдвига приемника, но в буфер приемника не заносится.

7.19.4.4 SSPRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.

Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения, либо после приема новых слов данных по входной линии SSP_RXD. Кроме того, оно может быть снято путем записи 1 в бит RTIC регистра сброса прерывания SSPTICR.

7.19.4.5 SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR с учетом маскирования.

7.19.5 Программное управление модулем

7.19.5.1 Общая информация

В микроконтроллере реализован модуль SSP. Базовый адрес регистров управления модулем указан в таблице . Смещение каждого регистра относительно базового адреса постоянно. Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;
- адреса со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

7.19.6 Описание регистров

Данные о регистрах модуля SSP приведены в таблице .

Таблица 110 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4009_5000	MDR_SSP0				Контроллер SSP0
Смещение					
0x000	CR0	RW	0x0000	16	Регистр управления 0
0x004	CR1	RW	0x0	4	Регистр управления 1
0x008	DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	5	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x0	4	Регистр маски прерывания
0x018	RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования
0x01C	MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	2	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти

Примечание – В поле «тип» указан вид доступа к регистру: RW – чтение и запись, RO – только чтение, WO – только запись.

7.19.6.1 CR0

Регистр управления 0

Регистр CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице ниже (Таблица 111).

Таблица 111 – Формат регистра CR0

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению: $F_{SSPCLK} / (CPSDVSR * (1 + SCR))$, где CPSDVSR – четное число в диапазоне от 2 до 254 (см. регистр SSPCPSR), а SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат синхронного обмена SPI фирмы Motorola»
5...4	FRF	Формат информационного кадра. 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв
3...0	DSS	Размер слова данных: 0000 – резерв 0001 – резерв 0010 – резерв 0011 – 4 бита 0100 – 5 бит 0101 – 6 бит 0110 – 7 бит 0111 – 8 бит 1000 – 9 бит 1001 – 10 бит 1010 – 11 бит 1011 – 12 бит 1100 – 13 бит 1101 – 14 бит 1110 – 15 бит 1111 – 16 бит

7.19.6.2 CR1

Регистр управления 1

Регистр CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице ниже (Таблица 112).

Таблица 112 – Регистр CR1

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...4		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешено. 1 – управление линией SSP_TXD в ведомом режиме запрещено
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника

7.19.6.3 DR

Регистр данных

Регистр SSPDR имеет разрядность 16 бит и предназначен для чтения принятых, и записи передаваемых данных.

Операция чтения обеспечивает доступ к последней несчитанной ячейке буфера FIFO приемника. Запись данных в этот буфер FIFO осуществляет блок приемника.

Операция записи позволяет занести очередное слово в буфер FIFO передатчика. Извлечение данных из этого буфера осуществляет блок передатчика. При этом извлеченные данные помещаются в регистр сдвига передатчика, откуда последовательно выдаются на линию SSP_TXD с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSPDR описано в таблице ниже (Таблица 113).

Таблица 113– Формат регистра DR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись). В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника

7.19.6.4 SR

Регистр состояния

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, а также занятости модуля SSP.

Назначение бит регистра SSPSR представлено в таблице ниже (Таблица 114).

Таблица 114– Регистр SR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...5		Резерв, при чтении результат не определен
4	BSY	Флаг активности модуля: 0 – модуль SSP не активен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст

7.19.6.5 CPSR

Регистр делителя тактовой частоты

Регистр SSPCPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Таблица 115 отображает назначение бит регистра SSPCPSR.

Таблица 115 – Регистр CPSR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Резерв. При чтении результат не определен. При записи следует заполнить нулями
7... 0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль

7.19.6.6 IMSC

Регистр установки и сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение бит регистра IMSC показано в таблице ниже (Таблица 116).

Таблица 116– Регистр IMSC

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO передатчика. 1 – не маскирована. 0 – маскирована
2	RXIM	Маска прерывания по заполнению на 50% и более буфера FIFO приемника. 1 – не маскирована. 0 – маскирована
1	RTIM	Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попыток его чтения в течение времени таймаута). 1 – не маскирована. 0 – маскирована
0	RORIM	Маска прерывания по переполнению буфера приемника. 1 – не маскирована. 0 – маскирована

7.19.6.7 RIS

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Таблица 117 отображает назначение бит в регистре RIS.

Таблица 117 – Регистр RIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31... 4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR

7.19.6.8 MIS

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPMIS представлено в таблице ниже (Таблица 118).

Таблица 118 – Регистр MIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

7.19.6.9 ICR

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре SSPICR представлено в таблице ниже (Таблица 119).

Таблица 119 – Регистр ICR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31... 2		Резерв. Не модифицируйте. При чтении выдаются нули
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR

7.19.6.10 DMACR

Регистр управления прямым доступом к памяти

Регистр доступен по чтению и записи. После сброса, все биты регистра обнуляются. Назначение бит регистра DMACR представлено в таблице ниже (Таблица 120).

Таблица 120 – Регистр DMACR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

7.20 Контроллер UART (UARTx)

Модуль универсального асинхронного приемопередатчика (UART – Universal Asynchronous Receiver Transmitter) является периферийным устройством микроконтроллера.

7.20.1 Основные сведения

Основные сведения о модуле представлены в следующих разделах:

- основные характеристики;
- программируемые параметры;
- отличия от приемопередатчика 16C650.

7.20.2 Основные характеристики модуля UART

- Содержит независимые буферы приема (16x12) и передачи (16x8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.
- Программное отключение FIFO позволяет ограничить размер буфера одним байтом.
- Программное управление скоростью обмена. Обеспечивается возможность деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.
- Поддержка стандартных элементов асинхронного протокола связи – стартового и стопового бит, а также бита контроля четности, которые добавляются перед передачей и удаляются после приема.
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.
- Поддержка прямого доступа к памяти.
- Обнаружение ложных стартовых бит.
- Формирование и обнаружения сигнала разрыва линии.
- Полностью программируемый асинхронный последовательный интерфейс с характеристиками:
 - данные длиной 5, 6, 7 или 8 бит;
 - формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
 - формирование 1 или 2 стоповых бит;
 - скорость передачи данных – от 0 до UARTCLK/16 Бод.
- Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

7.20.3 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная часть числа;
- количество бит данных;
- количество стоповых бит;
- режим контроля четности;
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);
- порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);
- частота внутреннего тактового генератора (номинальное значение – 1,8432 МГц) может быть задана в диапазоне 1,42 – 2,12 МГц для обеспечения возможности

- формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
- режим аппаратного управления потоком данных.

7.20.3.1 Отличия от контроллера UART 16C650

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика 16C650 следующими характеристиками:

- пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, 1/4, 1/2, 3/4 и 7/8;
- пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, 1/4, 1/2, 3/4 и 7/8;
- отличается распределение адресов внутренних регистров и назначение бит в регистрах;
- недоступны изменения сигналов состояния модема.

Следующие возможности контроллера 16C650 не поддерживаются:

- полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита);
- независимое задание тактовой частоты приемника и передатчика.

7.20.3.2 Функциональные возможности

Устройство выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму.

Процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии модуля. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 байтов, независимо для режимов приема и передачи.

Модуль приемопередатчика:

- содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UARTCLK;
- обеспечивает возможности, сходные с возможностями промышленного стандарта – контроллера UART 16C650.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR_H и регистрами делителя скорости передачи данных – целой части (UARTIBRD) и дробной части (UARTFBRD).

Устройство может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;
- общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;
- сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере FIFO. В случае

переполнения буфера немедленно устанавливается соответствующий бит в регистре переполнения, а доступ к записи в буфер FIFO блокируется.

Существует возможность программно ограничить размер буфера FIFO одним байтом, что позволяет реализовать общепринятый интерфейс асинхронной последовательной связи с двойной буферизацией.

7.20.4 Описание функционирования блока UART

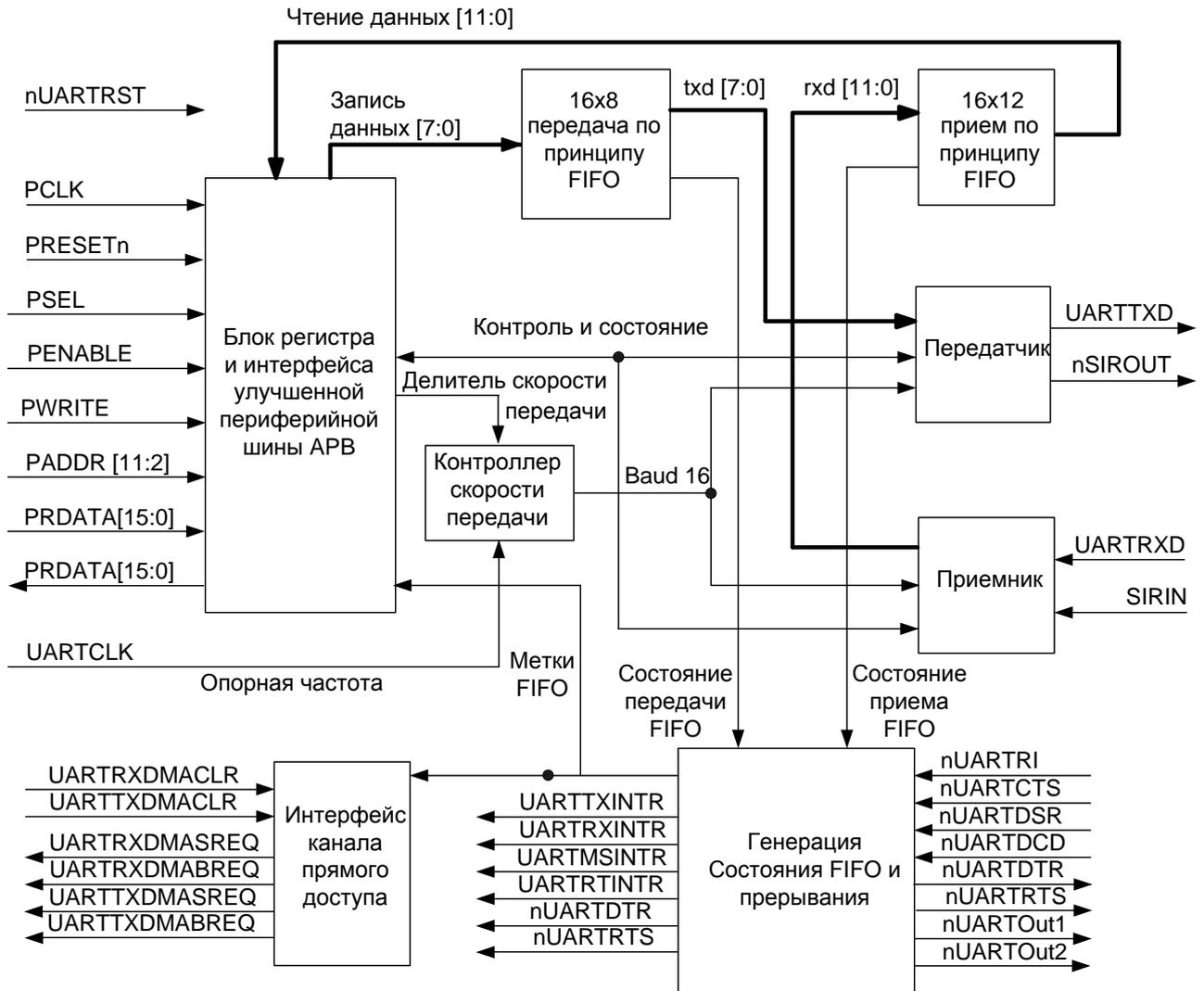


Рисунок 149 – Блок-схема универсального асинхронного приёмопередатчика (UART)

7.20.4.1 Генератор тактового сигнала приемопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK и частотой, в 16 раз выше скорости передачи данных.

7.20.4.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 8 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны логической схемой

передачи данных. Существует возможность запретить буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

7.20.4.3 Буфер FIFO приемника

Буфер приемника имеет ширину 12 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие коды ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

7.20.4.4 Блок передатчика

Логические схемы передатчика осуществляют преобразование данных, считанных из буфера передатчика, из параллельной в последовательную форму. Управляющая логика выдает последовательный поток бит в порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит проверки на четность, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления.

7.20.4.5 Блок приемника

Логические схемы приемника преобразуют данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того, производятся проверки переполнения буфера, проверки на ошибки контроля четности, на ошибки в структуре сигнала, а также на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

7.20.4.6 Блок формирования прерываний

Контроллер генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Другой подход состоит в подаче на системный контроллер прерываний независимых линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. подраздел «Прерывания».

7.20.5 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

7.20.6 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU_CLK и UARTCLK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU_CLK в область действия UARTCLK, так и наоборот.

7.20.7 Описание работы UART

7.20.7.1 Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в подразделе «Программное управление модулем».

7.20.7.2 Тактовые сигналы

Частота тактового сигнала UARTCLK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F_{UARTCLK}(min) \geq 16 \times baud_rate_{max};$$

$$F_{UARTCLK}(max) \leq 16 \times 65535 \times baud_rate_{min}.$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 МГц до 115,34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU_CLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты CPU_CLK.

$$F_{UARTCLK} \leq \frac{5}{3} \times F_{CPU_CLK}.$$

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UARTCLK 14,7456 МГц, частота CPU_CLK должна быть не менее 8,85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO.

7.20.7.3 Работа универсального асинхронного приемопередатчика

Управляющая информация хранится в регистре управления линией UARTLCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

- UARTLCR_H – определяет:
 - параметры передачи данных;
 - длину слова;
 - режим буферизации;
 - количество передаваемых стоповых бит;
 - режим контроля четности;
 - формирование сигнала разрыва линии;
- UARTIBRD – определяет целую часть коэффициента деления для скорости передачи данных;

- UARTFBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

7.20.7.4 Коэффициент деления частоты

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом 16 бит выделено для представления его целой части, а 6 бит – дробной части. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными со стандартными информационными скоростями, при этом используя в качестве UARTCLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр UARTIBRD. Шестиразрядная дробная часть записывается в регистр UARTFBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом:

$$\text{Коэффициент деления} = \frac{\text{UARTCLK}}{(16 \times \text{скорость передачи данных})} = \text{BRD_I} + \text{BRD_F},$$

где BRD_I – целая часть, а BRD_F – дробная часть коэффициента деления.



Рисунок 150 – Коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2^n , где n – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$$M = \lfloor \text{BRD_F} \times 2^n + 0.5 \rfloor,$$

где $\lfloor \ \rfloor$ – операция отсечения дробной части числа, $n = 6$.

В модуле формируется внутренний сигнал Vaud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UARTCLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

7.20.7.5 Передача и прием данных

Принятые или передаваемые данные заносятся в 16-элементные буферы FIFO, при этом каждый элемент приемного буфера FIFO, кроме байта данных, хранит также четыре бита информации о состоянии модема.

Данные для передачи заносятся в буфер FIFO передатчика. Если работа приемопередатчика разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией UARTLCR_H. Передача данных продолжается до опустошения буфера FIFO передатчика. После записи элемента в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние. Это состояние сохраняется в течение всего времени передачи данных. Сигнал BUSY переходит в низкое состояние только после того, как буфер FIFO передатчика станет пуст, а последний бит данных (включая стоповые биты) будет передан. Сигнал BUSY может находиться в высоком состоянии даже в случае, если приемопередатчик будет переведен из разрешенного состояния в запрещенное.

Для каждого бита данных (в приемной линии) производится три измерения уровня, решение принимается по мажоритарному принципу.

В случае, если приемник находился в неактивном состоянии (на линии входного сигнала UART_RXD постоянно присутствовала единица), и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), то включается счетчик, тактируемый сигналом Vaud16. После чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Vaud16. Более частая выборка данных в режиме ИК

обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UART_RXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае, если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (что соответствует длительности одного символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART_RXD). В случае, если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым словом (Таблица 121).

7.20.7.6 Биты ошибки

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся в разряды [10...8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO в разряде 11 слова данных.

Таблица 121 описывает назначение всех бит слова данных в FIFO-буфере приемника.

Таблица 121 – Назначение бит слова данных в FIFO-буфере приемника

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – «разрыв линии»
9	Ошибка проверки на четность
8	Ошибка формирования кадра
7...0	Принятые данные

7.20.7.7 Бит переполнения буфера

Бит переполнения непосредственно не связан с конкретным символом в буфере приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен к моменту, когда очередной символ данных полностью принят (находится в регистре сдвига). При этом данные из регистра сдвига не попадают в буфер приемника и теряются с началом приема очередного символа. Как только в буфере приемника появляется свободное место, очередной принятый символ данных заносится в буфер FIFO вместе с текущим значением признака переполнения. После успешной записи данных в буфер признак переполнения сбрасывается.

7.20.7.8 Запрет буфера FIFO

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая сторона контроллера UART располагают лишь однобайтными буферными регистрами. Бит переполнения буфера устанавливается при этом тогда, когда очередной символ данных уже принят, однако предыдущий еще не был считан.

В настоящей реализации модуля буферы FIFO физически не отключаются, необходимая функциональность достигается за счет логических манипуляций с флагами. При этом в случае, если буфер FIFO отключен, а сдвиговый регистр передатчика пуст (не используется), запись байта данных происходит непосредственно в регистр сдвига, минуя буферный регистр.

7.20.7.9 Проверка по шлейфу

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в 1 бита LBE в регистре управления контроллером UARTCR.

7.20.8 Интерфейс прямого доступа к памяти

Модуль универсального асинхронного приемопередатчика оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA UARTDMACR.

Интерфейс DMA включает в себя следующие сигналы:

– **Для приема:**

- UARTRXDMASREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, один символ.
- UARTRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO, путем записи значения в регистр UARTIFLS.
- UARTRXDMACLR – сброс запроса на DMA, инициируется модулем приемопередатчика с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

– **Для передачи:**

- UARTTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.
- UARTTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если заполнение буфера FIFO передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UARTIFLS.
- UARTTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимоисключаемыми, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае, если количество данных в буфере приема меньше порогового значения формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов, а порог заполнения буфера FIFO установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание – Для оставшихся трех символов контроллер UART не может инициировать процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления DMA TXDMAE или RXDMAE в регистре управления DMA UARTDMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы; как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления DMA UARTRXDMASREQ и UARTTXDMASREQ. Для информации о запрете буферов FIFO см. описание регистра управления линией UARTLCR_H.

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. Таблица 122 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 122 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
1/4	12	4
1/2	8	8
3/4	4	12
7/8	2	14

В регистре управления DMA UARTDMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки UARTEINTR. При этом соответствующие линии запроса DMA – UARTRXDMASREQ и UARTRXDMABREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA, обслуживающие передатчик, состояние UARTEINTR не влияет.

На рисунке ниже (Рисунок 151) показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с CPU_CLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.

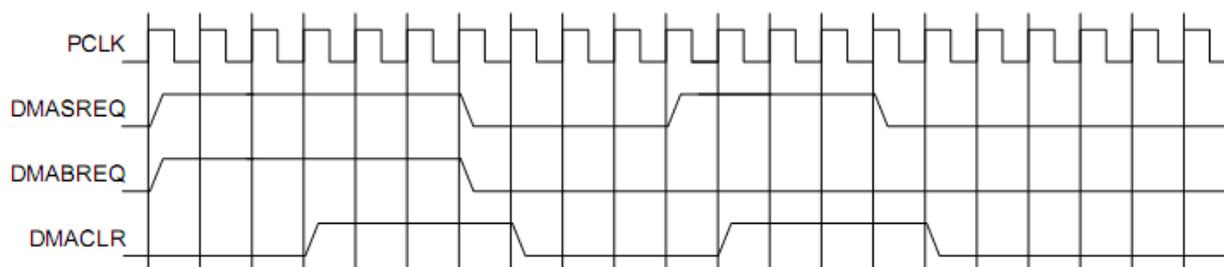


Рисунок 151 – Временные диаграммы одноэлементного и блочного запросов DMA

7.20.9 Прерывания

В модуле предусмотрено 7 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

UARTRXINTR – прерывание от приемника.

UARTTXINTR – прерывание от передатчика.

UARTRTINTR – прерывание по таймауту приемника.

UARTEINTR – ошибка:

- UARTOEINTR, переполнение буфера;
- UARTBEINTR, прерывание приема – разрыв линии;
- UARTPEINTR, ошибка контроля четности;
- UARTFEINTR, ошибка в структуре кадра.

UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибки UARTEINTR формируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака ошибки.

Прерывание по состоянию модема представляет собой комбинацию признаков изменения отдельных линий состояния модема.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTRMIS.

7.20.9.1 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

7.20.9.2 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

Для занесения данных в буфер FIFO передатчика необходимо записать данные в буфер либо перед разрешением работы приемопередатчика и прерываний, либо после разрешения работы приемопередатчика и прерываний.

Примечание – Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если модуль и прерывания от него разрешены, до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

7.20.9.3 UARTRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

7.20.9.4 UARTEINTR

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;
- ошибка контроля четности;
- разрыв линии;
- переполнение буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний UARTRIS, либо содержимое маскированного регистра прерываний UARTRMIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания UARTICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

7.20.9.5 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTRTXINTR, UARTRTINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

7.20.10 Программное управление модулем

7.20.10.1 Общая информация

Следующая информация применима ко всем регистрам контроллера:

- Базовый адрес контроллера не фиксирован и может быть различным в разных системах. Смещение каждого регистра относительно базового адреса постоянно.
- Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.
- За исключением специально оговоренных в настоящем документе случаев:
 - не следует изменять значения не определенных в документе разрядов регистров;
 - не следует использовать значения не определенных в документе разрядов регистров;
 - все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение 0 после сброса по включению питания или системного сброса.
- Столбец «Тип» (Таблица 123) определяет режим доступа к регистру в соответствии с обозначениями:
 - RW – чтение и запись;
 - RO – только чтение;
 - WO – только запись.

7.20.11 Описание регистров

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице ниже (Таблица 123).

Таблица 123 – Обобщенные данные о регистрах устройства

Адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4009_9000	MDR_UART1				Контроллер UART1
0x4009_A000	MDR_UART2				Контроллер UART2
Смещение					
0x000	DR	RW	0x----	12/8	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистр состояния приемника / Сброс ошибки приемника
0x008-0x014					Зарезервировано
0x018	FR	RO	0b-10010----	9	Регистр флагов
0x01C					Зарезервировано
0x020					Зарезервировано
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	8	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	11	Регистр маски прерывания
0x03C	RIS	RO	0x00-	11	Регистр состояния прерываний
0x040	MIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления DMA
0x04C-FDC					Зарезервировано
0xFE0	PerID0	RO	0x11	8	Идентификационный регистр 0
0xFE4	PerID1	RO	0x10	8	Идентификационный регистр 1
0xFE8	PerID2	RO	0x14	8	Идентификационный регистр 2
0xFEC	PerID3	RO	0x00	8	Идентификационный регистр 3
0xFF0	PerID4	RO	0x0D	8	Идентификационный регистр 4
0xFF4	PerID5	RO	0xF0	8	Идентификационный регистр 5
0xFF8	PerID6	RO	0x05	8	Идентификационный регистр 6
0xFFC	PerID7	RO	0xB1	8	Идентификационный регистр 7

7.20.11.1 DR

Регистр данных

В ходе передачи данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика.

В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).

Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

В ходе приема данных:

Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере.

В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считывается путем чтения из регистра UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра UARTRSR/UARTECR (Таблица 124).

Таблица 124 – Формат регистра UARTDR

№ бита	Сигнал	Назначение
15...12		Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место
10	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
9	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTLCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
8	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
7...0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись)

Примечание – Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приёмопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

7.20.11.2 RSR_ECR

Регистр состояния приемника / сброса ошибки

Состояние приемника также может быть считано из регистра UARTRSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных UARTDR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним, считанным из регистра UARTDR, байтом данных).

Запись в регистр UARTECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.

Таблица 125 показывает назначение бит регистра UARTRSR/UARTECR.

Таблица 125 – Регистр UARTRSR/UARTECR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...4		Резерв, при чтении результат не определен
3	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после записи в регистр UARTECR. Содержимое буфера остается верным, так как был перезаписан только регистр сдвига. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO
2	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящемся на вершине буфера. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
1	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARLPCR_H (стр. 3-12). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера
0	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера

Примечания:

- 1 Перед чтением регистра состояния UARTRSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTRSR обновляет свое состояние только после чтения регистра UARTDR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR.
- 2 Запись в регистр UARTRSR/UARTECR любого кода сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера

7.20.11.3 FR

Регистр флагов

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице ниже (Таблица 126) представлена информация о назначении бит регистра UARTFR.

Таблица 126 – Регистр UARTFR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...9		Резерв. Не модифицируйте. При чтении заполняются нулями
8	-	Зарезервировано
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит равен 1, когда буферный регистр передатчика занят. В противном случае он равен 1, если заполнен буфер FIFO передатчика
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника пуст. В противном случае он равен 1, если пуст буфер FIFO приемника
3	BUSY	UART занят. Бит равен 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
2	-	Зарезервировано
1	-	Зарезервировано
0	-	Зарезервировано

7.20.11.4 IBRD

Регистр целой части делителя скорости передачи данных

Назначение бит регистра UARTIBRD показано ниже.

Таблица 127 – Регистр UARTIBRD

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

7.20.11.5 FBRD

Регистр дробной части делителя скорости передачи данных

Таблица 128 показывает назначение бит регистра.

Таблица 128 – Регистр UARTFBRD

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5...0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

Коэффициент деления вычисляется по формуле:

$$BAUDDIV = F_{UARTCLK} / 16 \times Baud_rate'$$

где $F_{UARTCLK}$ – тактовая частота контроллера UART, $Baud_rate$ – требуемая скорость передачи данных.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV_INT и BAUDDIV_FRAC, соответственно.

Примечания:

- 1 Изменения содержимого регистров UARTIBRD и UARTFBRD вступают в силу только после завершения передачи и приема текущего символа данных.
- 2 Минимальный допустимый коэффициент деления – 1, максимальный 65535 ($2^{16} - 1$). Таким образом, значение UARTIBRD, равное 0, является недопустимым, при этом значение регистра UARTFBRD игнорируется.
- 3 Аналогично, при UARTIBRD равном 65535 (0xFFFF), значение UARTFBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

Пример. Вычисление коэффициента деления

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала UARTCLK равна 4 МГц. Тогда:

$$\text{Коэффициент деления} = (4 * 10^6) / (16 * 230400) = 1,085.$$

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр UARTBFRD, равно

$$m = integer((0,085 \times 64) + 0,5) = 5.$$

Реальное значение коэффициента деления равно

$$1 + 5/64 = 1,078.$$

Реальная скорость передачи данных равна

$$(4 \times 10^6)/(16 \times 1.078) = 231911 \text{ бит/с.}$$

Ошибка установки скорости равна

$$\frac{231911-230400}{230400} \times 100\% = 0,656\%.$$

Максимальная ошибка установки скорости передачи данных с использованием шестиразрядного регистра UARTBFRD = $1/64 \times 100\% = 1,56\%$. Такая ошибка возникает в случае $m = 1$, при этом разница накапливается в течение 64 тактовых интервалов.

В следующей таблице (Таблица 129) представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 7,3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр UARTFBRD должен быть записан ноль.

Таблица 129 – Коэффициенты деления при частоте UARTCLK = 7,3728 МГц

Коэффициент деления	Скорость передачи данных
0x0001	460800
0x0002	230400
0x0004	115200
0x0006	76800
0x0008	57600
0x000C	38400
0x0018	19200
0x0020	14400
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В таблице ниже приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 130 – Коэффициенты деления при частоте UARTCLK = 4 МГц

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0.656
0x002	0x0B	115200	115101	0.086
0x003	0x10	76800	76923	0.160
0x006	0x21	38400	38369	0.081
0x011	0x17	14400	14401	0.007
0x068	0x0B	2400	2400	~ 0
0x8E0	0x2F	110	110	~ 0

7.20.11.6 LCR_H

Регистр управления линией

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.

Таблица 131 показывает назначение разрядов регистра UARTLCR_H.

Таблица 131 – Регистр UARTLCR_H

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...8		Резерв. Не модифицируйте. При чтении выдаются нули.
7	SPS	Передача бита четности с фиксированным значением. 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещен
6...5	WLEN	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит 0b10 – 7 бит 0b01 – 6 бит 0b00 – 5 бит
4	FEN	Разрешение работы буфера FIFO приемника и передатчика. 0 – запрещено; 1 – разрешено
3	STP2	Режим передачи двух стоповых бит. 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
2	EPS	Четность/нечетность. 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного; 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещена
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе UARTTXD устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в 0

Содержимое регистров UARTLCR_H, UARTIBRD и UARTFBRD совместно образует общий 30-разрядный регистр UARTLCR, который обновляется по стробу, формируемому при записи в UARTLCR_H. Для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после изменения значения регистров UARTIBRD и/или UARTFBRD необходимо осуществить запись данных в регистр UARTLCR_H.

Примечания:

- 1 Изменение значений трех регистров можно осуществить корректно двумя способами:
 - запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H;
 - запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H.

- 2 Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD) необходимо выполнить следующий шаг:
- запись UARTIBRD (или UARTFBRD), запись UARTLCR_H.

Таблица 132 показывает таблицу истинности для бит управления контролем четности PEN, EPS и SPS регистра управления линией UARTLCR_H.

Таблица 132 – Управление режимом контроля четности

PEN	EPS	SPS	Бит контроля четности
0	X	X	Не передается, не проверяется
1	1	0	Проверка четности слова данных
1	0	0	Проверка нечетности слова данных
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечания:

- 1 Регистры UARTLCR_H, UARTIBRD и UARTFBRD не должны изменяться:
 - при разрешенной работе приемопередатчика;
 - во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.
- 2 Целостность данных в буферах FIFO не гарантируется в следующих случаях:
 - после установки бита разрыва линии BRK;
 - если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO после его повторного перевода в разрешенное состояние.

7.20.11.7 CR

Регистр управления

После сброса все биты регистра управления, за исключением бит 9 и 8 устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние. Назначение разрядов регистра управления показано в следующей таблице (Таблица 133).

Таблица 133 – Регистр управления UARTCR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...10	-	Зарезервировано
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. В случае перевода приемопередатчика в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой
7	LBE	0 – запрещено; 1 – шлейф разрешен. Выходная линия передатчика UARTTXD коммутируется на вход приемника UARTRXD. После сброса бит устанавливается в 0
6...3	-	Резерв. Не модифицируйте. При чтении выдаются нули
2, 1	-	Зарезервировано
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN

Примечание – Для того, чтобы разрешить передачу данных, биты TXE и UARTEN необходимо установить в логическую 1. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN.

Примечание – Рекомендуется следующая последовательность действий для программирования регистров управления:

- 1 остановите работу приемопередатчика;
- 2 дождитесь окончания приема и/или передачи текущего символа данных;
- 3 сбросьте буфер передатчика путем установки бита FEN регистра UARLCLR_H в 0;
- 4 измените настройки регистра UARTCR;
- 5 возобновите работу приемопередатчика.

7.20.11.8 IFLS

Регистр порога прерывания по заполнению буфера FIFO

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра UARTIFLS и значения его бит представлены в таблице .

Таблица 134 – Регистр UARTIFLS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
--------	-------------------------	--

31...6		Резерв. Не модифицируйте. При чтении выдаются нули.
5...3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = буфер заполнен на 1/8 b001 = буфер заполнен на 1/4 b010 = буфер заполнен на 1/2 b011 = буфер заполнен на 3/4 b100 = буфер заполнен на 7/8 b101-b111 = резерв
2...0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = буфер заполнен на 1/8 b001 = буфер заполнен на 1/4 b010 = буфер заполнен на 1/2 b011 = буфер заполнен на 3/4 b100 = буфер заполнен на 7/8 b101-b111 = резерв Примечание – если сдвиговый регистр передатчика пуст, слово, записанное в FIFO, будет сразу же переписано в сдвиговый регистр. Следовательно, для генерирования события прерывания от передатчика блока UART необходимо произвести запись в FIFO такого количества слов, которое превысит установленный порог хотя бы на одно слово с учетом описанного случая.

7.20.11.9 IMSC

Регистр установки сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние. Назначение бит регистра UARTIMSC показано в таблице ниже (Таблица 135).

Таблица 135 – Регистр UARTIMSC

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR: 1 – установлена; 0 – сброшена
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR: 1 – установлена; 0 – сброшена
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR: 1 – установлена; 0 – сброшена
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR: 1 – установлена; 0 – сброшена
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR: 1 – установлена; 0 – сброшена
5	TXIM	Маска прерывания от передатчика UARTTXINTR. 1 – установлена; 0 – сброшена
4	RXIM	Маска прерывания от приемника UARTRXINTR. 1 – установлена; 0 – сброшена
3...0	-	Зарезервировано

7.20.11.10 RIS

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит регистра UARTRIS представлено в следующей таблице (Таблица 136).

Таблица 136 – Регистр UARTRIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR. <i>Примечание</i> – Бит RTRIS может быть установлен только при установленной маске прерывания по таймауту приема данных UARTRTINTR в регистре UARTIMSC. Чтение состояния прерывания по таймауту из регистров UARTRIS и UARTRIS приводит к одинаковым результатам
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR
4	RXRIS	Состояние прерывания от приемника UARTRXINTR
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCCDRMIS	Состояние прерывания UARTDCCDINTR по изменению линии nUARTDCD
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI

7.20.11.11 MIS

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит регистра UARTMIS представлено в таблице ниже (Таблица 137).

Таблица 137 – Регистр UARTMIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR
3...0	-	Зарезервировано

7.20.11.12 ICR

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись нуля в любой из разрядов регистра игнорируется.

Назначение бит регистра UARTICR представлено в таблице ниже (Таблица 138).

Таблица 138 – Регистр UARTICR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR
5	TXIC	Сброс прерывания от передатчика UARTTXINTR
4	RXIC	Сброс прерывания от приемника UARTRXINTR
3...0	-	Зарезервировано

7.20.11.13 DMACR

Регистр управления прямым доступом к памяти

Регистр доступен по чтению и записи. После сброса, все биты регистра обнуляются. Назначение бит регистра UARTDMACR представлено в таблице ниже (Таблица 139).

Таблица 139 – Регистр UARTDMACR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3		Зарезервировано. Не модифицируйте. При чтении выдаются нули
2	DMAONERR	Если бит установлен в 1, то в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMSREQ и UARTRXDMSBREQ
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, то разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, то разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

7.21 Контроллер интерфейса ARINC (ARINCx)

Контроллер интерфейса ARINC состоит из 8 приемников и 4 передатчика по ГОСТ 18977-79 (далее ARINC). Каждый приёмник поддерживает функцию распознавания меток (или адресов). Для каждого приёмника может быть запрограммировано до 16 (32 с ревизии 3) 8-разрядных меток. Помимо этого фильтрация входных данных может осуществляться не только на базе меток, но и на базе двух бит Источник/Приёмник. Каждый передатчик поддерживает однонаправленную передачу 32-х разрядных слов по двухпроводной витой паре, используя форму т кодирования RZ. Доступна возможность запрограммировать 32-й бит либо как данные, либо как бит паритета. В случае формирования бита паритета, программируется его чётность или нечётность. Каждый приёмник и передатчик использует собственный буфер FIFO для хранения данных. Размеры буфера FIFO варьируются от 32x32 до 256x32. Статус наполненности FIFO определяется на основе соответствующих бит статуса для каждого FIFO. Контроллер поддерживает различные скорости приёма и передачи данных. Работа контроллера осуществляется на базовой частоте 1 МГц, что позволяет обнаруживать ошибки в скорости приёма/передачи данных, а также в паузах между сообщениями.

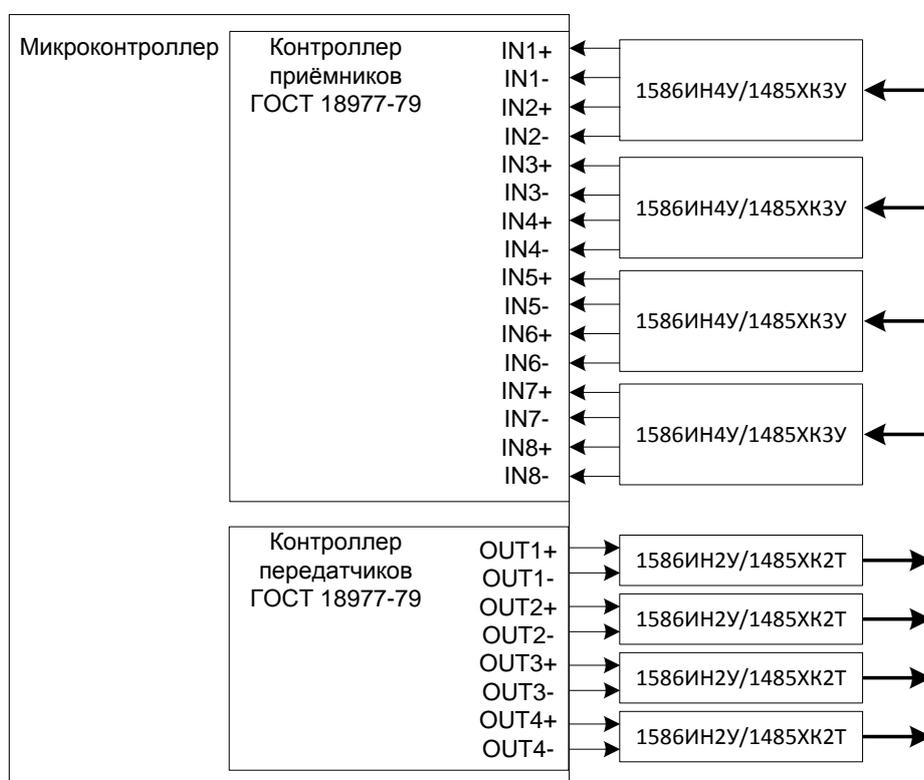


Рисунок 152 – Структурная схема контроллера интерфейса ARINC

Особенности:

- симплексный режим приёма/передачи со скоростями 12,5 кГц или 100 кГц;
- фильтрация входных данных на базе меток 16x8 (32x8 с ревизии 3) и двух бит Источник/Приёмник для каждого приёмника;
- возможность передачи 32 бита, как данных, так и паритета;
- выбор чётности/нечётности бита паритета;
- размеры буферов FIFO передатчиков: одно 256x32, три 64x32;
- размеры буферов FIFO приёмников: два 256x32, четыре 64x32, два 32x32;
- возможность формирования прерываний при разных статусах наполненности буферов FIFO и при возникновении ошибок скорости передачи слова и паузы между словами.

7.21.1 Формат слова

Слова в интерфейсе ARINC всегда 32-х разрядные, и включают в себя 5 полей: паритет, SSM, данные, источник/приёмник, метка. Биты передаются младшими разрядами вперёд, за исключением метки, которая передаётся старшими разрядами вперёд. В результате можно описать порядок следования бит по шине ARINC следующим образом:

8,7,6,5,4,3,2,1,9,10,11,12,13...32.

32	31	30	29	11	10	9	8	1
P	SSM		DATA MSB	LSB	SDI		LABEL	

Старший разряд всегда бит паритета. Стандартом установлено, что бит паритета должен дополнять слово до нечетного. Таким образом, количество единиц в 32-х разрядном слове должно быть нечётным. Например, если биты 1 ÷ 31 содержат чётное количество единиц, то бит паритета должен быть установлен в единицу, с другой стороны, если биты 1 ÷ 31 содержат нечётное количество единиц, то бит паритета должен быть сброшен в ноль.

Биты 31 и 30 содержат знак или статус. В контроллере эти биты рассматриваются как обычные данные и помещаются в FIFO вместе с полем данных без изменений и дополнительной обработки.

Как пример биты 31 и 30 могут кодировать следующие характеристики, представленные в таблице ниже (Таблица 140).

Таблица 140 – Значения битов 31 и 30

Бит		Значение
31	30	
0	0	плюс, север, восток, справа, к, выше
0	1	не вычислительные данные
1	0	функциональный тест
1	1	минус, юг, запад, слева, от, ниже

Биты 10 и 9 позволяют распознать Источник/Приёмник данных. Это применяется при нескольких приёмниках на шине ARINC, чтобы определить, для кого из них предназначаются данные. В системе со сложной структурой эти биты могут также использоваться, чтобы определить источник передачи. В остальных случаях эти разряды используются как данные. Следует отметить, что в интерфейсе ARINC на одной витой паре может быть один передатчик и до 20 приёмников. Если включена функция проверки этих бит, то при их несовпадении с битами, заданными программно в контроллере, сообщение не будет помещено в FIFO.

Биты с 1 по 8 позволяют идентифицировать тип данных оставшейся части слова, следовательно, методы преобразования, применяемые к данным. Помимо этого в контроллере метки используются для фильтрации входных данных, то есть если метка в принятом сообщении не соответствует ни одной из меток определённой в памяти меток приемного канала, то данные не помещаются в FIFO. Это может служить аналогом того, что приёмник не может интерпретировать метод обработки этих данных, следовательно, эти данные предназначены для другого приёмника.

В случае если приёмник принимает данные с неправильным битом паритета, они также не будут помещены в FIFO.

7.21.2 Структурная схема канала приёма

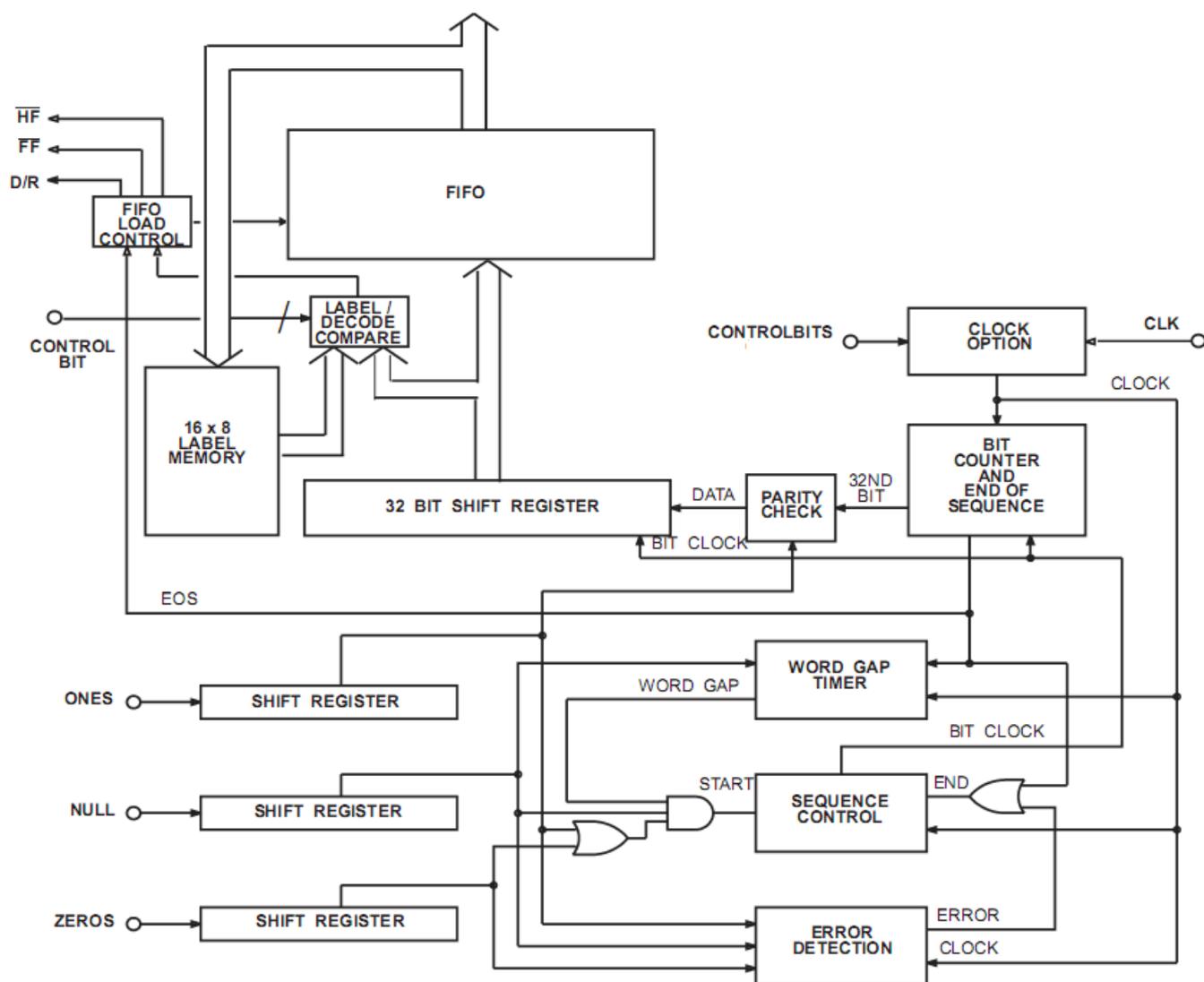


Рисунок 153 – Схема канала приёма ARINC

Представленная на рисунке схема (Рисунок 153) работает на частоте CLK = 1 МГц, в этом случае ошибка обнаружения бита в линии не будет составлять более 0,1 %.

Сдвиговые регистры длиной 10 бит, предназначенные для обнаружения в линии трёх последовательностей единиц (Ones), нулей (Zeros) и отсутствие сигнала (Null), позволяют считать данные действительными. В дополнении к этому, для бит данных – One или Zero в верхних битах сдвигового регистра, должны сопровождаться Null в нижних битах в пределах битового интервала. В пределах паузы между сообщениями, три последовательных бита Null должны быть сэмплированы в верхней и нижней части сдвигового регистра Null. В этом случае гарантируется минимальная ширина импульса данных.

Каждый бит данных должен быть обнаружен в пределах от 8 до 12 сэмплов. В этом случае скорость передачи считается верной.

Таймер паузы между сообщениями сэмплирует сдвиговый регистр Null каждые 10 входных тактов (или 80 тактов для скорости 12,5 кГц) после последнего полученного бита данных. Если Null обнаружен, то таймер инкрементируется. Значение таймера равное трём разрешает следующий приём.

Схема паритета считает количество принятых единиц, включая бит паритета. Если результат нечётный, то на выходе схемы формируется сигнал равный нулю.

После того как приняты все 32 бита логика приёмника формирует сигнал конец последовательности (EOS). В зависимости от состояния бит LB_EN, SD_EN, SDI1, SDI2 регистра управления принимается решение о загрузке принятых данных в FIFO. Если в

принятом слове биты 9 и 10 не соответствует правилам или не совпала метка, то слово не загружается в FIFO. Ниже приведены случаи, когда происходит загрузка FIFO принятыми данными (Таблица 141).

Таблица 141 – Загрузка FIFO принятыми данными

LB_EN	Результат сравнения слова ARINC с меткой	SD_EN	Результат сравнения бит 9,10 слова ARINC с SDI1, SDI2	FIFO
0	X	0	X	Загружается
1	не совпала	0	X	Игнорируются
1	совпала	0	X	Загружается
0	X	1	не совпали	Игнорируются
0	X	1	совпали	Загружается
1	совпала	1	не совпали	Игнорируются
1	не совпала	1	совпали	Игнорируются
1	не совпала	1	не совпали	Игнорируются
1	совпала	1	совпали	Загружается

Если хотя бы одно слово загружено в FIFO, то устанавливается в единицу сигнал DR, что отражается в регистре статуса контроллера. Флаг остаётся в неизменном состоянии, пока последнее слово не будет прочитано из FIFO, и оно не будет пустым. Помимо этого применяются ещё два сигнала характеризующие состояние FIFO, а именно HF означает, что FIFO наполовину полно, и FF означает, что FIFO полно. Установка этих сигналов также отражается в регистре статуса. Каждый из этих флагов может быть источником прерывания, в случае если оно разрешено соответствующим битом маскирования регистра управления.

7.21.3 Структурная схема канала передачи

Если флаг TX_R в состоянии логической единицы, это значит, что FIFO пусто, и в него могут быть загружены 31 или 32 битные данные. Количество слов данных определяется размером FIFO для выбранного канала передачи. Если флаг TX_R в состоянии логического нуля, тогда данные можно загрузить только в доступные в FIFO ячейки. Если FIFO заполнено полностью, флаг FFT установлен в единицу, то FIFO игнорирует дальнейшие попытки записи в него. FIFO наполовину полно, если установлен флаг HFT, в этом случае можно загрузить данными оставшуюся половину буфера FIFO.

В нормальном режиме работы 32-ой бит передаваемых данных является битом паритета. Чётность или нечётность выбирается битом ODD регистра управления. Если бит разрешения паритета (EN_PAR) сброшен в ноль, то тогда 32-ой бит передается, как бит данных из FIFO.

Если бит CH_EN установлен в единицу и FIFO передачи не пусто, то начинается передача слов данных из FIFO до тех пор, пока FIFO не будет пусто или не будет сброшен бит CH_EN.

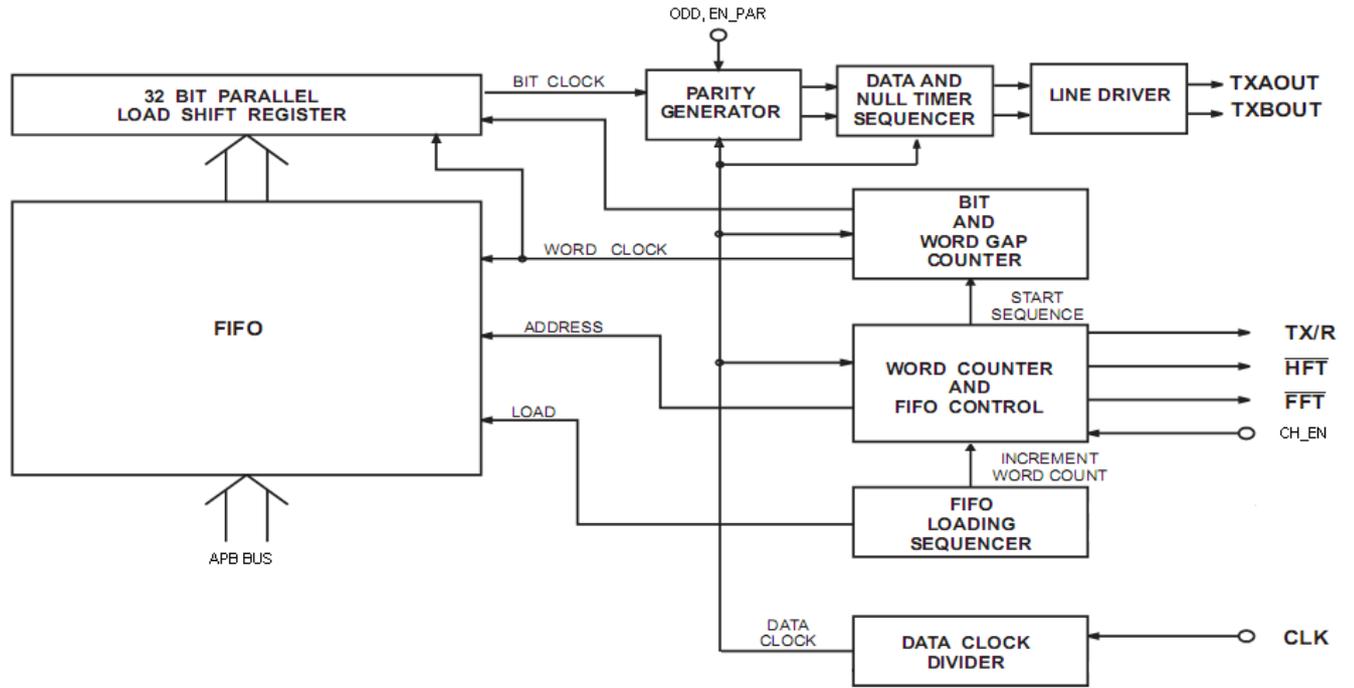


Рисунок 154 – Структурная схема канала передачи

7.21.4 Описание регистров приемника

Таблица 142 – Описание регистров приемника

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4009_E000		ARINC429R0		Контроллер интерфейса приёмников ARINC429
	0x0000_0000	CONTROL1		Регистр управления приёмников 1
	0x0000_0004	CONTROL2		Регистр управления приёмников 2
	0x0000_0008	CONTROL3		Регистр управления приёмников 3
	0x0000_000C	STATUS1		Регистр состояния приёмников 1
	0x0000_0010	STATUS2		Регистр состояния приёмников 2
	0x0000_0014	CONTROL4		Регистр настройки индивидуального делителя частоты каналов 1–4 с ревизии 2
	0x0000_0018	CONTROL5		Регистр настройки индивидуального делителя частоты каналов 5–8 с ревизии 2
	0x0000_001C	CHANNEL		Регистр номера канала приёмников
	0x0020	LABEL		FIFO меток
	0x0024	DATA_R		FIFO принимаемых данных
	0x0030	DATA_R1		FIFO принимаемых данных канала 1 при CHANNEL=14 (с ревизии 3)
	0x0034	DATA_R2		FIFO принимаемых данных канала 2 при CHANNEL=14 (с ревизии 3)
	0x0038	DATA_R3		FIFO принимаемых данных канала 3 при CHANNEL=14 (с ревизии 3)
	0x003C	DATA_R4		FIFO принимаемых данных канала 4 при CHANNEL=14 (с ревизии 3)
	0x0040	DATA_R5		FIFO принимаемых данных канала 5 при CHANNEL=14 (с ревизии 3)
	0x0044	DATA_R6		FIFO принимаемых данных канала 6 при CHANNEL=14 (с ревизии 3)
	0x0048	DATA_R7		FIFO принимаемых данных канала 7 при CHANNEL=14 (с ревизии 3)
	0x004C	DATA_R8		FIFO принимаемых данных канала 8 при CHANNEL=14 (с ревизии 3)
	0x0000_0068	INTMASK		Регистр индивидуальной настройки разрешения прерывания по заполненности FIFO каждого канала
	0x0000_0070	CONTROL8		Регистр управления приёмников 8
	0x0000_0074	CONTROL9		Регистр управления приёмников 9
	0x1000-0x2000	Память приёмников		При CHANNEL=0xF прямой доступ в память приёмников

7.21.4.2 CONTROL1

Регистр управления приёмников 1

Base ADDR=		0x4009_E000				Offset=		0x0000_0000									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
DIV[3:0]				-	-	-	-	-	-	CLK[7:2]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK[1:0]		-	-	-	-	-		CH_EN[7:0]							

Бит	Имя	Значение	Описание
31..28	DIV[3:0]		Делитель частоты ядра до 1 МГц Содержит младшие 4 разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц. Значение частоты не может быть более 125 МГц.
27..22			Зарезервировано
21..14	CLK[7:0]		Скорость приёма данных Бит 0 – канал 1, бит 7 – канал 8. 1 – частота приёма данных = опорная частота/80 (12,5 кГц, если DIV не равен 0) 0 – частота приёма данных = опорная частота/10 (100 кГц, если DIV не равен 0) Опорная частота для каждого канала задаётся делителем в регистрах CONTROL4 и CONTROL5, если DIV=0 с ревизии 2
13..8			Зарезервировано
7..0	CH_EN[7:0]		Разрешение работы канала Бит 0 – канал 1, бит 7 – канал 8. 1 – приём по каналу разрешён 0 – канал приёма находится в состоянии сброса

7.21.4.3 CONTROL2

Регистр управления приёмников 2

Base ADDR=		0x4009_E000					Offset=		0x0000_0004							
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DA							SD_EN[7:0]									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					LB_EN[7:0]								DIV[6:4]		

Бит	Имя	Значение	Описание
31	DA		<p>Бит прямого доступа в FIFO1 и FIFO2 1 – память приёма каналов 1 и 2 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов</p> <p>0x400D1000 – 0x400D13FC для FIFO1 0x400D1400 – 0x400D14FC для FIFO2</p> <p>0 – обычный режим работы FIFO</p> <p>Адрес должен быть кратен четырём, обращения только по 32 разрядным словам. CHANNEL = 0 или 1 в зависимости от канала При приёме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.</p>
30..25			Зарезервировано
24..17	SD_EN[7:0]		<p>Разрешение декодирования бит данных 9 и 10 Бит 0 – канал 1, бит 7 – канал 8.</p> <p>1 – разрешено сравнение бит данных 9 и 10 со значением бит SDI1 и SDI2 соответствующего канала 0 – декодирование отключено, все принятые данные помещаются в FIFO</p>
16..11			Зарезервировано
10..3	LB_EN[7:0]		<p>Разрешение обнаружения меток Бит 0 – канал 1, бит 7 – канал 8.</p> <p>1 – разрешено обнаружение меток в первых 8 принятых битах 0 – обнаружение отключено, все принятые данные помещаются в FIFO</p>
2..0	DIV[6:4]		<p>Делитель частоты ядра до 1 МГц Содержит старшие 3 разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц</p>

7.21.4.4 CONTROL3

Регистр управления приёмников 3

Base ADDR=		0x4009_E000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
INTEHF	INTEFF	INTEER	INTEDR							SDI2_1						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI2_1							SDI1_1								

Бит	Имя	Значение	Описание
31	INTEHF		Разрешение прерывания FIFO наполовину полно 1 – разрешено прерывание, если FIFO наполовину полно 0 – прерывание запрещено
30	INTEFF		Разрешение прерывания FIFO полно 1 – разрешено прерывание при переполнении FIFO данных 0 – прерывание запрещено
29	INTEER		Разрешение прерывания ошибка приёма 1 – разрешено прерывания при возникновении ошибки в скорости приёма или во времени паузы 4Т между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено
28	INTEDR		Разрешение прерывания наличие данных в FIFO 1 – разрешено прерывание, если FIFO приёма данных не пусто 0 – прерывание запрещено
27..22			Зарезервировано
21..14	SDI2[7:0]		Бит сравнения SDI2 Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала Бит 0 – канал 1, бит 7 – канал 8.
13..8			Зарезервировано
7..0	SDI1[7:0]		Бит сравнения SDI1 Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала Бит 0 – канал 1, бит 7 – канал 8.

7.21.4.5 CONTROL4

Регистр управления приёмников 4

Base ADDR=		0x4009_E000					Offset=		0x0000_0014										
REG Name:																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
DIV_CH4[7:0]								DIV_CH3[7:0]											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
DIV_CH2[7:0]								DIV_CH1[7:0]											

Бит	Имя	Значение	Описание
31..24	DIV_CH4[7:0]		Делитель частоты ядра для получения опорной частоты канала 4
23..16	DIV_CH3[7:0]		Делитель частоты ядра для получения опорной частоты канала 3
15..8	DIV_CH2[7:0]		Делитель частоты ядра для получения опорной частоты канала 2
7..0	DIV_CH1[7:0]		Делитель частоты ядра для получения опорной частоты канала 1

7.21.4.6 CONTROL5

Регистр управления приёмников 5

Base ADDR=		0x4009_E000					Offset=		0x0000_0018										
REG Name:																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
DIV_CH8[7:0]								DIV_CH7[7:0]											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
DIV_CH6[7:0]								DIV_CH5[7:0]											

Бит	Имя	Значение	Описание
31..24	DIV_CH8[7:0]		Делитель частоты ядра для получения опорной частоты канала 8
23..16	DIV_CH7[7:0]		Делитель частоты ядра для получения опорной частоты канала 7
15..8	DIV_CH6[7:0]		Делитель частоты ядра для получения опорной частоты канала 6
7..0	DIV_CH5[7:0]		Делитель частоты ядра для получения опорной частоты канала 5

7.21.4.7 CONTROL8

Регистр управления приёмников 8

Base ADDR=		0x4009_E000				Offset=		0x0000_0070							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DA2	DA1							ODD[7:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODD[7:0]								ENPAR[7:0]							

Бит	Имя	Значение	Описание
31	-	-	Зарезервировано
30	-	-	Зарезервировано
29	DA2	0	Бит прямого доступа в FIFO2 1 – память приёма канала 2 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов 0x400D1400 – 0x400D14FC для FIFO2 0 – обычный режим работы FIFO Адрес должен быть кратен четырём, обращения только по 32 разрядным словам. CHANNEL = 1 При приёме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения. Бит DA регистра CONTROL2 должен быть в нуле.
28	DA1	0	Бит прямого доступа в FIFO1 1 – память приёма канала 1 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов 0x400D1000 – 0x400D13FC для FIFO1 0 – обычный режим работы FIFO Адрес должен быть кратен четырём, обращения только по 32-х разрядным словам. CHANNEL = 0 При приёме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения. Бит DA регистра CONTROL2 должен быть в нуле.
27...22	-	-	Зарезервировано
21..14	ODD[7:0]	0xFF	Выбор чётности или нечётности бита паритета для каналов 1–8 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
13...8	-	-	Зарезервировано
7...0	ENPAR[7:0]	0xFF	Разрешение 32 бита паритета для каналов 1–8 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных Запрещено сбрасывать этот бит в ноль в штатном режиме работы контроллера

7.21.4.8 CONTROL9

Регистр управления приёмников 9

Base ADDR=		0x4009_E000				Offset=		0x0000_0074									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								ENSYNC[7:0]								

Бит	Имя	Значение	Описание
31..24	-		Зарезервировано
23..16	-		Зарезервировано
15..8	-		Зарезервировано
7..0	ENSYNC[7:0]		Разрешение работы входов приёмника в режиме данных и синхросигнала 1 – разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала вход IN_A работает как данные (D), вход IN_B как синхросигнал (SYN)

7.21.4.9 INTMASK

Регистр индивидуального разрешения прерываний каналов

Base ADDR=		0x4009_E000				Offset=		0x0000_0068								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
IEHF8	IEFF8	IEER8	IEDR8	IEHF7	IEFF7	IEER7	IEDR7	IEHF6	IEFF6	IEER6	IEDR6	IEHF5	IEFF5	IEER5	IEDR5	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEHF4	IEFF4	IEER4	IEDR4	IEHF3	IEFF3	IEER3	IEDR3	IEHF2	IEFF2	IEER2	IEDR2	IEHF1	IEFF1	IEER1	IEDR1

Бит	Имя	Значение	Описание
31..28	IEDR8 IEER8 IEFF8 IEHF8	0000	Для канала 8
27..24	IEDR7 IEER7 IEFF7 IEHF7	0000	Для канала 7
23..20	IEDR6 IEER6 IEFF6 IEHF6	0000	Для канала 6
19..16	IEDR5 IEER5 IEFF5 IEHF5	0000	Для канала 5
15..12	IEDR4 IEER4 IEFF4 IEHF4	0000	Для канала 4
11..8	IEDR3 IEER3 IEFF3 IEHF3	0000	Для канала 3
7..5	IEDR2 IEER2 IEFF2 IEHF2	0000	Для канала 2
3..0	IEDR1 IEER1 IEFF1 IEHF1	0000	Для канала 1 IEDR1 1 – разрешено прерывание, если FIFO приёма данных не пусто 0 – прерывание запрещено IEER1 1 – разрешено прерывания при возникновении ошибки в скорости приёма или во времени паузы 4Т между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено IEFF1 1 – разрешено прерывание при переполнении FIFO данных 0 – прерывание запрещено

			IEHF1 1 – разрешено прерывание, если FIFO наполовину полно 0 – прерывание запрещено
--	--	--	---

7.21.4.10 STATUS1

Регистр состояния приёмников 1

Base ADDR=	0x4009_E000				Offset=	0x0000_000C									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										ERR[7:2]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR[1:0]								DR[7:0]							

Бит	Имя	Значение	Описание
31..22	-		Зарезервировано
21..14	ERR[7:0]		Бит ошибки. Бит 0 – канал 1, бит 7 – канал 8. 0 – нет ошибок 1 – возникла ошибка приёма
13..8	-		Зарезервировано
7..0	DR[7:0]		Бит наличия данных в FIFO. Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO пусто 1 – FIFO содержит данные

7.21.4.11 STATUS2

Регистр состояния приёмников 2

Base ADDR=		0x4009_E000				Offset=		0x0000_0010									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
										HF[7:2]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HF[1:0]								FF[7:0]							

Бит	Имя	Значение	Описание
31..22			Зарезервировано
21..14	HF[7:0]		Бит наполненности FIFO. Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO не наполнено до половины 1 – FIFO наполнено до половины
13..8			Зарезервировано
7..0	FF[7:0]		Бит полноты FIFO. Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO не полно 1 – FIFO полно

7.21.4.12 CHANNEL

Регистр номера канала приёмников

Base ADDR=		0x4009_E000				Offset=		0x0000_001C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CHAN[3:0]			

Бит	Имя	Значение	Описание
31...4	-		Зарезервировано
3...0	CHAN[3:0]		Биты выбора канала. Значение этих бит определяет к данным и меткам, какого канала будет осуществляться доступ. 0000 – канал 1 0001 – канал 2 0010 – канал 3 0011 – канал 4 0100 – канал 5 0101 – канал 6 0110 – канал 7 0111 – канал 8 1110 – прямой доступ в память каналов 1 и 2 при DR=1, либо доступ по индивидуальным адресам для каждого FIFO

7.21.4.13 LABEL

FIFO меток

FIFO меток, с которыми сравниваются первые 8 принимаемых бит, если установлен LB_EN бит соответствующего канала. Размер FIFO для каждого канала 16x8, либо 32x8. Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. При записи или чтении FIFO указатель FIFO инкрементируется. Для возврата в начало FIFO необходимо осуществить запись в регистр CHANNEL.

7.21.4.14 DATA_R

FIFO принимаемых данных

В FIFO помещаются 32 разрядные данные, принимаемые из соответствующего канала. Размер FIFO для каждого канала разный:

- канал 1 – 256x32;
- канал 2 – 256x32;
- канал 3 – 64x32;
- канал 4 – 64x32;
- канал 5 – 64x32;
- канал 6 – 64x32;
- канал 7 – 32x32;
- канал 8 – 32x32.

Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. Наличие или отсутствие данных в FIFO контролируется битами статуса DR, HF, FF соответствующего канала.

7.21.4.15 DATA_R1 DATA_R2 DATA_R3 DATA_R4 DATA_R5 DATA_R6 DATA_R7 DATA_R8

FIFO принимаемых данных в случае записи в регистр CHANNEL значения 14.

7.21.5 Описание регистров передатчика

Таблица 143 – Описание регистров передатчика

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x400A_0000		ARINC429T0		Контроллер интерфейса передатчиков ARINC429
	0x0000_0000	CONTROL1		Регистр управления передатчиков 1
	0x0000_0004	CONTROL2		Регистр управления передатчиков 2
	0x0000_0008	STATUS		Регистр состояния передатчиков
	0x0000_000C	DATA1_T		Регистр передаваемых данных канала 1
	0x0000_0010	DATA2_T		Регистр передаваемых данных канала 2
	0x0000_0014	DATA3_T		Регистр передаваемых данных канала 3
	0x0000_0018	DATA4_T		Регистр передаваемых данных канала 4
	0x0000_001C	CONTROL3		Регистр настройки индивидуального делителя частоты каналов
	0x0000_0020	CONTROL4		Регистр управления передатчиков 4
	0x0000_1000-0x0000_2000	Память передатчиков		Прямой доступ в память передатчиков

7.21.5.1 CONTROL1

Регистр управления передатчиками 1

Base ADDR=		0x400A_0000				Offset=		0x0000_0000								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											INTE_HFT2	INTE_TXR2	INTE_FFT2	INTE_HFT1	INTE_TXR1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
INTE_FFT1	DIV[6:0]						ODD2	EN_PAR2	CLK2	CH_EN2	ODD1	EN_PAR1	CLK1	CH_EN1		

Бит	Имя	Значение	Описание
31..21			Зарезервировано
20	INTE_HFT2		Разрешение прерывания при заполнении наполовину буфера FIFO канала 2 1 – разрешено прерывание, FIFO наполовину полно 0 – прерывание запрещено
19	INTE_TXR2		Разрешение прерывания при опустошении буфера FIFO канала 2 1 – разрешено прерывание, буфер FIFO пуст 0 – прерывание запрещено

Бит	Имя	Значение	Описание
18	INTE_FFT2		Разрешение прерывания при полном заполнении буфера FIFO канала 2 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
17	INTE_HFT1		Разрешение прерывания при заполнении наполовину буфера FIFO канала 1 1 – разрешено прерывание FIFO наполовину полно 0 – прерывание запрещено
16	INTE_TXR1		Разрешение прерывания при опустошении буфера FIFO канала 1 1 – разрешено прерывание FIFO передачи данных пусто 0 – прерывание запрещено
15	INTE_FFT1		Разрешение прерывания при полном заполнении буфера FIFO канала 1 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
14..8	DIV[6:0]		Делитель частоты ядра до 1 МГц Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц
7	ODD2		Выбор чётности или нечётности бита паритета для канала 2 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
6	EN_PAR2		Разрешение 32 бита паритета для канала 2 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
5	CLK2		Скорость передачи данных по 2 каналу 1- частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю) 0- частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)
4	CH_EN2		Разрешение работы канала 2 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
3	ODD1		Выбор чётности или нечётности бита паритета для канала 1 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
2	EN_PAR1		Разрешение 32 бита паритета для канала 1 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
1	CLK1		Скорость передачи данных по 1 каналу 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю) 0 – частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN1		Разрешение работы канала 1 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса

7.21.5.2 CONTROL2

Регистр управления передатчиков 2

Base ADDR=		0x400A_0000				Offset=		0x0000_0004								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
											INTE_HFT4	INTE_TXR4	INTE_FFT4	INTE_HFT3	INTE_TXR3	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
INTE_FFT3	-							ODD4	EN_PAR4	CLK4	CH_EN4	ODD3	EN_PAR3	CLK3	CH_EN3	

Бит	Имя	Значение	Описание
31..21			Зарезервировано
20	INTE_HFT4		Разрешение прерывания при заполнении наполовину буфера FIFO канала 4 1 – разрешено прерывание, FIFO наполовину полно 0 – прерывание запрещено
19	INTE_TXR4		Разрешение прерывания при опустошении буфера FIFO канала 4 1 – разрешено прерывание, буфер FIFO пуст 0 – прерывание запрещено
18	INTE_FFT4		Разрешение прерывания при полном заполнении буфера FIFO канала 4 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
17	INTE_HFT3		Разрешение прерывания при заполнении наполовину буфера FIFO канала 3 1 – разрешено прерывание FIFO наполовину полно 0 – прерывание запрещено
16	INTE_TXR3		Разрешение прерывания при опустошении буфера FIFO канала 3 1 – разрешено прерывание FIFO передачи данных пусто 0 – прерывание запрещено
15	INTE_FFT3		Разрешение прерывания при полном заполнении буфера FIFO канала 3 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
14..8	-		Зарезервировано
7	ODD4		Выбор чётности или нечётности бита паритета для канала 4 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)

6	EN_PAR4		Разрешение 32 бита паритета для канала 4 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
5	CLK4		Скорость передачи данных по 4 каналу 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю) 0 – частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)
4	CH_EN4		Разрешение работы канала 4 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
3	ODD3		Выбор чётности или нечётности бита паритета для канала 3 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
2	EN_PAR3		Разрешение 32 бита паритета для канала 3 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
1	CLK3		Скорость передачи данных по 3 каналу 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю) 0 – частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN3		Разрешение работы канала 3 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса

7.21.5.3 CONTROL3

Регистр управления передатчиков 3

Base ADDR=		0x400A_0000				Offset=		0x0000_001C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_CH4[7:0]								DIV_CH3[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_CH2[7:0]								DIV_CH1[7:0]							

Бит	Имя	Значение	Описание
31..24	DIV_CH4		Делитель частоты ядра для получения опорной частоты канала 4
23..16	DIV_CH3		Делитель частоты ядра для получения опорной частоты канала 3
15..8	DIV_CH2		Делитель частоты ядра для получения опорной частоты канала 2
7..0	DIV_CH1		Делитель частоты ядра для получения опорной частоты канала 1

7.21.5.4 CONTROL4

Регистр управления передатчиков 4

Base ADDR=		0x400A_0000				Offset=		0x0000_0020							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ENSYNC[3:0]			

Бит	Имя	Значение	Описание
31..24	-		
23..16	-		
15..4	-		
3..0	ENSYNC[3:0]		Разрешение работы выходов передатчика в режиме данных и синхросигнала 1- разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала выход OUT_A работает как данные (D), выход OUT_B как синхросигнал(SYN).

7.21.5.5 STATUS

Регистр состояния передатчиков

Base ADDR=		0x400A_0000				Offset=		0x0000_0008								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		HFT4	FFT4	TX_R4	HFT3	FFT3	TX_R3			HFT2	FFT2	TX_R2	HFT1	FFT1	TX_R1

Бит	Имя	Значение	Описание
31..14			Зарезервировано
13	HFT4		Флаг наполненности FIFO канала 4 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
12	FFT4		Флаг полноты FIFO канала 4 1 – FIFO полно 0 – FIFO неполно
11	TX_R4		Флаг наличия данных в FIFO канала 4 1 – FIFO пусто 0 – FIFO содержит данные
10	HFT3		Флаг наполненности FIFO канала 3 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
9	FFT3		Флаг полноты FIFO канала 3 1 – FIFO полно 0 – FIFO неполно
8	TX_R3		Флаг наличия данных в FIFO канала 3 1 – FIFO пусто 0 – FIFO содержит данные
7..6			Зарезервировано
5	HFT2		Флаг наполненности FIFO канала 2 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
4	FFT2		Флаг полноты FIFO канала 2 1 – FIFO полно 0 – FIFO неполно
3	TX_R2		Флаг наличия данных в FIFO канала 2 1 – FIFO пусто 0 – FIFO содержит данные
2	HFT1		Флаг наполненности FIFO канала 1 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
1	FFT1		Флаг полноты FIFO канала 1 1 – FIFO полно 0 – FIFO неполно
0	TX_R1		Флаг наличия данных в FIFO канала 1 1 – FIFO пусто 0 – FIFO содержит данные

7.21.5.6 DATA1_T

FIFO передаваемых данных канала 1

FIFO может содержать данные объёмом 256x32 для передачи по каналу 1. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R1, HFT1, FFT1.

7.21.5.7 DATA2_T

FIFO передаваемых данных канала 2

FIFO может содержать данные объёмом 64x32 для передачи по каналу 2. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R2, HFT2, FFT2.

7.21.5.8 DATA3_T

FIFO передаваемых данных канала 3

FIFO может содержать данные объёмом 64x32 для передачи по каналу 3. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R3, HFT3, FFT3.

7.21.5.9 DATA4_T

FIFO передаваемых данных канала 4

FIFO может содержать данные объёмом 64x32 для передачи по каналу 4. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R4, HFT4, FFT4.

7.22 Контроллер интерфейса МКПД (MILx)

В микроконтроллере имеется два независимых контроллера интерфейса мультимплексного канала передачи данных по ГОСТ Р 52070-2003 (далее МКПД). Каждый из которых, содержит необходимую логику и память для обработки и хранения командных слов и слов данных одного полного сообщения МКПД. Каждый контроллер содержит два канала для приёма/передачи сообщений МКПД: основной и резервный. В один момент времени может работать только один из каналов основной или резервный. Одновременная работа двух каналов не предусмотрена. Контроллер может работать как в режиме контроллера шины, так и в режиме оконечного устройства. Для хранения входящих и исходящих командных и статусных слов, а также команд управления используются 16-разрядные регистры. Для хранения данных используется шестнадцатиразрядная двухпортовая память, в которой данные хранятся в области памяти, соответствующей подадресу командного слова. В каждом подадресе можно хранить только одно полное сообщение МКПД. При передаче сообщения данные в память можно заносить как на «лету», так и до начала передачи. При приёме сообщения, данные можно считывать из памяти, как на «лету», так и после установки флага VALMESS.

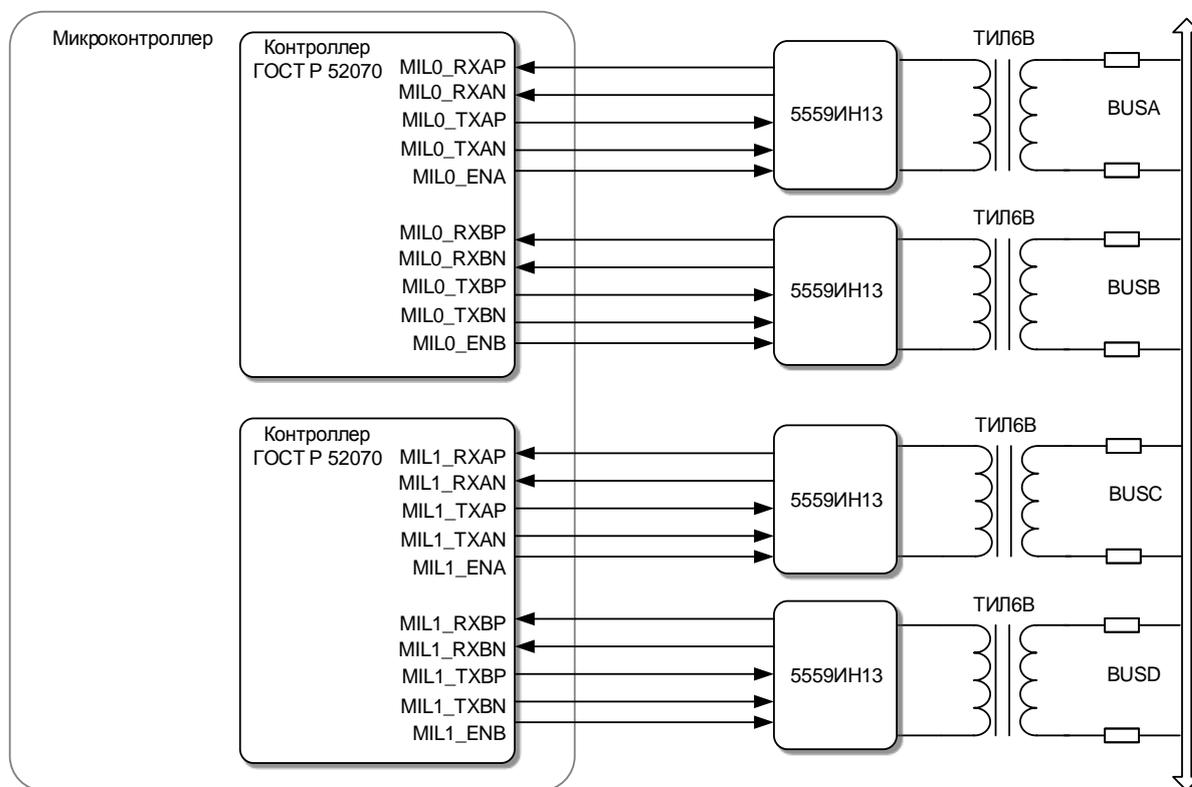


Рисунок 155 – Структурная схема контроллера интерфейса МКПД

Особенности:

- Поддержка основных (формат 1 – формат 6) и групповых (формат 7 – формат 10) форматов сообщений;
- Поддержка режимов работы: контроллер шины, оконечное устройство, монитор;
- Скорость передачи данных 1 Мбит/с в полудуплексном режиме;
- Поддержка двух каналов связи: основного и резервного;
- Двухпортовая память принимаемых данных 1Кx16;
- Двухпортовая память передаваемых данных 1Кx16;
- Возможность формирования прерываний при успешном приёме и при возникновении ошибок на шине;
- Маскирование прерываний.

7.22.1 Режимы работы

Контроллер поддерживает три режима работы: контроллера шина (КШ), оконечного устройства (ОУ) и неадресуемого монитора (М).

Контроллер шины

В этом режиме контроллер передаёт команды в магистраль, участвует в пересылке слов данных, принимает и контролирует ответную информацию о состоянии ОУ. Помимо этого КШ реализует все команды управления. Для того чтобы реализовать передачу командного слова в магистраль, используется регистр CommandWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр CommandWord2. Ответная информация о состоянии ОУ после приема из магистрали хранится в регистре StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для передачи и приёма слов данных команд управления (КУ), форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита BCMODE и сбросом RTMODE.

Оконечное устройство

В этом режиме контроллер осуществляет проверку достоверности командных слов, поступающих к нему от КШ. Командное слово считается достоверным, если не возникло ошибок в магистрали при его приёме, или если поле «Адрес ОУ» соответствует коду собственного адреса ОУ или коду 11111 (групповая команда). Если командное слово определено как достоверное, то ОУ посылает в линию ответное слово (ОС) и в зависимости от поля «Приём/Передача» принимает или передаёт число данных, соответствующее полю «Число СД/Код КУ». Если же происходит приём от КШ команды управления, то ОУ реагирует в соответствии с форматами сообщений команд управления. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ для передачи в магистраль помещается в регистр StatusWord1. Помимо этого, для сообщения формата 3, этот регистр содержит принятое ответное слово от другого ОУ. Для передачи и приёма слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита RTMODE и сбросом BCMODE.

Монитор

В этом режиме осуществляется прослушивание магистрали и отбор необходимой информации для проведения: технического обслуживания, регистрации эксплуатационных параметров, анализа решаемых задач или обеспечения информацией резервного КШ. Монитор пассивно прослушивает выбранную шину и захватывает весь трафик на шине, но никогда не передаёт информацию на шину. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ принятое из магистрали помещается в регистр StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для приёма слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой битов RTMODE и BCMODE. Для быстрой расшифровки сообщений можно применить регистр MSG. Каждому формату сообщения на магистрали соответствует определённый код в этом регистре.

7.22.2 Форматы сообщений

Сообщения, передаваемые по информационной магистрали, имеют формат, соответствующий форматам основных или групповых сообщений. Любые другие типы сообщений, не соответствующие ГОСТ Р 52070-2003 не поддерживаются.

Форматы основных сообщений, приведённые ниже (Рисунок 156), используются для передачи информации предназначенной одному ОУ и предусматривают выдачу ОС. В данном случае КС – командное слово, СД – слово данных, ОС – ответное слово. Времена t_1 и t_2 формируются аппаратно и не могут быть изменены программно. Пауза t_2 между сообщениями, формируемая КШ, не менее 4 мкс. А пауза перед передачей ОС, формируемая ОУ, в пределах от 4 до 12 мкс. Если после ожидания 14 мкс так и не поступило ОС от ОУ, то фиксируется отсутствие ОС от ОУ и формируется соответствующий признак ошибки.

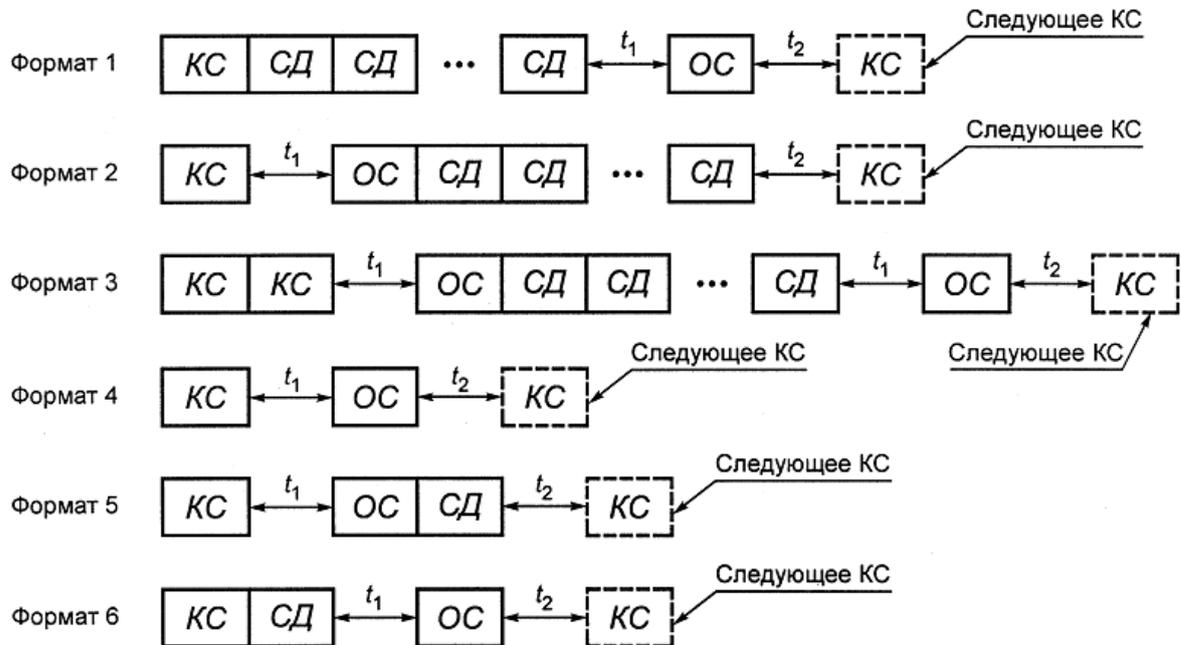


Рисунок 156 – Форматы сообщений (часть 1)

Время непрерывной передачи данных в линию не превышает 660 мкс, что соответствует командному слову и 32-м словам данных.

- Формат 1 – передача данных от КШ к ОУ
- Формат 2 – передача данных от ОУ к КШ
- Формат 3 – передача данных от ОУ к ОУ
- Формат 4 – передача КУ
- Формат 5 – передача КУ и приём СД от ОУ
- Формат 6 – передача КУ и СД оконечному устройству

Групповые сообщения, приведённые ниже (Рисунок 157), начинающиеся с передачи КШ групповой команды с кодом адреса 11111, используются для передачи информации одновременно нескольким ОУ без выдачи ими ОС.

- Формат 7 – передача данных (в групповом сообщении) от КШ к оконечным устройствам
- Формат 8 – передача данных (в групповом сообщении) от оконечного устройства к оконечным устройствам
- Формат 9 – передача групповой команды управления
- Формат 10 – передача групповой команды управления со словом данных

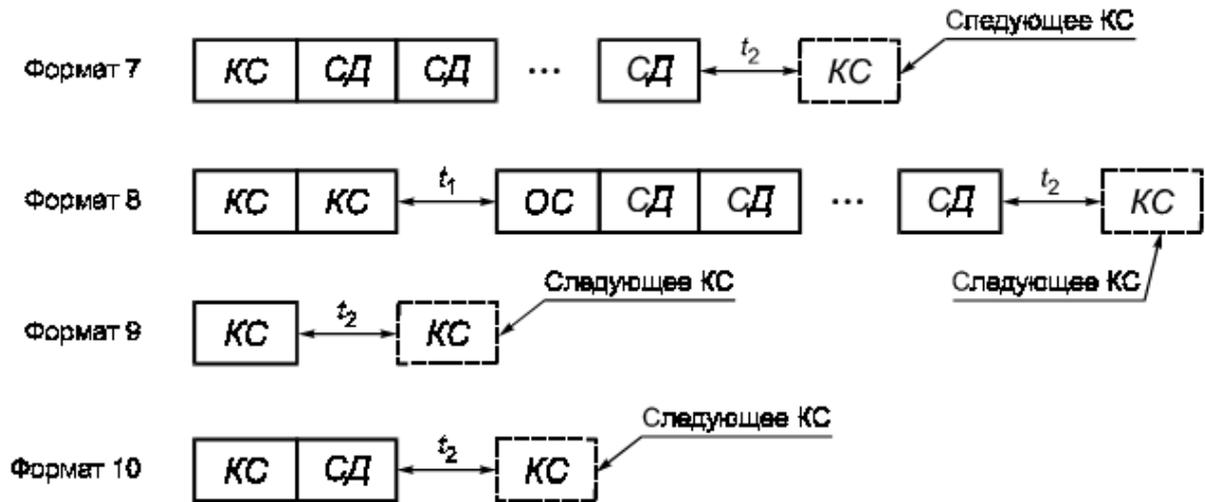


Рисунок 157 – Форматы сообщений (часть 2)

Если ОУ в формате сообщения ОУ-ОУ получило достоверное командное слово на приём информации, то первое СД должно быть им принято через паузу не более (57±3) мкс, в противном случае формируется соответствующий признак ошибки.

7.2.2.3 Формат слов

Каждое слово начинается с сигнала пословной синхронизации (с синхросигнала) и имеет 17 информационных разрядов, включая разряд контроля по чётности. Форматы слов приведены на рисунке ниже (Рисунок 158).

Разрядная сетка	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Командное слово	Синхро-сигнал			Адрес ОУ				К	Подадрес Режим управления				Число СД Код команды				Р				
Слово данных	Синхро-сигнал			Данные																Р	
Ответное слово	Синхро-сигнал			Адрес ОУ				Признаки													Р
	1 - 3			4 - 8				9	10	11	12 - 14				15	16	17	18	19	20	
								Ошибка в сообщении	Передача ОС	Запрос на обслуживание	Резерв				Принята групповая команда	Абонент занят	Неисправность абонента	Принято управление интерфейсом	Неисправность ОУ		

Рисунок 158 – Форматы слов

Командное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- разряд «Приём/Передача»;
- поле «Подадрес/Режим управления»;
- поле «Число СД/Код КУ»;
- разряд контроля по чётности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала положительная, а вторая отрицательная.

Адрес ОУ содержит код адреса из диапазона кодов 00000 – 11110, которому предназначено КС. КС с кодом адреса 11111 называется групповой командой, а сообщение, содержащее групповую команду – групповым.

Разряд «Приём/Передача» указывает на действие, которое должно выполнить ОУ (принимать или передавать СД). Логический нуль означает, что ОУ должно принимать СД, а логическая единица – передавать СД.

Поле «Подадрес/Режим управления» содержит код подадреса ОУ или код признака режима управления 00000 или 11111.

Поле «Число СД/Код КУ» содержит код числа слов данных, которые должны быть переданы или приняты ОУ в связи с приёмом адресованного ему КС, или код КУ. В одном сообщении может быть передано не более 32 СД. Числовое значение двоичных кодов, обозначающих число СД, соответствует их десятичным эквивалентам, за исключением кода 00000, который соответствует числу 32.

Разряд контроля по чётности используется для контроля по чётности предшествующих ему 16 разрядов КС. Разряд принимает такое значение, чтобы сумма значений всех 17 информационных разрядов слова (включая контрольный разряд) была нечётной.

Слово данных содержит:

- синхросигнал;
- данные;
- разряд контроля по чётности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала отрицательная, а вторая положительная.

Поле данных содержит передаваемые данные, а разряд контроля по чётности формируется так же, как в командном слове.

Ответное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- поле разрядов признаков состояния: ошибка в сообщении, передача ОС, запрос на обслуживание, принята групповая команда, абонент занят, неисправность абонента, принято управление интерфейсом, неисправность ОУ;
- разряд контроля по четности.

Синхросигнал аналогичен синхросигналу КС. Поле «Адрес ОУ» содержит собственный адрес ОУ. Поле разрядов признаков состояния ОУ отображает текущее состояние ОУ. Разряд контроля по чётности формируется так же, как в командном слове.

7.22.4 Структурная схема в режиме КШ

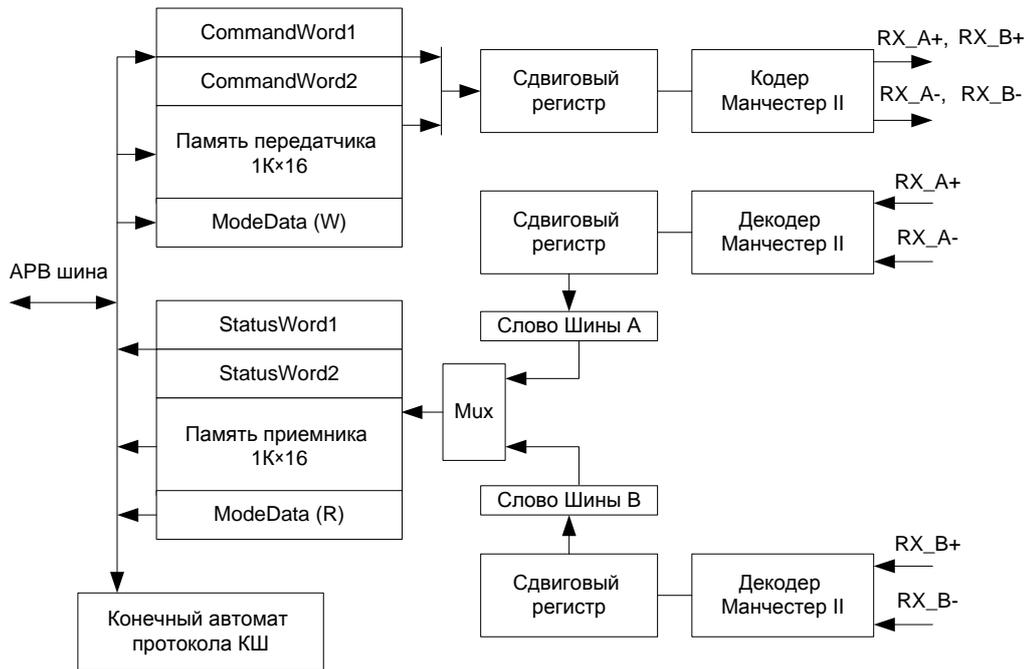


Рисунок 159 – Структурная схема в режиме КШ

7.22.5 Структурная схема в режиме ОУ

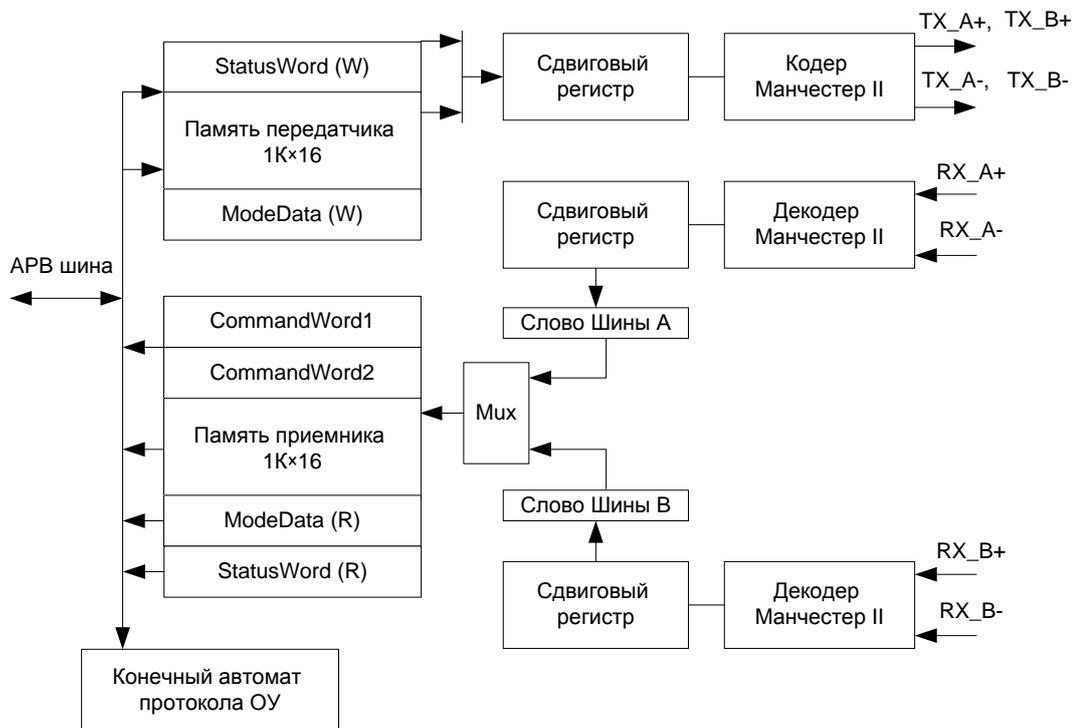


Рисунок 160– Структурная схема в режиме ОУ

7.22.6 Структурная схема в режиме М

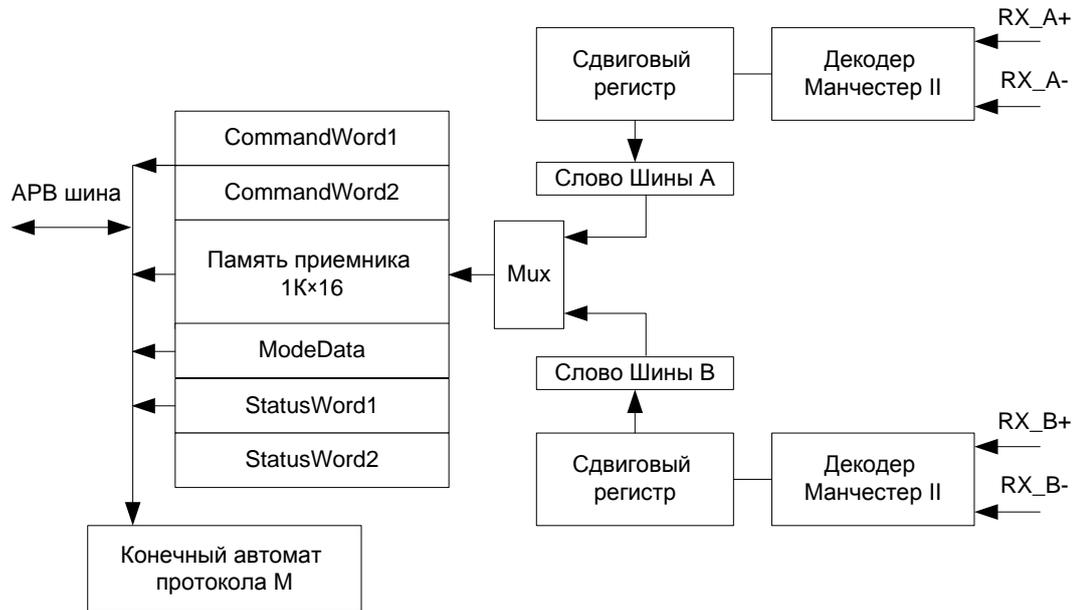


Рисунок 161 – Структурная схема в режиме М

7.22.7 Инициализация

Перед началом работы с контроллером в первую очередь необходимо сбросить контроллер, чтобы очистить все регистры сообщений. Это достигается установкой бита MR регистра CONTROL в логическую единицу. Затем бит необходимо сбросить в нуль. Далее нужно задать в регистре CONTROL значение делителя частоты DIV таким образом, чтобы при делении частоты ядра HCLK на это значение получить опорную частоту блока контроллера 1 МГц. После этого с помощью бит RTMODE и BCMODE выбирается соответствующий режим работы ОУ или КШ. Одновременная установка двух этих бит запрещена.

Для того чтобы выбрать какой канал будет использован для передачи данных основной или резервный, устанавливается соответствующий бит (TRA – основной канал, TRB – резервный канал). В режиме КШ командные слова будут передаваться только по тому каналу, который выбран. В режиме ОУ необходимо установить оба бита, так как ОУ не может выбирать, по какому каналу ему передавать СД и ОС, и поэтому их передача происходит по тому каналу, по которому было принято КС.

Для режима ОУ в битах RTA4 – RTA0 регистра CONTROL задается адрес ОУ, который должен соответствовать адресу в поле «Адрес ОУ» командного слова, если идет обращение к этому ОУ.

Пример инициализации ОУ

```
*((volatile unsigned int *) (0x400A6000))=0x00000001; //установка бита MR=1
```

```
*((volatile unsigned int *) (0x400A6000))=0x00014078;
```

```
//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40
```

Пример инициализации КШ

```
*((volatile unsigned int *) (0x400A6000))=0x00000001; //установка бита MR=1
```

```
*((volatile unsigned int *) (0x400A6000))=0x00014014;
```

```
//BCMODE=1, TRA=1, TRB=0, RTA=0, DIV=40
```

В обоих случаях значения делителя частоты DIV = 40, что соответствует частоте ядра 40 МГц, и для получения опорной частоты контроллера необходимо 40 МГц/DIV = 1 МГц.

7.22.8 Приём и передача в режиме ОУ

Для того, чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты, описанные в подразделе 7.22.7 «Инициализация». После этого необходимо задать ответное слово для КШ с помощью регистра StatusWord1. В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ответное слово, полученное от передающего ОУ в транзакции ОУ-ОУ.

Пример записи ответного слова ОУ

```
*((volatile unsigned int *) (0x40051018))=0x00000800;
```

В данном случае в регистр заносятся только старшие 5 разрядов, соответствующие адресу ОУ. Остальные разряды признаки состояния ОУ можно оставить в нуле. Но в процессе работы может возникать необходимость изменять эти биты. Для этого необходимо программно устанавливать и сбрасывать эти биты, так как аппаратно они не изменяются.

Для того чтобы обеспечить формат сообщения 5, необходимо задавать слово данных, передаваемое КШ в команде управления. Для этих целей используется регистр ModeData.

Пример записи слова данных команды управления

```
*((volatile unsigned int *) (0x40051014))=0x000055AA;
```

После того как проведена инициализация, заданы ответное слово и слово данных команды управления, ОУ сразу готово к работе и может отвечать на все возможные форматы сообщений.

Так как в процессе работы ОУ в каждый момент времени требуется передача определённых СД и СД команды управления, то программно необходимо обновлять те области памяти, которые содержат эти данные. Без обновления при запросе данных КШ будут переданы последние записанные в эти области памяти данные. Поэтому при написании программы следует помнить и обновлять данные и слова данных команды управления.

Для хранения СД применяется адресное пространство 0x000-0xFFC (относительно базового адреса периферийного блока). Данные 16-ти разрядные, но обращение к ним должно быть выровнено по границе 32-х разрядного слова. То есть 2 младших разряда не участвуют в формировании адреса.

Пример инициализации данных для подадреса 1

```
addon=0x80;
```

```
for (i=1;i<=32;i++)
```

```
{
```

```
*((volatile unsigned int *) (0x40050000+addon))=i;
```

```
addon = addon +4;
```

}

Из примера видно, что стартовый адрес памяти СД для подадреса 1 – 0x80, для последующих подадресов: n*0x80, где n-номер подадреса (n=1-31).

При приёме СД или слова данных команды управления, признаком обновления их значений является флаг VALMESS. После того как флаг установлен можно считать новые данные или слово данных команды управления. Но следует учитывать то, что этот флаг автоматически сбрасывается через 4 мкс, после его установки, поэтому желательно применять прерывания по установке сигнала VALMESS.

Пример чтения слова данных команды управления

```
i=((volatile unsigned int *) (0x40051014));
```

В переменную *i* будет прочитано значение слова данных команды управления.

Пример чтения данных для подадреса 1

```
addon=0x80;
```

```
for (i=1;i<=32;i++)
```

```
{
```

```
    mas[i] = ((volatile unsigned int *) (0x40050000+addon));
```

```
    addon = addon + 4;
```

```
}
```

Для упрощения декодирования команд управления КШ в режиме ОУ доступен регистр кодов MSG полученных сообщений. Каждому формату сообщения на магистрали соответствует определённый код в этом регистре. Чтение и последующая дешифрация этого кода упрощает процедуру декодирования сообщения и уменьшает время обработки сообщений. При использовании регистра экономится время на чтение двух командных регистров и разбор значений бит этих регистров.

7.22.9 Прием и передача в режиме КШ

В отличие от режима ОУ, в режиме КШ необходимо задавать не ответное слово, а командное слово или два командных слова в режиме работы ОУ-ОУ. Помимо этого нужно инициировать процедуру приема или передачи данных установкой бита BCSTART. После завершения транзакции на шине этот бит автоматически сбрасывается в ноль. Поэтому для инициирования новой транзакции нужно повторно устанавливать этот бит.

Пример записи командных слов и бита BCSTART

```
((volatile unsigned int *) (0x4005100C))=0x00000820; //Командное слово 1
```

```
((volatile unsigned int *) (0x40051010))=0x00000000; //Командное слово 2
```

```
((volatile unsigned int *) (START_ADDR_APB+0x1000))=0x00014016; //Регистр управления
```

Как видно из примера, в командном слове 1 задаётся код слов данных 00000, что соответствует 32 СД. Данные будут передаваться от контроллера шины оконечному устройству с адресом 1 из подадреса 1. Второе командное слово задаётся равным нулю и никак не влияет на транзакцию. В регистре управления устанавливается бит BCMODE, что соответствует режиму работы КШ, а также устанавливается бит BCSTART, что инициирует начало транзакции, выбирается канал А для передачи (TRA=1), а также устанавливается делить частоты 40, что соответствует частоте работы ядра 40 МГц.

Для того чтобы инициировать приём в этом примере, необходимо только установить бит 10 равным единице в командном слове 1.

Если транзакция завершена успешно (признак VALMESS установился в единицу), то полученные СД или слово данных команды управления могут быть прочитаны. В противном случае устанавливается один из флагов ошибки. Сброс этих флагов осуществляется либо установкой битом MR, либо инициированием новой транзакции битом BCSTART.

7.22.10 Прерывания

Для уменьшения потерь времени программы на опрос флагов контроллера, введено одно прерывание, генерируемое при установке любого из флагов контроллера. Прерывание может генерировать установка одного из четырёх флагов:

- флаг ошибки;
- флаг успешного завершения транзакции в канале;
- флаг приёма достоверного КС, ОС или слова данных команды управления;
- флаг неактивности контроллера.

Каждый из флагов может быть маскирован битами разрешения прерывания по какому-либо флагу.

7.22.11 Описание регистров

Таблица 144 – Описание регистров контроллера интерфейса МКПД

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x400A_6000		MIL-STD-1553B0		
0x400A_8000		MIL-STD-1553B1		
	0x0000-0x0FFC	DATA		Память принимаемых/передаваемых СД
	0x1000	CONTROL		Регистр управление контроллером
	0x1004	STATUS		Регистр состояния контроллера
	0x1008	ERROR		Регистр ошибок контроллера
	0x100C	CommandWord1		Регистр командного слова 1
	0x1010	CommandWord2		Регистр командного слова 2
	0x1014	ModeData		Слово данных команды управления
	0x1018	StatusWord1		Регистр ответного слова 1
	0x101C	StatusWord2		Регистр ответного слова 2
	0x1020	INTEN		Регистр разрешения прерываний
	0x1024	MSG		Регистр декодирования сообщений

7.22.11.1 CONTROL

Регистр управления

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x1000							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								RT_HW	INPINV	AUTOTUNE	ENFILTER	INVTR	RERR	DIV[6:5]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[4:0]					RTA4	RTA3	RTA2	RTA1	RTA0	TRB	TRA	RTMODE	BCMODE	BCSTART	MR

Бит	Имя	Значение	Описание
31..24			Зарезервировано
23	RT_HW		Бит аппаратной поддержки ГОСТ 51765-2001 для режима ОУ. 1 – аппаратная поддержка разрешена 0 – аппаратная поддержка запрещена При установке этого бита достаточно программной реализация циклического подадреса, установки и сброса признаков абонент занят, неисправность абонента, неисправность ОУ, запрос на обслуживание, а также задание адреса ОУ и бита паритета посредством внешнего соединения.
22	INPINV		Бит инверсии входов приёмника MILx_RXxP и MILx_RXxN 1 – инверсия разрешена 0 – инверсия запрещена Вход MILx_RXxP инвертируется и коммутируется на отрицательный вход декодера манчестерского кода. Вход MILx_RXxN инвертируется и коммутируется на положительный вход декодера манчестерского кода. Применение этого бита актуально для приёмопередатчиков с принудительной установкой выхода приемника в состояние логическая «1», например, 5559ИН74Т (Н1574).
21	AUTOTUNE		Бит автоматической подстройки середины битовых интервалов (с ревизии 3) 0 – автоподстройка разрешена 1- автоподстройка запрещена
20	ENFILTER		Бит разрешения фильтрации потока NRZ (с ревизии 3) 1 – фильтрация разрешена 0 – фильтрация запрещена В случае применения драйверов с некорректной скважностью и длительностью импульсов NRZ кода, таких как 5559ИН67Т, необходимо устанавливать бит в единицу для корректного приёма. В этом случае контроль длительностей импульсов NRZ не осуществляется.
19	INVTR		Разрешение инверсии сигналов (с ревизии 3) MIL0_ENA, MIL0_ENB, MIL1_ENA, MIL1_ENB 1 – инверсия 0 – прямой выход

Бит	Имя	Значение	Описание
18	RERR		Сброс ошибок в режиме ОУ 1 – ошибки могут быть сброшены только битом MR 0 – сброс ошибок происходит автоматически, после установки бита IDLE
17..11	DIV[6:0]		Делитель частоты PCLK блока МКИО до 1 МГц Содержит значение, на которое необходимо поделить частоту PCLK, чтобы получить 1 МГц. Частота PCLK обязательно должна быть не более 120 МГц и кратна 8. Если PCLK не кратна 8, то DIV[6:3]=(PCLK/8)+1, DIV[2:0]=0, но стабильность приёма не гарантируется.
10..6	RTA4-RTA0		Адрес оконечного устройства Содержит адрес, который присвоен устройству, если контроллер работает в режиме оконечного устройства RTMODE=1; BCMODE=0
5	TRB		Блокировка передатчика резервного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
4	TRA		Блокировка передатчика основного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
3..2	RTMODE BCMODE		Выбор режима работы контроллера 10 – режим оконечного устройства 01 – режим контроллера шины 11 – режим неадресуемого монитора
1	BCSTART		Иницирует передачу сообщения в канал в режиме КШ. 1 – старт сообщения 0 – стоп сообщения Сбрасывается в ноль автоматически по завершению сообщения.
0	MR		Сброс контроллера. 1 – контроллер сбрасывается в исходное состояние 0 – разрешение работы контроллера

7.22.11.2 STATUS

Регистр состояния

Base ADDR=		0x400A_6000				Offset=				0x1004											
		0x400A_8000																			
REG Name:																					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
					RCVB_	RCVA_	ERR_	VALMESS_	RFLAGN_	RCVB	RCVA	ERR	VALMESS	RFLAGN	IDLE					
					stick	stick	stick	stick	stick											

Бит	Имя	Значение	Описание
31..11	-		Зарезервировано
10	RCVB_stick		Признак активности резервного канала 0 – канал неактивен 1 – канал активен Устанавливается при установке бита RCVB. Сбрасывается программной записью нуля, если RCVB=0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету.
9	RCVA_stick		Признак активности основного канала 0 – канал неактивен 1 – канал активен Устанавливается при установке бита RCVA. Сбрасывается программной записью нуля, если RCVA=0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету.
8	ERR_stick		Ошибка в сообщении. 0 – нет ошибок 1 – в последней транзакции возникла ошибка Устанавливается при установке бита ERR. Сбрасывается программной записью нуля, если ERR=0. В установленном состоянии не формирует прерывание и не влияет на приём и передачу пакетов. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету.
7	VALMESS_stick		Успешное завершение транзакции в канале. 0 – транзакция завершена с ошибкой, либо транзакции нет в канале 1 – транзакция завершена успешно Устанавливается при установке бита VALMESS. Сбрасывается программной записью нуля, если VALMESS =0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету.

Бит	Имя	Значение	Описание
6	RFLAGN_stick		Получено достоверное слово из канала. 0 – нет достоверных слов в канале 1 – в режиме КШ получено достоверное ответное слово 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления Устанавливается при установке бита RFLAGN. Сбрасывается программной записью нуля, если VALMESS =0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету.
5	RCVB		Признак активности резервного канала 0 – канал неактивен 1 – канал активен
4	RCVA		Признак активности основного канала 0 – канал неактивен 1 – канал активен
3	ERR		Ошибка в сообщении. 0 – нет ошибок 1 – в последней транзакции возникла ошибка В режиме ОУ и М, если сброшен бит RERR, автоматически сбрасывается не менее чем через 4 мкс после установки.
2	VALMESS		Успешное завершение транзакции в канале. 0 – транзакция завершена с ошибкой, либо транзакции нет в канале 1 – транзакция завершена успешно В режиме ОУ и М автоматически сбрасывается не менее чем через 4 мкс после установки.
1	RFLAGN		Получено достоверное слово из канала. 0 – нет достоверных слов в канале 1 – в режиме КШ получено достоверное ответное слово 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления Между сообщениями бит автоматически сбрасывается в ноль.
0	IDLE		Состояние контроллера. 1 – контроллер в неактивном состоянии 0 – контроллер в состоянии обмена сообщениями

7.22.11.3 ERROR

Регистр ошибок

Base ADDR=	0x400A_6000	Offset=	0x1008												
REG Name:	0x400A_8000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									PROERR	CONERR	GAPERR	CSYCERR/ SEQERR	DSYCERR/ SYNCERR	MANERR	NORCV

Бит	Имя	Значение	Описание
31..7			Зарезервировано
6	PROERR		Ошибка в протоколе. 1 – недопустимое слово обнаружено на шине во время обмена сообщениями 0 – нет ошибок
5	CONERR		Ошибка непрерывности сообщения. 1 – передача сообщения не непрерывная 0 – нет ошибок
4	GAPERR		Недопустимая активность на шине. 1 – обнаружена активность на шине в интервале 4 мкс после успешного завершения сообщения 0 – нет ошибок
3	CSYCERR/ SEQERR		Ошибка синхронизации команды в режиме КШ (CSYCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных 0 – ошибок нет Ошибка после приёма команды в режиме ОУ и М (SEQERR). 1 – обнаружена пауза после приёма командного слова с битом 10 в регистре команды 1 (CommandWord1), равным нулю, или обнаружены слова данных после приёма командного слова с битом 10 (CommandWord1), равным единице 0 – ошибок нет
2	DSYCERR/ SYNCERR		Ошибка синхронизации данных в режиме КШ (DSYCERR). 1 – ожидался синхроимпульс данных, а получен синхроимпульс команды 0 – ошибок нет Ошибка синхронизации в режиме ОУ и М (SYNCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных или наоборот 0 – ошибок нет
1	MANERR		Ошибка декодирования NRZ кода. 1 – ошибка в количестве принятых бит или ошибка в бите контроля чётности 0 – ошибок нет
0	NORCV		Ошибка приёма. 1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные 0 – ошибок нет

7.22.11.4 CommandWord1

Регистр команды 1

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x100C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Адрес ОУ					Приём/ Передача	Подадрес/ Режим управления					Число СД/ Код команды				

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача		Бит приёма/передачи 1 – режим работы ОУ-КШ 0 – режим работы КШ-ОУ
9..5	Подадрес / Режим управления		Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД. В случае передачи команды, содержит код 00000 или 11111
4..0	Число СД / Код команды		Содержит количество принимаемых или передаваемых слов данных. В случае передачи команды содержит код команды из Таблицы 1 ГОСТ Р52070-2003

Примечание – В режиме ОУ и М регистр доступен только на чтение, в режиме КШ – только на запись.

7.22.11.5 CommandWord2

Регистр команды 2

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x1010							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Адрес ОУ					1	Подадрес/ Режим управления					Число СД/ Код команды				

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача		Бит приёма/передачи 1 – режим работы ОУ-ОУ 0 – командное слово не используется
9..5	Подадрес		Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД
4..0	Число СД		Содержит количество принимаемых или передаваемых слов данных

Примечание – В режиме ОУ и М регистр доступен только на чтение и содержит второе командное слово транзакции ОУ-ОУ. В режиме КШ регистр доступен только на запись и используется для транзакции ОУ-ОУ, если установлен в единицу бит 10.

7.22.11.6 ModeData

Слово данных команды управления

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x1014							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Слово данных команды управления															

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..0	ModeData		Содержит принятое или передаваемое слово данных в команде управления

7.22.11.7 StatusWord1

Ответное слово 1

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x1018								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Адрес ОУ	Ошибка в сообщ.	Пер.ОС	Запр. на обл.	-	-	-	Прин. ГК	Абон. зан.	Неисп. абон.	Прин. упр. инт.	Неисп. ОУ

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес ОУ, от которого принято ответное слово в режиме КШ. Адрес ОУ, которое передаёт ответное слово в режиме ОУ
10	Ошибка в сообщ.		Ошибка в сообщении
9	Пер.ОС		Передача ответного слова
8	Запр. на обл.		Запрос на обслуживание
7-5			Зарезервировано
4	Прин. ГК		Принята групповая команды
3	Абон. зан.		Абонент занят
2	Неисп. абон.		Неисправность абонента
1	Прин. упр. инт.		Принято управление интерфейсом
0	Неисп. ОУ		Неисправность ОУ

Примечание – В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ОС, полученное от передающего ОУ в транзакции ОУ-ОУ. В режиме КШ и М регистр доступен только на чтение и содержит принятое от ОУ ответное слово.

7.22.11.8 StatusWord2

Ответное слово 2

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x101C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Адрес ОУ					Ошибка в сообщ.	Пер. ОС	Запр. на обл.				Прин. ГК	Абон. зан.	Неисп. абон.	Прин. упр. инт.	Неисп ОУ

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес ОУ, передающего данные в транзакции ОУ-ОУ
10	Ошибка в сообщ.		Ошибка в сообщении
9	Пер. ОС		Передача ответного слова
8	Запр. на обл.		Запрос на обслуживание
7-5			Зарезервировано
4	Прин. ГК		Принята групповая команды
3	Абон. зан.		Абонент занят
2	Неисп. абон.		Неисправность абонента
1	Прин. упр. инт.		Принято управление интерфейсом
0	Неисп. ОУ		Неисправность ОУ

Примечание – В режиме КШ и М регистр доступен только на чтение и содержит ОС, передающего ОУ в транзакции ОУ-ОУ. В режиме ОУ регистр не используется.

7.22.11.9 INTEN

Регистр разрешения прерываний

Base ADDR=	0x400A_6000	Offset=	0x1020												
REG Name:	0x400A_8000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ERRIE	VALMESSIE	RFLAGNIE	IDLEIE

Бит	Имя	Значение	Описание
31..4			Зарезервировано
3	ERRIE		Прерывание при возникновении ошибки в сообщении. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при возникновении ошибок в сообщении
2	VALMESSIE		Прерывание при успешном завершении транзакции в канале. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при успешном завершении обмена данными в канале
1	RFLAGNIE		Прерывание при приеме достоверного слова. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при приеме достоверного ОС в режиме КШ или достоверного КС, ОС или слова данных команды управления в режиме ОУ
0	IDLEIE		Прерывание неактивности контроллера. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание по переходу контроллера в неактивное состояние

7.22.11.10 MSG

Регистр декодирования сообщений

Base ADDR=	0x400A_6000 0x400A_8000	Offset=	0x1024												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Код сообщения													

Бит	Имя	Значение	Описание
31...14	-		Зарезервировано
13...0	Код сообщения	0x0000	Регистр содержит код сообщения, полученного в режиме ОУ или М, и доступен только на чтение. В режиме КШ регистр не используется. Регистр обновляется каждый раз при получении нового достоверного КС

Таблица 145 – Коды сообщений регистра MSG

Код сообщения	CommandWord1				CommandWord2			
	15:11	10	9:5	4:0	15:11	10	9:5	4:0
<i>Команды обмена данными</i>								
0001 Команда приёма КШ-ОУ, не групповая	RTA	0	00001-11110	XXXXX				
0080 Команда приёма КШ-ОУ, групповая	11111	0	00001-11110	XXXXX				
0004 Команда приёма ОУ-ОУ, не групповая	RTA	0	00001-11110	XXXXX	XXXXX	1	00001-11110	XXXXX
0100 Команда приёма ОУ-ОУ, групповая	11111	0	00001-11110	XXXXX	не RTA	1	00001-11110	XXXXX
0402 Команда передачи ОУ-КШ	RTA	1	00001-11110	XXXXX				
1008 Команда передачи ОУ-ОУ, не групповая	не F	0	00001-11110	XXXXX	не F	0	00001-11110	XXXXX
0200 Команда передачи ОУ-ОУ, групповая	11111	0	00001-11110	XXXXX	RTA	1	00001-11110	XXXXX
<i>Команда управления</i>								
0410 Код 0-15 К=1 нет данных, не групповая	RTA	1	00000	11111	0XXXX			
0400 Код 0-15 К=1 нет данных, групповая	11111	1	00000	11111	0XXXX			
2420 Код 16-31 К=1 с данными, не групповая	RTA	1	00000	11111	1XXXX			
0040 Код 16-31 К=0 с данными, не групповая	RTA	0	00000	11111	1XXXX			
0800 Код 16-31 К=0 с данными, групповая	1111	0	00000	11111	1XXXX			

7.22.11.11 DATA

Память принимаемых/передаваемых данных

Base ADDR=		0x400A_6000 0x400A_8000				Offset=		0x0000- 0x0FFC								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0			Данные читаются из памяти или записываются в память в соответствии с подадресом (биты с 9 по 5) достоверного командного слова. Каждому подадресу соответствует 32x16 ячеек памяти на приём и 32x16 ячеек памяти на передачу. Общий объём памяти данных 2Кx16.

7.23 Контроллер АЦП (ADCx)

Микроконтроллер содержит два независимых блока АЦП. Каждый АЦП одновременно может осуществлять преобразование от одного из 64-х каналов. Всего в системе 18 внешних аналоговых каналов и 16 внутренних. Структурная схема АЦП представлена на рисунке. Распределение каналов по АЦП представлено в таблице.

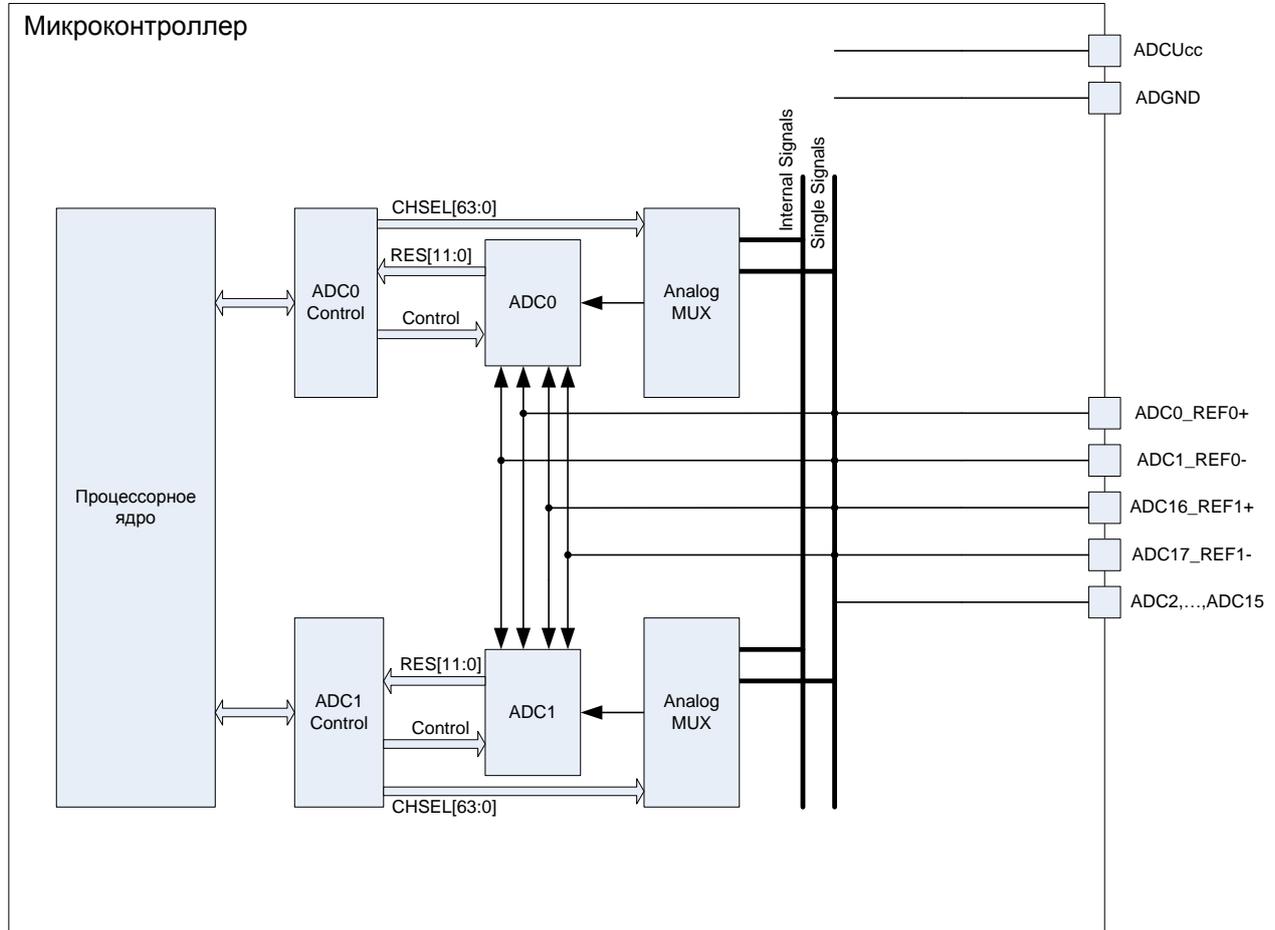


Рисунок 162 – Структурная схема контроллера АЦП

Таблица 146 – Распределение каналов по АЦП

Источник сигнала		ADC0	ADC1	Описание
Тип	Имя			
	ADC0_REF0+		Канал 0	Внешний вход АЦП/Внешнее опорное напряжение
	ADC1_REF0-		Канал 1	Внешний вход АЦП/Внешнее опорное напряжение
	ADC2	Канал 0	Канал 2	Внешний вход АЦП
	ADC3	Канал 1	Канал 3	Внешний вход АЦП
	ADC4	Канал 2	Канал 4	Внешний вход АЦП
	ADC5	Канал 3	Канал 5	Внешний вход АЦП
	ADC6	Канал 4	Канал 6	Внешний вход АЦП
	ADC7	Канал 5	Канал 7	Внешний вход АЦП
	ADC8	Канал 6	Канал 8	Внешний вход АЦП
	ADC9	Канал 7	Канал 9	Внешний вход АЦП
	ADC10	Канал 8	Канал 10	Внешний вход АЦП
	ADC11	Канал 9	Канал 11	Внешний вход АЦП
	ADC12	Канал 10	Канал 12	Внешний вход АЦП
	ADC13	Канал 11	Канал 13	Внешний вход АЦП

Источник сигнала		ADC0	ADC1	Описание
Тип	Имя			
	ADC14	Канал 12	Канал 14	Внешний вход АЦП
	ADC15	Канал 13	Канал 15	Внешний вход АЦП
	ADC16_REF1+	Канал 14		Внешний вход АЦП/Внешнее опорное напряжение
	ADC17_REF1-	Канал 15		Внешний вход АЦП/Внешнее опорное напряжение
	DUCC0	Канал 16		Напряжение питания домена 0
	DUCC1	Канал 17		Напряжение питания домена 1
	DUCC2	Канал 18		Напряжение питания домена 2
	DUCC3	Канал 19		Напряжение питания домена 3
	DAC0_OUT	Канал 20		Выход ЦАП 0
	DAC1_OUT	Канал 21		Выход ЦАП 1
	DUcc_B	Канал 22		Напряжение питания батарейного домена
	TEMP_0	Канал 23		Датчик температуры 0
	DUccPLL		Канал 16	Напряжение питания домена PLL
	CURRDUCC		Канал 17	Ток потребления батарейного домена
	CURRDUCC0		Канал 18	Ток потребления LDO0
	CURRDUCC1		Канал 19	Ток потребления LDO1
	CURRDUCC2		Канал 20	Ток потребления LDO2
	CURRDUCC3		Канал 21	Ток потребления LDO3
	CURRDUCCPLL		Канал 22	Ток потребления домена PLL
	TEMP_1		Канал 23	Датчик температуры 1

7.23.1 Преобразуемые сигналы

7.23.1.1 Внутренние сигналы

Каждый АЦП может преобразовать внутренние сигналы:

- напряжение питания DUcc0 (канал 16 АЦП 0);
- напряжение питания DUcc1 (канал 17 АЦП 0);
- напряжение питания DUcc2 (канал 18 АЦП 0);
- напряжение питания DUcc3 (канал 19 АЦП 0);
- выходное напряжения DAC0_OUT (канал 20 АЦП 0);
- выходное напряжения DAC1_OUT (канал 21 АЦП 0);
- напряжение питания DUcc_B (канал 22 АЦП 0);
- датчик температуры 0 (канал 23 АЦП 0);
- напряжение питания DUccPLL (канал 16 АЦП 1);
- ток потребления DUcc_B (канал 17 АЦП 1);
- ток потребления DUcc0 (канал 18 АЦП 1);
- ток потребления DUcc1 (канал 19 АЦП 1);
- ток потребления DUcc2 (канал 20 АЦП 1);
- ток потребления DUcc3 (канал 21 АЦП 1);
- ток потребления DUcc_PLL (канал 22 АЦП 1);
- датчик температуры 1 (канал 23 АЦП 1).

7.23.1.2 Внешние одиночные сигналы

Внешними сигналами АЦП могут являться до 18-ти выводов микросхемы (выводы ADC0_REF0+...ADC17_REF1-). При этом часть выводов может быть преобразована на обоих АЦП (выводы ADC2...ADC15), часть только на ADC0 (выводы ADC16_REF1+, ADC17_REF1-), и часть только на ADC1 (выводы ADC0_REF0+, ADC1_REF0-). Для проведения преобразования должен быть выбран номер канала, соответствующего выводу, и произведён запуск АЦП.

7.23.2 Источники опорного напряжения

Для проведения преобразования может быть использован следующий способ формирования опорного напряжения:

- REFMODE = 0

В качестве опорного напряжения преобразования выступают напряжение питания ADCUсс и общий ADCGND.

- REFMODE = 1

В качестве опорного напряжения преобразования выступают дополнительные сигналы, задаваемые битами REFSEL[5:0].

7.23.3 Синхронизация АЦП

Работа контроллера АЦП и само преобразование в блоке АЦП выполняется на тактовой частоте ADCCLK, формируемой в блоке CLKNTR. При этом запись и чтение регистров управления, регистров статуса и результатов осуществляется на частоте PCLK. В общем случае эти частоты могут быть асинхронными, таким образом, если необходимо обеспечить минимальный джиттер начала преобразования по какому-либо внешнему сигналу или программному запуску, необходимо учитывать соотношение рабочей частоты ADCCLK и PCLK.

7.23.4 Запуск преобразования

Преобразование АЦП могут быть начаты по следующим событиям:

- одиночное преобразование по записи бита GO;
- непрерывное преобразование по завершению предыдущего;
- одиночное преобразование по фронту сигнала EXT_GO. В качестве источников сигнала EXT_GO могут выступать:
 - события на внешних выводах;
 - события в таймерах;
 - сигналы ШИМ таймеров;
 - сигнал от второго АЦП.

Описание событий приведено в таблице ниже (Таблица 147).

Таблица 147 – События запуска преобразований

Бит EXT_GO_SEL	ADC0	Описание	ADC1	Описание
0	TMR0_EVNT	CNT==ARR в TMR0	TMR0_EVNT	CNT==ARR в TMR0
1	TMR1_EVNT	CNT==ARR в TMR1	TMR1_EVNT	CNT==ARR в TMR1
2	TMR2_EVNT	CNT==ARR в TMR2	TMR2_EVNT	CNT==ARR в TMR2
3	TMR3_EVNT	CNT==ARR в TMR3	TMR3_EVNT	CNT==ARR в TMR3
4	TMR4_EVNT	CNT==ARR в TMR4	TMR4_EVNT	CNT==ARR в TMR4
5	TMR5_EVNT	CNT==ARR в TMR5	TMR5_EVNT	CNT==ARR в TMR5
6	TMR0_CH0	Сигнал ШИМ канала 0 в TMR0	TMR0_CH2	Сигнал ШИМ канала 2 в TMR0
7	TMR1_CH0	Сигнал ШИМ канала 0 в TMR1	TMR1_CH2	Сигнал ШИМ канала 2 в TMR1
8	TMR2_CH0	Сигнал ШИМ канала 0 в TMR2	TMR2_CH2	Сигнал ШИМ канала 2 в TMR2
9	TMR3_CH0	Сигнал ШИМ канала 0 в TMR3	TMR3_CH2	Сигнал ШИМ канала 2 в TMR3
10	TMR4_CH0	Сигнал ШИМ канала 0 в TMR4	TMR4_CH2	Сигнал ШИМ канала 2 в TMR4
11	TMR5_CH0	Сигнал ШИМ канала 0 в TMR5	TMR5_CH2	Сигнал ШИМ канала 2 в TMR5
12	PORTA[0]	Сигнал на выводе PA[0]	PORTA[8]	Сигнал на выводе PA[8]
13	PORTA[16]	Сигнал на выводе PA[16]	PORTA[24]	Сигнал на выводе PA[24]
14	PORTB[0]	Сигнал на выводе PB[0]	PORTB[8]	Сигнал на выводе PB[8]
15	PORTB[16]	Сигнал на выводе PB[16]	PORTB[24]	Сигнал на выводе PB[24]
16	PORTC[0]	Сигнал на выводе PC[0]	PORTC[8]	Сигнал на выводе PC[8]
17	PORTC[16]	Сигнал на выводе PC[16]	PORTC[24]	Сигнал на выводе PC[24]
18	PORTD[0]	Сигнал на выводе PD[0]	PORTD[8]	Сигнал на выводе PD[8]
19	PORTD[16]	Сигнал на выводе PD[16]	PORTD[24]	Сигнал на выводе PD[24]
20	PORTE[0]	Сигнал на выводе PE[0]	PORTE[8]	Сигнал на выводе PE[8]
21	PORTE[16]	Сигнал на выводе PE[16]	PORTE[24]	Сигнал на выводе PE[24]
22	ADC0_IF	Окончание преобразования АЦП 0	ADC0_IF	Окончание преобразования АЦП 0
23	ADC1_IF	Окончание преобразования АЦП 1	ADC1_IF	Окончание преобразования АЦП 1

Примечание – для запуска преобразования по сигналу от внешних выводов необходимо, чтобы они были сконфигурированы в цифровой режим, при этом выполняемая функция не имеет значения (вывод может работать как на вход, так и на выход).

7.23.5 Последовательный опрос каналов АЦП

АЦП может осуществлять автоматическое переключение каналов при преобразовании:

- SELMODE = 0, автоматическое переключение не производится;
- SELMODE = 1, после завершения преобразования автоматически производится переключение на следующий канал (номера каналов, участвующих в автоматическом переключении, задаются в регистре AUTO_CH_SEL).

7.23.6 Получение результатов преобразования

Результаты преобразования АЦП сохраняются в выходном FIFO размером 16 записей. Каждая запись содержит в себе поля значения результата и поле номера канала преобразования. При наличии данных в FIFO-очереди формируется флаг наличия результатов ADC_NE, который может вызвать соответствующий обработчик прерывания либо опрошен программно. При переполнении FIFO-очереди формируется флаг ошибки ADC_OF. При этом происходит приостановка записи в FIFO. В регистрах RESULTCHx всегда отображается последнее значение, полученное при преобразовании соответствующего канала (не зависимо

от настройки контроля уровней). Сброс FIFO флагом FIFOCLR регистра CONFIG0 допустимо производить только при отключенной частоте ADCCLK.

7.23.7 Обработка результатов с помощью DMA

При необходимости результаты работы АЦП могут быть обработаны с помощью DMA. При использовании DMA не допускается программное считывание данных из FIFO. Блок АЦП формирует 5 различных запросов DMA с программируемой привязкой к каналам. Для запросов DMA_REQ0, DMA_REQ1, DMA_REQ2 и DMA_REQ3 задается режим работы с жесткой привязкой к конкретному номеру канала, DMA_REQ4 обрабатывает все остальные каналы. Не должна допускаться обработка одного канала двумя запросами, либо наличия канала без обработки.

7.23.8 Преобразование с контролем границ

Для обработки событий попадания уровня сигнала в заданную область можно применить режим работы с автоматическим контролем границ значения результата:

- контроль границ не выполняется, все результаты попадают в FIFO;
- контроль границ выполняется, и в FIFO попадают результаты, если результирующее значение более или равно MIN_LEVEL и менее или равно MAX_LEVEL;
- контроль границ выполняется, и в FIFO попадают результаты, если результирующее значение менее или равно MIN_LEVEL, или более или равно MAX_LEVEL.

7.23.9 Автоматическая калибровка АЦП

Блок АЦП имеет встроенный механизм автоматической калибровки. Для проведения автоматической калибровки необходимо выполнить преобразования для 60...63 каналов. Для полученных результатов вычислить среднее значение и записать в поле калибровки ADCTRIM регистра CONFIG1. Выполнение преобразования при калибровке занимает большее время, чем обычное преобразование.

7.23.10 Диаграмма работы АЦП

На рисунке представлена временная диаграмма работы АЦП. Длительность фаз pause и setup задается в тактах ADCCLK в соответствующих полях регистра CONFIG1. Любая из фаз или обе фазы могут быть пропущены записью 0 в соответствующее поле регистра CONFIG1. Фаза setup позволяет увеличить время заряда внутренней емкости АЦП. Минимальное время заряда внутренней емкости АЦП без использования фазы setup – 6 тактов ADCCLK. Следующие 36 тактов, пока АЦП выполняет преобразование, вход отключен от АЦП. Минимальный цикл АЦП в режиме непрерывного преобразования составляет 43 такта ADCCLK.

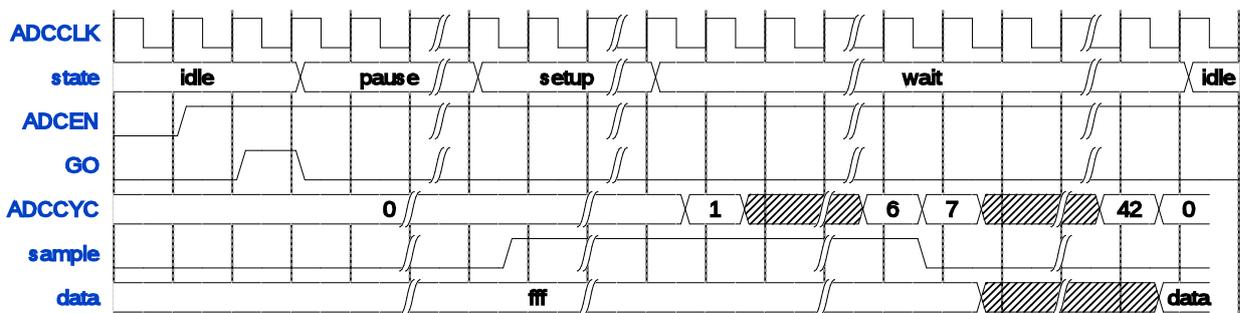


Рисунок 163 – Диаграмма работы АЦП

7.23.11 Схема входного каскада АЦП

На рисунке представлена эквивалентная схема входного каскада АЦП в режиме семплирования входа (соответствует активному уровню сигнала *sample* на рисунке). Значение параметров $R_0 < 4,5 \text{ кОм}$, $C_0 = 6,84 \text{ пФ}$ необходимо учитывать при выборе частоты $ADCCLK$ и количества циклов фазы *setup* для получения требуемой точности. Для достижения N-битной точности должно выполняться следующее соотношение

$$(R_{in} + R_0) * C_0 \leq \frac{t_{sample}}{\ln(2^N)}$$

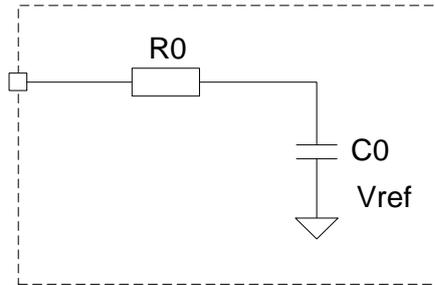


Рисунок 164 – Схема входного каскада АЦП в режиме семплирования входа

Например, для частоты семплирования $F_{sample} = 250 \text{ kSps}$ с нулевым количеством циклов фазы *setup* $t_{sample} = (0 + 6) * \frac{4_{\text{мкс}}}{43} = 558 \text{ нс}$. $F_{ADCCLK} = \frac{43}{4_{\text{мкс}}} = 10.75 \text{ МГц}$.

7.23.12 Описание регистров

Таблица 148 – Описание регистров контроллера АЦП

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x400A_A000		ADC0		
0x400A_B000		ADC1		
	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	CONFIG0		Регистр настройки работы контроллера АЦП
	0x0000_0008	CONFIG1		Регистр настройки работы контроллера АЦП
	0x0000_000C	CONFIG2		Регистр настройки работы контроллера АЦП
	0x0000_0010	CONTROL		Регистр управления контроллера АЦП
	0x0000_0014	STATUS		Регистр статуса контроллера АЦП, регистр недоступен до записи ключа
	0x0000_0018	RESULT		Регистр выхода FIFO результатов преобразования АЦП, регистр недоступен до записи ключа
	0x0000_001C	RESULTCH0		Регистр последнего значения, полученного по каналу 0, регистр недоступен до записи ключа
		...		
	0x0000_0118	RESULTCH63		Регистр последнего значения, полученного по каналу 63, регистр недоступен до записи ключа
	0x0000_011C	MAX_LEVEL		Регистр верхней границы преобразования
	0x0000_0120	MIN_LEVEL		Регистр нижней границы преобразования
	0x0000_0124	CHSEL0		Регистр задания каналов автоматического перебора
	0x0000_0128	CHSEL1		Регистр задания каналов автоматического перебора
	0x0000_012C	DMAREQ		Регистр задания запросов DMA
	0x0000_0130	STATRCH0		Статус приема данных с АЦП
	0x0000_0134	STATRCH1		Статус приема данных с АЦП
	0x0000_0138	FIFOEN0		Разрешение записи в FIFO
	0x0000_013C	FIFOEN1		Разрешение записи в FIFO

7.23.12.1 KEY

Base ADDR=	0x400A_A000	Offset=	0x0000_0000												
REG Name:	KEY														

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
RW-0															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока ADC_CNTR

7.23.12.2 CONFIG0

Регистр конфигурации 0

Base ADDR=	0x400A_A000	Offset=	0x0000_0004												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXT_GO_SEL[23:8]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXT_GO_SEL[7:0]								EXT_GO_INV	FIFOCLR	SELMODE	RH_MODE	REFMODE	WORKMODE [1:0]	ADCEN	
RW-0								RW-0	W-1	RW-0	RW-0	RW-0	RW-0	RW-0	

Бит	Имя	Значение	Описание
31:8	EXT_GO_SEL[23:0]	0x000000	Биты выбора источника запуска АЦП по внешнему событию EXT_GO 0000_0000_0000_0000_0000_0000 запрет всех источников (по умолчанию) 0000_0000_0000_0000_0000_0001 вывод EXT_GO[0] (ADCx_GO) 0000_0000_0000_0000_0000_0010 сигнал EXT_GO[1] (CH0o Timer0) 0000_0000_0000_0000_0000_0100 сигнал EXT_GO[2] (CH0no Timer0) Всего 24 источника
7	EXT_GO_INV	0	Управляемая инверсия сигнала EXT_GO 0 – сигнал не инвертируется (по умолчанию) 1 – сигнал инвертируется
6	FIFOCLR	0	Запись «1» в поле сбрасывает одновременно FIFO
5	SELMODE	0	Режим автоматического переключения каналов 0 – номер канала по завершению преобразования не переключается (по умолчанию) 1 – по завершению преобразования производится автоматическое переключение на следующий канал, заданный в CHSEL (по записи значения SELMODE=1 предустанавливается счетчик перебора каналов)
4	RH_MODE	0	Выбор режима задания опорного тока АЦП 0 – опорный ток от источника REF25 1 – опорный ток от напряжения питания АЦП
3	REFMODE	0	Выбор источника опорного напряжения для преобразования 0 – REF+ и REF- задаются напряжением питания АЦП (ADCUcc и ADCGND) 1 – REF+ и REF- задаются отдельными сигналами, задаваемыми битами REFSEL
2:1	WORKMODE[1:0]	00	Режим работы АЦП 00 – запуск преобразования по записи бита GO (регистр CONTROL.GO) (по умолчанию) 01 – запуск преобразования по фронту сигнала EXT_GO 10 – Непрерывное преобразование
0	ADCEN	0	Включение АЦП: 0 – АЦП выключен (по умолчанию) 1 – АЦП включен (по записи значения ADCEN=1 предустанавливается счетчик перебора каналов)

7.23.12.3 CONFIG1

Регистр конфигурации 1

Base ADDR=		0x400A_A000				Offset=		0x0000_0008									
		0x400A_B000															
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
ADCTRIM[11:0]												PAUSE[7:4]					
RW-0												RW-0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAUSE[3:0]				SETUP[8:0]								REFTRIM[2:0]			
RW-0				RW-0								RW-0			

Бит	Имя	Значение	Описание
31:20	ADCTRIM[11:0]	0x000	Регистр подстройки АЦП после проведения автокалибровки
19:12	PAUSE[7:0]	0	Время паузы перед началом процесса преобразования в тактах ADCCLK: 0 – нет паузы; 1 – пауза 1 ADCCLK; ... 255 – пауза 255ADCCLK;
11:3	SETUP[8:0]	0	Время «зарядки» АЦП в тактах ADCCLK: 0 – нет паузы; 1 – пауза 1 ADCCLK; ... 511 – пауза 511ADCCLK;
2:0	REFTRIM[2:0]	000	Подстройка источника опорного напряжения refADC 000 – типовое значение 001... 011 – увеличение опорного напряжения 100... 111 – уменьшение опорного напряжения

7.23.12.4 CONFIG2

Регистр конфигурации 2

Base ADDR=	0x400A_A000	Offset=	0x0000_000C													
REG Name:	0x400A_B000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-								DT_MODE	REF_TRIMR[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								REFSEL[7:0]	LEVLCTRL[1:0]		IE_ERFIN	IE_AF	IE_NAE	IE_OF	IE_NE
RW-0								RW-0			RW-0	RW-0	RW-0	RW-0	RW-0

Бит	Имя	Значение	Описание
31:25		-	Зарезервировано
25	SELR	0	Выбор резистора источника опорного тока 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен) 1 – используется внешний стабильный резистор на выводе ADC_EXTR на землю.
24	DT_MODE	0	Режим работы датчика температуры 0 – максимальный ток (время установки 5 мкс) 1 – минимальный ток (время установки 30 мкс)
23:16	REF_TRIMR [7:0]	0x00	Подстройка внутреннего резистора источника опорного тока 1000_0000 Типовое значение опорного тока (10 мкА) 0000_0000 Минимальный опорный ток (5 мкА) 1111_1111 Максимальный опорный ток (15 мкА) <i>Для функционирования должно быть задано типовое значение</i>
15:14	-	-	Зарезервировано
13:8	REFSEL[5:0]	000000	Биты выбора источников опорного напряжения при REFMODE = 1 REFSEL[0] = 1 – REF+ = VREF25 внутреннего источника опорного напряжения REFSEL[1] = 1 – REF+ = ADC16_REF1+ для ADC0, ADC0_REF0+ для ADC1 REFSEL[2] = 1 – REF+ = ADC0_REF0+ для ADC0, ADC16_REF1+ для ADC1 REFSEL[3] = 1 – REF- = GND внутреннего источника опорного напряжения REFSEL[4] = 1 – REF- = ADC17_REF1- для ADC0, ADC1_REF0- для ADC1 REFSEL[5] = 1 – REF- = ADC1_REF0- для ADC0, ADC17_REF1- для ADC1
7:6	LEVLCTRL[1:0]	00	Управление заданием контроля границ 00 – нет контроля границ (по умолчанию) 01 – результаты попадают в FIFO, если результирующее значение более или равно MIN_LEVEL и менее или равно MAX_LEVEL 10 – результаты попадают в FIFO, если результирующее значение менее или равно MIN_LEVEL, или более или равно MAX_LEVEL 11 – зарезервировано
5		-	Зарезервировано

4	IE_ERFIN	0	Флаг разрешения запроса прерывания по наличию ошибки в интерфейсе с АЦП: 0 – прерывание запрещено (по умолчанию) 1 – прерывание разрешено
3	IE_AF	0	Флаг разрешения запроса прерывания по событию ADC_AF (AlmostFull): 0 – прерывание запрещено (по умолчанию) 1 – прерывание разрешено
2	IE_NAE	0	Флаг разрешения запроса прерывания по событию ADC_NAE (NotAlmostEmpty): 0 – прерывание запрещено (по умолчанию) 1 – прерывание разрешено
1	IE_OF	0	Флаг разрешения запроса прерывания по событию ADC_OF (OverFlow): 0 – прерывание запрещено (по умолчанию) 1 – прерывание разрешено
0	IE_NE	0	Флаг разрешения запроса прерывания по событию ADC_NE (NotEmpty): 0 – прерывание запрещено (по умолчанию) 1 – прерывание разрешено

7.23.12.5 CONTROL

Регистр управления

Base ADDR=	0x400A_A000	Offset=	0x0000_0010												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		CHANNEL[5:0]													GO
		RW-0													W-1

Бит	Имя	Значение	Описание
31:14		-	Зарезервировано
13:8	CHANNEL[5:0]	000000	Номер канала, выбранного для преобразования (используется от 0 до 63)
7:1		-	Зарезервировано
0	GO	0	Бит запуска начала преобразования Запись 1 начинает процесс преобразования, запись 1 в процессе преобразования игнорируется, запись 0 не имеет эффекта

7.23.12.6 STATUS

Регистр статуса

Base ADDR=	0x400A_A000	Offset=	0x0000_0014												
REG Name:	0x400A_B000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		RTCHVAL										ADC_ERFIN	ADC_AF	ADC_NAE	ADC_OF	ADC_NE
		R-0										R-0	R-0	R-0	R-0	

Бит	Имя	Значение	Описание
13:8	RTCHVAL		Номер самого младшего, не обработанного канала по чтению регистров RESULTCHx, если все каналы обработаны, то сохраняется последнее значение, по умолчанию 0
4	ADC_ERFIN		Флаг наличия ошибки в интерфейсе с АЦП: 0 – без ошибки 1 – Ошибка интерфейса с АЦП – сигнал FIN менее или более 39 тактов ADC_CLK. Флаг сбрасывается записью 1 и при CONFIG0.ADCEN=0
3	ADC_AF		Флаг наличия данных на выходе FIFO (AlmostFull): 0 – в FIFO 4 и более свободных записей 1 – в FIFO менее 4 свободных записей Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) и уменьшения данных в FIFO менее 13
2	ADC_NAE		Флаг наличия данных на выходе FIFO (NotAlmostEmpty): 0 – в FIFO менее 4 записей 1 – в FIFO 4 и более записей Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) и уменьшения данных в FIFO менее 4
1	ADC_OF		Флаг переполнения FIFO 0 – нет переполнения FIFO 1 – было переполнение FIFO Флаг сбрасывается записью 1 и при CONFIG0.ADCEN=0
0	ADC_NE		Флаг наличия данных в FIFO 0 – FIFO пусто 1 – FIFO содержит данные Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) всех данных

7.23.12.7 RESULT

Регистр выхода FIFO результата

Base ADDR=		0x400A_A000				Offset=		0x0000_0018							
		0x400A_B000													
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										CHANNEL[5:0]					
										R-0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RESULT[11:0]											
				R-0											

Бит	Имя	Значение	Описание
21:16	CHANNEL[5:0]		Номер канала, по которому получены результаты
11:0	RESULT[11:0]		Результат преобразования

Примечание – Размер FIFO составляет 16 записей.

7.23.12.8 RESULTCHx

Регистр последнего полученного значения по “X” каналу

Base ADDR=		0x400A_A000				Offset=		0x0000_001C - 0x0000_0118							
		0x400A_B000													
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										CHANNEL[5:0]					
										R-x					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RESULT[11:0]											
				R-0											

Бит	Имя	Значение	Описание
21:16	CHANNEL[5:0]		Номер канала Для RESULTCH0 всегда 0 Для RESULTCH1 всегда 1 ... Для RESULTCH63 всегда 63
11:0	RESULT[11:0]		Результат преобразования

7.23.12.9 MAX_LEVEL

Регистр верхней границы

Base ADDR=	0x400A_A000	Offset=	0x0000_011C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				MAX_LEVEL[11:0]											
				RW-0											

Бит	Имя	Значение	Описание
11:0	MAX_LEVEL[11:0]		Верхняя граница при контроле границ преобразования

7.23.12.10 MIN_LEVEL

Регистр нижней границы

Base ADDR=	0x400A_A000	Offset=	0x0000_0120												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				MIN_LEVEL[11:0]											
				RW-0											

Бит	Имя	Значение	Описание
11:0	MIN_LEVEL[11:0]		Нижняя граница при контроле границ преобразования

7.23.12.11 CHSEL0

Регистр каналов автоперебора 0

Base ADDR=	0x400A_A000	Offset=	0x0000_0124												
REG Name:															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHSEL[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL[15:0]															
RW-0															

Бит	Имя	Значение	Описание
31:0	CHSEL[31:0]		Унарный код задания каналов, участвующих в автоматическом переключении каналов 0 – канал не участвует 1 – канал участвует Номер бита соответствует каналу 0 – 0 каналу ... 31 – 31 каналу

7.23.12.12 CHSEL1

Регистр каналов автоперебора 1

Base ADDR=	0x400A_A000	Offset=	0x0000_0128												
	0x400A_B000														
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHSEL[63:48]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL[47:32]															
RW-0															

Бит	Имя	Значение	Описание
31:0	CHSEL[63:32]		Унарный код задания каналов, участвующих в автоматическом переключении каналов 0 – канал не участвует 1 – канал участвует Номер бита соответствует каналу 0 – 32 каналу ... 31 – 63 каналу

7.23.12.13 DMAREQ

Регистр управления запросами DMA

Base ADDR=	0x400A_A000	Offset=	0x0000_012C												
	0x400A_B000														
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	DMAREQEN[6:0]	DMAREQ3[5:0]	DMAREQ2[5:4]
	RW-0	RW-0	RW-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAREQ2[3:0]				DMAREQ1[5:0]					DMAREQ0[5:0]						
RW-0				RW-0					RW-0						

Бит	Имя	Значение	Описание
30:24	DMAREQEN[6:0]		Бит разрешения запросов ADC_DMA_REQ 0 – запрос DMA запрещен 1 – запрос DMA разрешен Бит 0 - ADC_DMA_SREQ0 Бит 1 - ADC_DMA_SREQ1 Бит 2 - ADC_DMA_SREQ2 Бит 3 - ADC_DMA_SREQ3 Бит 4 - ADC_DMA_SREQ4 Бит 5 - ADC_DMA_SREQ5 Бит 6 - ADC_DMA_REQ5 Запрос ADC_DMA_SREQ4 формируется, если он разрешен и есть данные, но нет ни одного, настроенного на данный канал, другого запроса DMA Запрос ADC_DMA_SREQ5 формируется, если он разрешен и есть хотя бы одно данное в FIFO – условие FIFO_NEMPTY Запрос ADC_DMA_REQ5 формируется, если он разрешен и есть четыре и более данных в FIFO – условие FIFO_NAEMPTY
23:18	DMASREQ3[5:0]		Номер канала вызывающий запрос ADC_DMA_REQ3 0 – канал 0 1 – канал 1 ... 63 – канал 63
17:12	DMASREQ2[5:0]		Номер канала вызывающий запрос ADC_DMA_REQ2 0 – канал 0 1 – канал 1 ... 63 – канал 63
11:6	DMASREQ1[5:0]		Номер канала вызывающий запрос ADC_DMA_REQ1 0 – канал 0 1 – канал 1 ... 63 – канал 63
5:0	DMASREQ0[5:0]		Номер канала вызывающий запрос ADC_DMA_REQ0 0 – канал 0 1 – канал 1 ... 63 – канал 63

7.23.12.14 STATRCH0

Регистр статуса по приему данных с АЦП

Base ADDR=	0x400A_A000	Offset=	0x0000_0130						
	0x400A_B000								

REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STRCH [31:16]															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRCH [15:0]															
R-0															

Бит	Имя	Значение	Описание
31:0	STRCH[31:0]		Статус приема данных с АЦП: 0 – данных нет 1 – есть новые данные в соответствующем регистре RESULTCHх Сброс происходит при чтении соответствующего регистра RESULTCHх, причем приход новых данных более приоритетен, чем сброс по чтению. Номер бита соответствует каналу 0 – 0 каналу ... 31 – 31 каналу

7.23.12.15 STATRCH1

Регистр статуса по приему данных с АЦП

Base ADDR=	0x400A_A000	Offset=	0x0000_0134												
REG Name:	0x400A_B000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STRCH [63:48]															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRCH [47:32]															
R-0															

Бит	Имя	Значение	Описание
31:0	STRCH[63:32]		Статус приема данных с АЦП: 0 – данных нет 1 – есть новые данные в соответствующем регистре RESULTCHx Сброс происходит при чтении соответствующего регистра RESULTCHx, причем приход новых данных более приоритетен, чем сброс по чтению. Номер бита соответствует каналу 0 – 32 каналу ... 31 – 63 каналу

7.23.12.16 FIFOEN0

Регистр разрешения записи результата из АЦП в FIFO

Base ADDR=	0x400A_A000	Offset=	0x0000_0138												
REG Name:	0x400A_B000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOEN[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOEN[15:0]															
RW-0															

Бит	Имя	Значение	Описание
31:0	FIFOEN[31:0]		Разрешение записи в FIFO 0 – канал не участвует 1 – канал участвует Номер бита соответствует каналу 0 – 0 каналу ... 31 – 31 каналу

7.23.12.17 FIFOEN1

Регистр разрешения записи данных из АЦП в FIFO

Base ADDR=	0x400A_A000	Offset=	0x0000_013C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOEN[63:48]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOEN[47:32]															
RW-0															

Бит	Имя	Значение	Описание
31:0	FIFOEN[63:32]		Разрешение записи в FIFO 0 – канал не участвует 1 – канал участвует Номер бита соответствует каналу 0 – 32 каналу ... 31 – 63каналу

7.24 Контроллер ЦАП (DACx)

Микроконтроллер содержит два встроенных ЦАП. Структура ЦАП представлена на рисунке ниже (Рисунок 165).

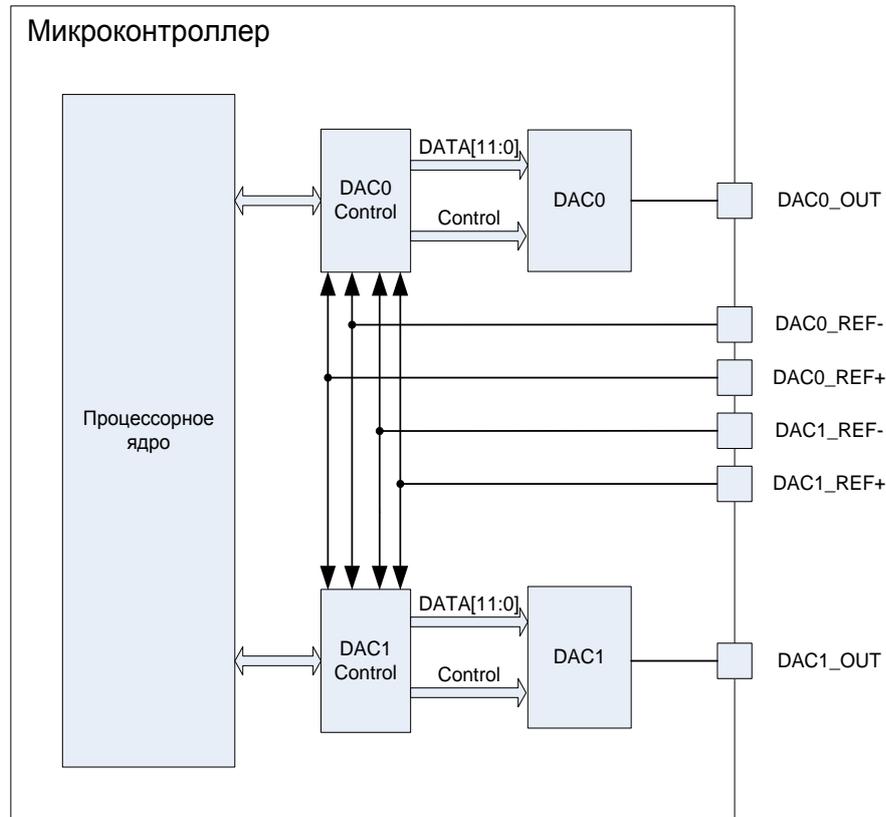


Рисунок 165 – Структурная схема контроллера ЦАП

7.24.1 Источники опорного напряжения

В качестве опорного напряжения ЦАП могут выступать:

- REFMODE = 00 опорное напряжение формируется напряжением питания ЦАП (AUcc0 и AGND0 для DAC0 и AUcc1и AGND1для DAC1);
- REFMODE = 01 опорное напряжение задается с внешних выходов (DAC0_REF-и DAC0_REF+для DAC0и DAC1_REF-и DAC1_REF+для DAC1);
- REFMODE = 10 опорное напряжение задается с внешних выходов (DAC1_REF-и DAC1_REF+для DAC0и DAC0_REF-и DAC0_REF+для DAC1);
- REFMODE = 11 опорное напряжение формируется встроенным источником опорного напряжения (refDAC0 для DAC0 и refDAC1 для DAC1). Типовое значение 2,5 В можно подстроить с помощью поля REFTRIM регистра CONFIG1.

7.24.2 Синхронизация ЦАП

Обновление значения на выходе DAC происходит на частоте DAC_CLK. При этом могут быть выбраны следующие режимы:

- непрерывное обновление, если в FIFO есть новые данные, то они выставляются на выход на следующий такт DAC_CLK после выставления предыдущих. Если новых данных нет, то на выходе удерживается ранее выставленное значение.
- обновление по событию, на выход DAC данные поступают из FIFO, если они там есть по фронту сигнала EXT_GO. Если есть фронт сигнала EXT_GO, но в FIFO нет новых данных, то на выходе удерживается ранее выставленное значение.

Таблица 149 – Сигналы запуска преобразования ЦАП

Бит EXT_GO_ SEL	DAC0	Описание	DAC1	Описание
0	TMR0_EVNT	CNT==ARR в TMR0	TMR0_EVNT	CNT==ARR в TMR0
1	TMR1_EVNT	CNT==ARR в TMR1	TMR1_EVNT	CNT==ARR в TMR1
2	TMR2_EVNT	CNT==ARR в TMR2	TMR2_EVNT	CNT==ARR в TMR2
3	TMR3_EVNT	CNT==ARR в TMR3	TMR3_EVNT	CNT==ARR в TMR3
4	TMR4_EVNT	CNT==ARR в TMR4	TMR4_EVNT	CNT==ARR в TMR4
5	TMR5_EVNT	CNT==ARR в TMR5	TMR5_EVNT	CNT==ARR в TMR5
6	TMR0_CH0	Сигнал ШИМ канала 0 в TMR0	TMR0_CH2	Сигнал ШИМ канала 2 в TMR0
7	TMR1_CH0	Сигнал ШИМ канала 0 в TMR1	TMR1_CH2	Сигнал ШИМ канала 2 в TMR1
8	TMR2_CH0	Сигнал ШИМ канала 0 в TMR2	TMR2_CH2	Сигнал ШИМ канала 2 в TMR2
9	TMR3_CH0	Сигнал ШИМ канала 0 в TMR3	TMR3_CH2	Сигнал ШИМ канала 2 в TMR3
10	TMR4_CH0	Сигнал ШИМ канала 0 в TMR4	TMR4_CH2	Сигнал ШИМ канала 2 в TMR4
11	TMR5_CH0	Сигнал ШИМ канала 0 в TMR5	TMR5_CH2	Сигнал ШИМ канала 2 в TMR5
12	PORTA[0]	Сигнал на выводе PA[0]	PORTA[8]	Сигнал на выводе PA[8]
13	PORTA[16]	Сигнал на выводе PA[16]	PORTA[24]	Сигнал на выводе PA[24]
14	PORTB[0]	Сигнал на выводе PB[0]	PORTB[8]	Сигнал на выводе PB[8]
15	PORTB[16]	Сигнал на выводе PB[16]	PORTB[24]	Сигнал на выводе PB[24]
16	PORTC[0]	Сигнал на выводе PC[0]	PORTC[8]	Сигнал на выводе PC[8]
17	PORTC[16]	Сигнал на выводе PC[16]	PORTC[24]	Сигнал на выводе PC[24]
18	PORTD[0]	Сигнал на выводе PD[0]	PORTD[8]	Сигнал на выводе PD[8]
19	PORTD[16]	Сигнал на выводе PD[16]	PORTD[24]	Сигнал на выводе PD[24]
20	PORTE[0]	Сигнал на выводе PE[0]	PORTE[8]	Сигнал на выводе PE[8]
21	PORTE[16]	Сигнал на выводе PE[16]	PORTE[24]	Сигнал на выводе PE[24]
22	ADC0_IF	Окончание преобразования АЦП 0	ADC0_IF	Окончание преобразования АЦП 0
23	ADC1_IF	Окончание преобразования АЦП 1	ADC1_IF	Окончание преобразования АЦП 1

7.24.3 Очередь данных

На входе по данным контроллера DAC установлено FIFO размером 16 записей. При переполнении FIFO новые записи игнорируются. Для работы с FIFO формируются сигналы DAC_EMPTY (FIFO пустое), DAC_FULL (FIFO полное), DAC_AEMPTY (в FIFO только 4 или менее записей) и DAC_AFULL (в FIFO только 4 или более свободных записей). Кроме того, контроллер DAC формирует запросы DMA (одиночные и множественные – 4 записи). При переполнении FIFO возникает событие DAC_OF.

7.24.4 Описание регистров

Таблица 150 – Описание регистров контроллера ЦАП

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x400A_C000		DAC0		
0x400A_D000		DAC1		
	0x000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x004	CONFIG0		Регистр настройки работы контроллера ЦАП
	0x008	CONFIG1		Регистр настройки работы контроллера ЦАП
	0x00C	DATA		Регистр данных ЦАП, регистр недоступен до записи ключа
	0x010	STATUS		Регистр состояние DAC
	0x014	IMSK		Регистр маскирования прерываний
	0x018	DMAREQ		Регистр управления запросами DMA

7.24.4.1 KEY

Base ADDR=	0x400A_C000 0x400A_D000	Offset=	0x000												
REG Name:	KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Бит	Имя	Значение	Описание
31...0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока DAC контроллера

7.24.4.2 CONFIG0

Регистр конфигурации 0

Base ADDR=	0x400A_C000	Offset=	0x004												
REG Name:	0x400A_D000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXT_GO_SEL[23:8]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXT_GO_SEL[7:0]								EXT_GO_INV			FIFOCLR	REFMODE [1:0]	SMODE	DACEN	

Бит	Имя	Значение	Описание
-	-		Зарезервировано
31...8	EXT_GO_SEL[23:0]		Биты выбора источника запуска ЦАП по внешнему событию EXT_GO 0000_0000_0000_0000_0000_0000 запрет всех источников (по умолчанию) 0000_0000_0000_0000_0000_0001 вывод EXT_GO[0] (DACx_GO) 0000_0000_0000_0000_0000_0010 сигнал EXT_GO[1] (CH0o Timer0) 0000_0000_0000_0000_0000_0100 сигнал EXT_GO[2] (CH0no Timer0) Всего 24 источника
7	EXT_GO_INV		Управляемая инверсия сигнала EXT_GO 0 – сигнал не инвертируется (по умолчанию) 1 – сигнал инвертируется
4	FIFOCLR		Запись «1» в поле сбрасывает одновременно FIFO, по чтению всегда 0
3, 2	REFMODE[1:0]		Выбор источника опорного напряжения для преобразования: - REFMODE = 00 опорное напряжение формируется напряжением питания ЦАП (AUcc0 и AGND0 для DAC0; AUcc1 и AGND1 для DAC1) (по умолчанию); - REFMODE = 01 опорное напряжение задается с внешних выходов (DAC0_REF- и DAC0_REF+ для DAC0 и DAC1_REF- и DAC1_REF+ для DAC1); - REFMODE = 10 опорное напряжение задается с внешних выходов (DAC1_REF- и DAC1_REF+ для DAC0 и DAC0_REF- и DAC0_REF+ для DAC1); - REFMODE = 11 опорное напряжение формируется встроенным источником опорного напряжения (refDAC0 для DAC0 и refDAC1 для DAC1)
1	SMODE		Режим работы ЦАП 0 – обновление по фронту EXT_GO при наличии данных в FIFO (по умолчанию) 1 – непрерывное обновление при наличии данных в FIFO по DAC_CLK
0	DACEN		Включение ЦАП 0 – ЦАП выключен (по умолчанию) 1 – ЦАП включен

7.24.4.3 CONFIG1

Регистр конфигурации 1

Base ADDR=		0x400A_C000				Offset=		0x008							
		0x400A_D000													
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFTRIMR[7:0]											REFTRIM [2:0]				

Бит	Имя	Значение	Описание
31..16	-	-	-
15...8	REFTRIMR[7:0]		Подстройка внутреннего резистора источника опорного тока 1000_0000 Типовое значение опорного тока (10 мкА) 0000_0000 Минимальный опорный ток (5 мкА) 1111_1111 Максимальный опорный ток (15 мкА) <i>Для функционирования должно быть задано типовое значение</i>
7	SELR		Выбор резистора источника опорного тока 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен) 1 – используется внешний стабильный резистор на выводе DAC_EXTR на землю
6..3	-	-	-
2...0	REFTRIM[2:0]		Подстройка источника опорного напряжения refDAC 000 – типовое значение (<i>по умолчанию</i>) 001... 011 – увеличение опорного напряжения 100... 111 – уменьшение опорного напряжения Значение подстройки определяется в процессе эксплуатации

7.24.4.4 DATA

Регистр данных ЦАП

Base ADDR=		0x400A_C000				Offset=		0x00C							
		0x400A_D000													
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DATA[11:0]											

Бит	Имя	Значение	Описание
31...12	-		Зарезервировано
11...0	DATA[11:0]		Регистр входных данных FIFO ЦАП При записи, если в FIFO есть свободное место, то записываются данные в FIFO. Если в FIFO нет свободного места, запись игнорируется и взводится флаг переполнения FIFODAC_OF При чтении читается последнее, корректно записанное значение

7.24.4.5 STATUS

Регистр состояния ЦАП

Base ADDR=	0x400A_C000		Offset=	0x010											
REG Name:	0x400A_D000														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DAC_OF	DAC_AFULL	DAC_FULL	DAC_AEMPTY	DAC_EMPTY

Бит	Имя	Значение	Описание
4	DAC_OF		Флаг возникновения переполнения FIFO 0 – нет факта переполнения FIFO 1 – было переполнения FIFO (запись в DATA при DAC_FULL=1) Очищается записью 1.
3	DAC_AFULL		FIFO частично заполнено 0 – в FIFO 4 и более свободных записей 1 – в FIFO менее 4 свободных записей
2	DAC_FULL		FIFO полное 0 – FIFO не заполнено, есть хотя бы одна свободная запись 1 – FIFO заполнено
1	DAC_AEMPTY		FIFO частично пустое 0 – в FIFO 4 и более записей 1 – в FIFO менее 4 записей
0	DAC_EMPTY		FIFO пустое 0 – FIFO не пусто, есть хотя бы одна запись 1 – FIFO пустое

7.24.4.6 IMSK

Регистр маскирования прерываний

Base ADDR=	0x400A_C000		Offset=	0x014											
	0x400A_D000														
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											IE_OF	IE_AFULL	IE_FULL	IE_AEMPTY	IE_EMPTY

Бит	Имя	Значение	Описание
4	IE_OF		Флаг разрешения запроса прерывания по событию DAC_OF 0 – прерывание запрещено (<i>по умолчанию</i>) 1 – прерывание разрешено
3	IE_AFULL		Флаг разрешения запроса прерывания по событию DAC_AFULL 0 – прерывание запрещено (<i>по умолчанию</i>) 1 – прерывание разрешено
2	IE_FULL		Флаг разрешения запроса прерывания по событию DAC_FULL 0 – прерывание запрещено (<i>по умолчанию</i>) 1 – прерывание разрешено
1	IE_AEMPTY		Флаг разрешения запроса прерывания по событию DAC_AEMPTY 0 – прерывание запрещено (<i>по умолчанию</i>) 1 – прерывание разрешено
0	IE_EMPTY		Флаг разрешения запроса прерывания по событию DAC_EMPTY 0 – прерывание запрещено (<i>по умолчанию</i>) 1 – прерывание разрешено

7.25 Векторы прерываний и регистры управления ядра

7.25.1 Распределение векторов прерываний

Таблица 151 – Распределение векторов прерываний

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
		0x0000	Initial SP value	
1		0x0004	Reset	
2	-14	0x0008	NMI	Зарезервировано
3	-13	0x000C	Hard fault	
4	-12	0x0010	Memory management fault	
5	-11	0x0014	Bus fault	
6	-10	0x0018	Usage fault	
7	-9	0x001C	Reserved	
8	-8	0x0020	Reserved	
9	-7	0x0024	Reserved	
10	-6	0x0028	Reserved	
11	-5	0x002C	SVCall	
12	-4	0x0030	Reserved for Debug	
13	-3	0x0034	Reserved	
14	-2	0x0038	PendSV	
15	-1	0x003C	Systick	
16	0	0x0040	Reserved	Зарезервировано
17	1	0x0044	Reserved	Зарезервировано
18-46	2-30	0x0048 – 0x00B8	...	Зарезервировано
47	31	0x00BC	Reserved	Зарезервировано
48	32	0x00C0	FT_IF0	Прерывание от событий EVENT0-EVENT4 FTCNTR при разрешенном сбросе по этим событиям
49	33	0x00C4	FT_IF1	Прерывание от событий EVENT0-EVENT4 FTCNTR, независящее от маски сброса
50	34	0x00C8	FT_IF2	Прерывание от событий EVENT5-EVENT8 FTCNTR (маскируемое)
51	35	0x00CC	FT_IF3	Прерывание от событий EVENT9-EVENT12 FTCNTR (маскируемое)
52	36	0x00D0	CLK_IF	Прерывание контроллера тактовых частот
53	37	0x00D4	PVD_IF	Прерывание от блока детектора напряжений питания
54	38	0x00D8	RTC_IF	Прерывание от блока часов реального времени
55	39	0x00DC	BKP_IF	Прерывание от контроллера батарейного домена

* Нумерация прерываний соответствует стандартной библиотеке CMSIS. Отрицательные номера используются для индексации регистров в SCS для управления системными прерываниями, неотрицательные номера используются для индексации регистров NVIC для управления внешними прерываниями.

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
56	40	0x00E0	EXT_INTERRUPT0	Прерывание контроллера EXT_BUS для региона 0
57	41	0x00E4	EXT_INTERRUPT1	Прерывание контроллера EXT_BUS для региона 1
58	42	0x00E8	EXT_INTERRUPT2	Прерывание контроллера EXT_BUS для региона 2
59	43	0x00EC	EXT_INTERRUPT3	Прерывание контроллера EXT_BUS для региона 3
60	44	0x00F0	EXT_INTERRUPT4	Прерывание контроллера EXT_BUS для региона 4
61	45	0x00F4	EXT_INTERRUPT5	Прерывание контроллера EXT_BUS для региона 5
62	46	0x00F8	EXT_INTERRUPT6	Прерывание контроллера EXT_BUS для региона 6
63	47	0x00FC	EXT_INTERRUPT7	Прерывание контроллера EXT_BUS для региона 7
64	48	0x0100	-	
65	49	0x0104	-	
66	50	0x0108	DMA_ERR	Прерывание от контроллера DMA по ошибке на шине
67	51	0x010C	DMA_DONE0	Прерывание от контроллера DMA по завершению цикла для канала 0
68	52	0x0110	DMA_DONE1	Прерывание от контроллера DMA по завершению цикла для канала 1
69-97	53-81	0x0114 – 0x0184	...	Прерывание от контроллера DMA по завершению цикла для канала 2-30
98	82	0x0188	DMA_DONE31	Прерывание от контроллера DMA по завершению цикла для канала 31
99	83	0x018C	IRQ_PORTA	Прерывание от контроллера портов ввода/вывода PORTA
100	84	0x0190	IRQ_PORTB	Прерывание от контроллера портов ввода/вывода PORTB
101	85	0x0194	IRQ_PORTC	Прерывание от контроллера портов ввода/вывода PORTC
102	86	0x0198	IRQ_PORTD	Прерывание от контроллера портов ввода/вывода PORTD
103	87	0x019C	IRQ_PORTE	Прерывание от контроллера портов ввода/вывода PORTE
104	88	0x01A0	-	
105	89	0x01A4	INT_ETH0	Прерывание от контроллера Ethernet
106	90	0x01A8	-	
107	91	0x01AC	INT_SPW0	Прерывание от контроллера SpaceWire
108	92	0x01B0	-	
109	93	0x01B4	INT_TMR0	Прерывание от контроллера таймера 0
110	94	0x01B8	INT_TMR1	Прерывание от контроллера таймера 1
111	95	0x01BC	INT_TMR2	Прерывание от контроллера таймера 2
112	96	0x01C0	INT_TMR3	Прерывание от контроллера таймера 3

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
113	97	0x01C4	-	
114	98	0x01C8	-	
115	99	0x01CC	INT_CAN0	Прерывание от контроллера CAN
116	100	0x01D0	-	
117	101	0x01D4	-	
118	102	0x01D8	-	
119	103	0x01DC	-	
120	104	0x01E0	INT_SSP0	Прерывание от контроллера SSP
121	105	0x01E4	-	
122	106	0x01E8	-	
123	107	0x01EC	-	
124	108	0x01F0	INT_UART0	Прерывание от контроллера UART0
125	109	0x01F4	INT_UART1	Прерывание от контроллера UART1
126	110	0x01F8	-	
127	111	0x01FC	-	
128	112	0x0200	-	
129	113	0x0204	INT_RX_ARC0	Прерывание от контроллера приёмника ARINC
130	114	0x0208	INT_TX_ARC0	Прерывание от контроллера передатчика ARINC
131	115	0x020C	-	
132	116	0x0210	-	
133	117	0x0214	INT_MIL0	Прерывание от контроллера МКПД 0
134	118	0x0218	INT_MIL1	Прерывание от контроллера МКПД 1
135	119	0x021C	INT_ADC0	Прерывание от контроллера АЦП 0
136	120	0x0220	INT_ADC1	Прерывание от контроллера АЦП 1
137	121	0x0224	INT_DAC0	Прерывание от контроллера ЦАП 0
138	122	0x0228	INT_DAC1	Прерывание от контроллера ЦАП 1
139	123	0x022C	-	
140	124	0x0230	-	
141	125	0x0234	-	
142	126	0x0238	-	
143	127	0x023C	-	

7.25.2 Описание регистров управления ядра (SCS)

Подробное описание регистров управления ядра приведено в документе Cortex-M4 Technical Reference Manual.

Таблица 152 – Описание регистров управления ядра (SCS)

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0xE000_0000		ITM		
0xE000_1000		DWT		
	0x000			
0xE000_E000		SC		
	0x004	ICTR		Регистр описания контроллера прерываний
	0x008	ACTLR		Регистр настройки ядра
	0x010	STCSR		Регистр настройки и состояния SysTick
	0x014	STRVR		Регистр основания счета SysTick
	0x018	STCVR		Регистр текущего значения SysTick
	0x01C	STCR		Регистр калибровки SysTick
	0x100-0x11C			
	0x180-0x19C			
	0x200-0x21C			
	0x300-0x31C			
	0x400-0x41C			
	0xD00	CPUID		Регистр ID процессорного ядра
	0xD04	ICSR		Регистр настройки и состояния прерываний
	0xD08	VTOR		Регистр смещения таблицы векторов прерывания
	0xD0C	AIRCR		Регистр прерываний приложений и сброса
	0xD10	SCR		Регистр настройки системы
	0xD14	CCR		Регистр настройки и конфигурации
	0xD18	SHPR1		Регистр настройки приоритета 1 обработчиков
	0xD1C	SHPR2		Регистр настройки приоритета 2 обработчиков
	0xD20	SHPR3		Регистр настройки приоритета 3 обработчиков
	0xD24	SHCSR		Регистр настройки и состояния обработчиков
	0xD28	CFSR		Регистр конфигурации состояния ошибки
	0xD2C	HFSR		Регистр состояния HardFault
	0xD30	DFSR		Регистр состояния ошибки отладки
	0xD34	MMFAR		Регистр адреса ошибки управления памяти

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0xD38	BFAR		Регистр адреса ошибки BusFault
	0xD3C	AFSR		Регистр состояния ошибки настройки ядра
	0xD40	ID_PFR0		Регистр
	0xD44	ID_PFR1		
	0xD48	ID_DFR0		
	0xD4C	ID_AFR0		
	0xD50	ID_MMFR0		
	0xD54	ID_MMFR1		
	0xD58	ID_MMFR2		
	0xD5C	ID_MMFR3		
	0xD60	ID_IASR0		
	0xD64	ID_IASR1		
	0xD68	ID_IASR2		
	0xD6C	ID_IASR3		
	0xD70	ID_IASR4		
	0xD88	CPACR		
	0xD90	MPU_TYPE		
	0xD94	MPU_CTRL		
	0xD98	MPU_RNR		
	0xD9C	MPU_RBAR		
	0xDA0	MPU_RASR		
	0xDA4	MPU_RBAR_A1		
	0xDA8	MPU_RASR_A1		
	0xDAC	MPU_RBAR_A2		
	0xDB0	MPU_RASR_A2		
	0xDB4	MPU_RBAR_A3		
	0xDB8	MPU_RASR_A3		
	0xDF0	DHCSR		
	0xDF4	DCRSR		
	0xDF8	DCRDR		
	0xDFC	DEMCR		
	0xF00	STIR		
	0xF34	FPCCR		
	0xF38	FPCAR		
	0xF3C	FPDSCR		
	0xF40	MVFR0		
	0xF44	MVFR1		
	0xFD0	Peripheral ID4		
	0xFE0	Peripheral ID0		
	0xFE4	Peripheral ID1		
	0xFE8	Peripheral ID2		
	0xFEC	Peripheral ID3		
	0xFF0	Component ID0		
	0xFF4	Component ID1		
	0xFF8	Component ID2		
	0xFFC	Component ID3		
0xE004_0000		TPUI		

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0xE00F_F000				
	0x000	SCS		
	0x004	DWT		
	0x008	FPB		
	0x00C	ITM		
	0x010	TPUI		
	0x014	ETM		
	0x018	End Marker		
	0xFFC	SYSTEM ACCESS		

7.26 Контроллер блока защиты памяти (MPU)

Подробное описание блока защиты памяти приведено в документе Cortex-M4 Technical Reference Manual.

7.27 Контроллер блока вычисления с плавающей запятой (FPU)

Подробное описание блока вычислений с плавающей запятой приведено в документе Cortex-M4 Technical Reference Manual.

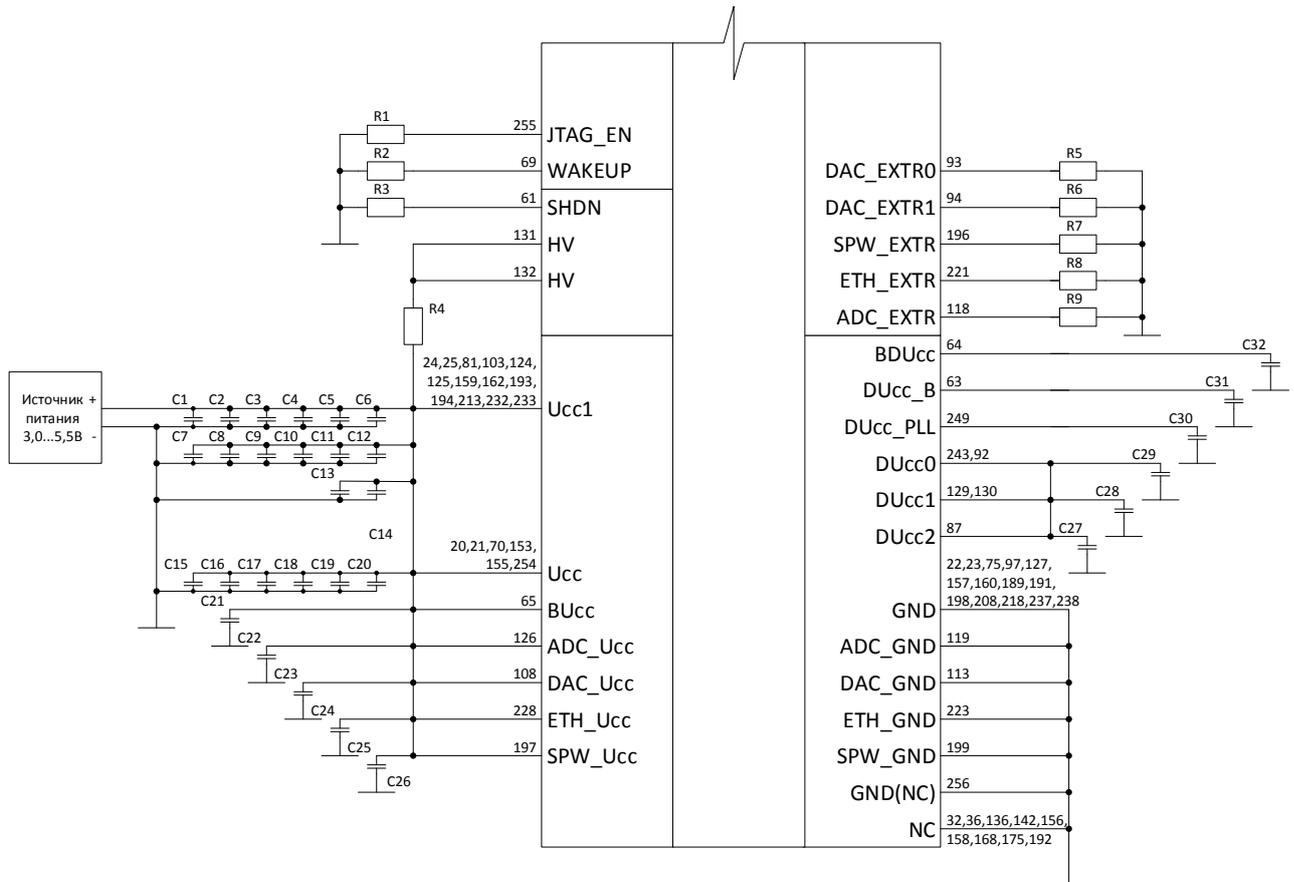
7.28 Контроллер порта трассировки (TPIU)

Подробное описание регистров управления ядра приведено в документе Cortex-M4 Technical Reference Manual.

7.29 Контроллер блока расширенной трассировки (ETM)

Подробное описание блока расширенной трассировки приведено в CoreSight ETM-M4 Technical Reference Manual.

8 Типовая схема включения



- C1 – конденсатор емкостью не менее 22 мкФ, номинальным напряжением не менее 10 В;
- C2 – C26 – конденсаторы емкостью не менее 0,1 мкФ, номинальным напряжением не менее 10 В;
- C27 – C32 – конденсаторы емкостью не менее 10 мкФ, номинальным напряжением не менее 10 В;
- R1, R3, R4 – резисторы сопротивлением 10 кОм ± 10 %;
- R2 – резистор сопротивлением 1 кОм ± 10 %;
- R5 – R9 – резисторы сопротивлением 120 кОм ± 5 %.

Конденсаторы должны располагаться максимально близко к соответствующим выводам питания.

Рисунок 166 – Типовая схема включения микросхемы

9 Основные зависимости

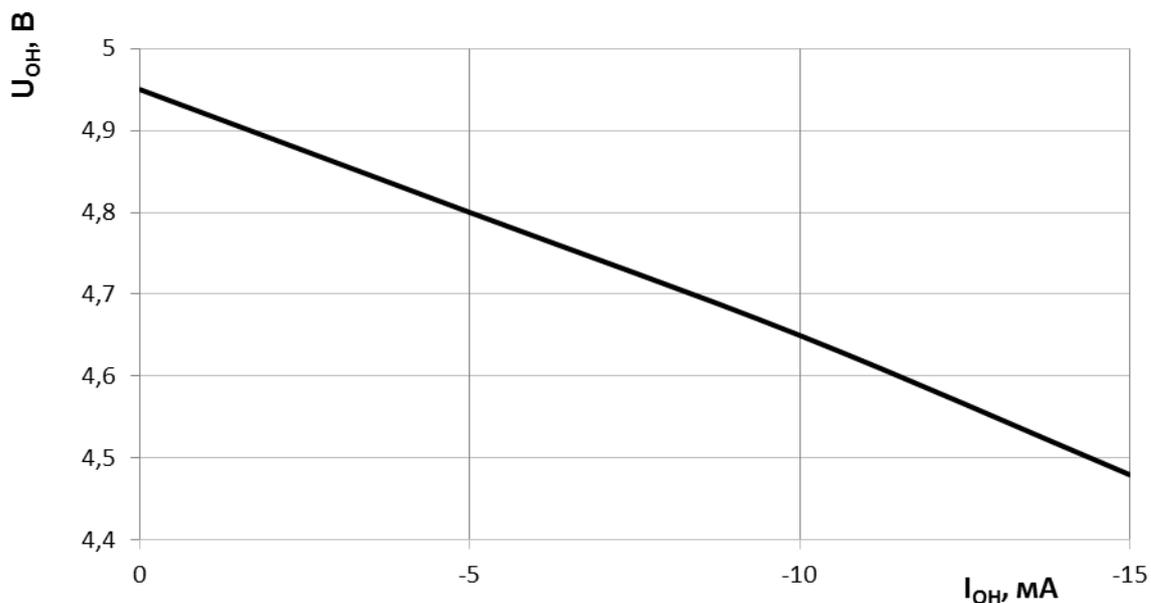


Рисунок 167 – Зависимость выходного напряжения высокого уровня U_{OH} микросхемы 1986BE8Т от тока нагрузки I_{OH} при напряжении питания 5,5 В

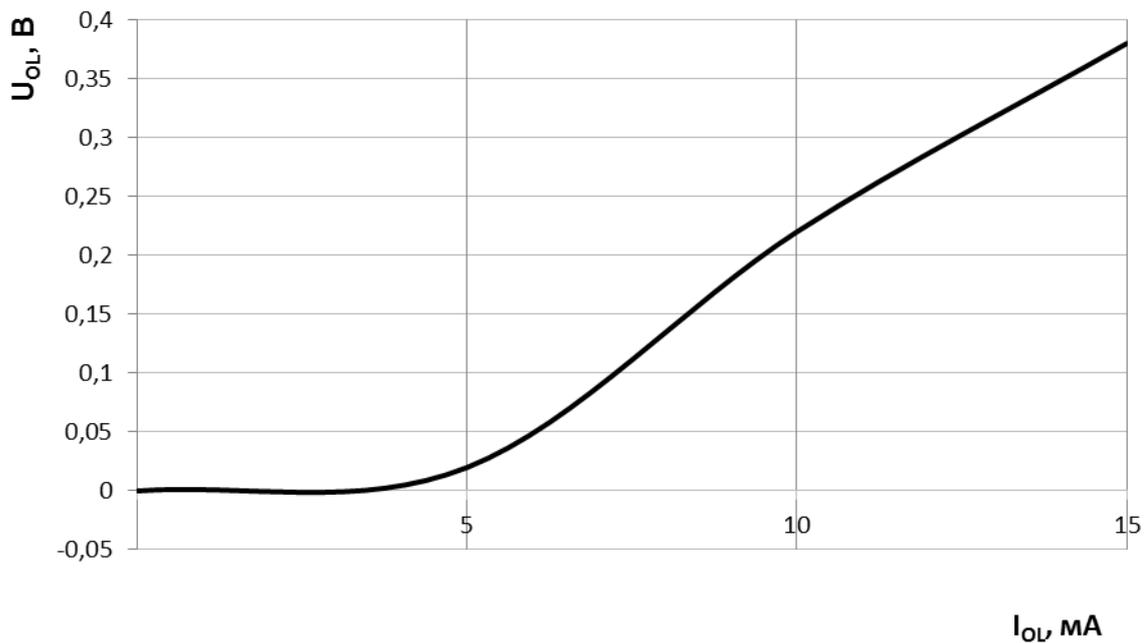


Рисунок 168 – Зависимость выходного напряжения низкого уровня U_{OL} микросхемы 1986BE8Т от тока нагрузки I_{OL} при напряжении питания 5,5 В

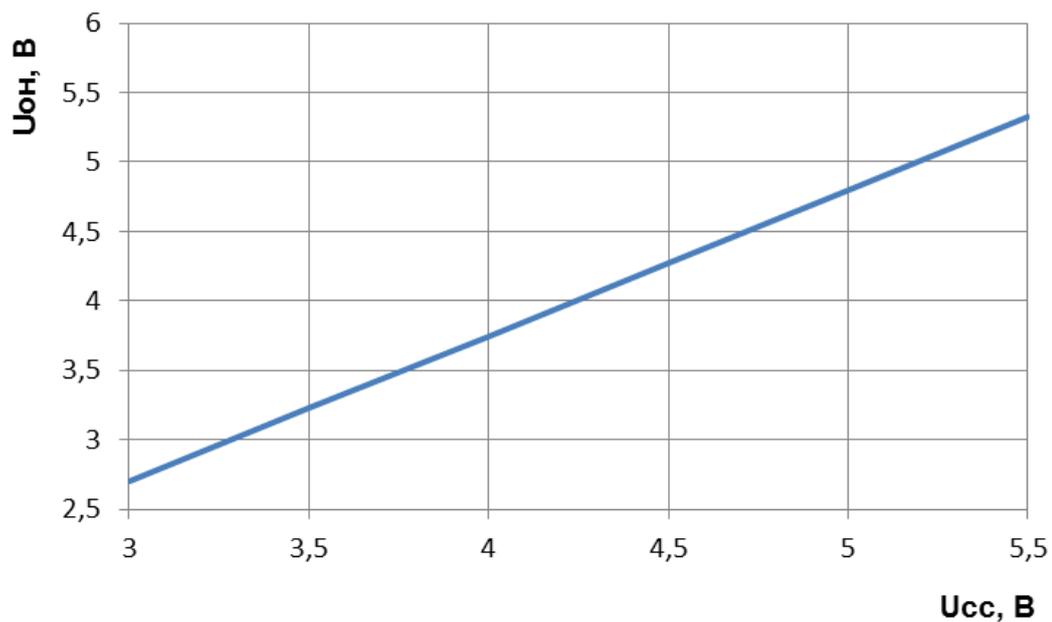


Рисунок 169 – Зависимость выходного напряжения высокого уровня U_{OH} микросхемы 1986BE8Т от напряжения питания при $I_{OH} = 6$ мА

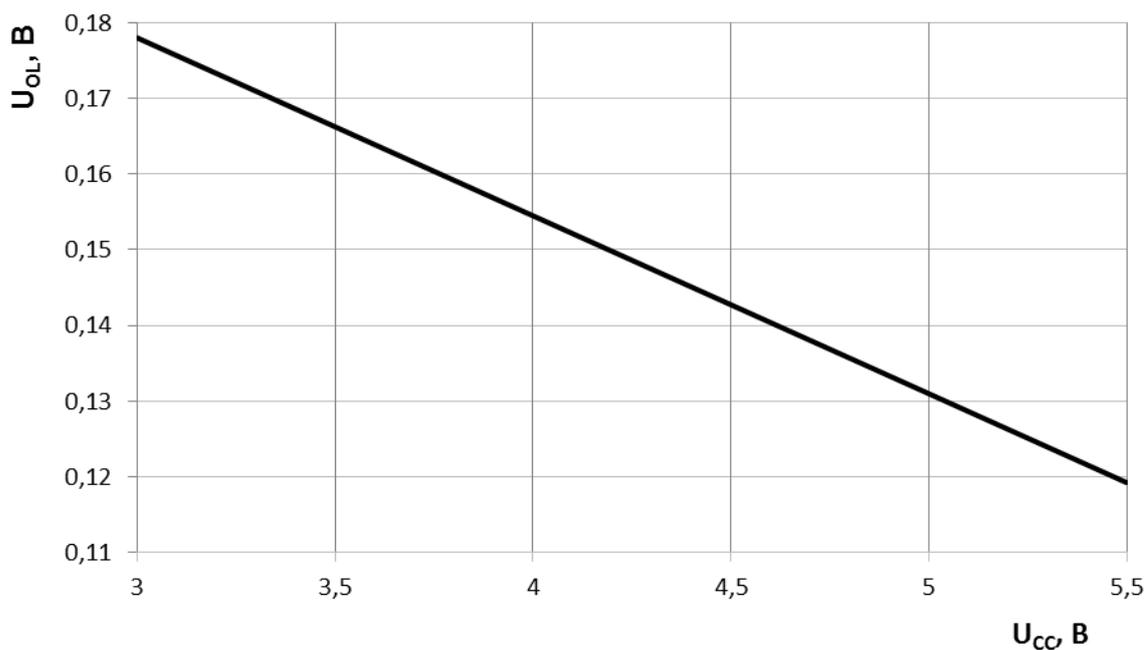


Рисунок 170 – Зависимость выходного напряжения низкого уровня U_{OL} микросхемы 1986BE8Т от напряжения питания при $I_{OL} = 6$ мА

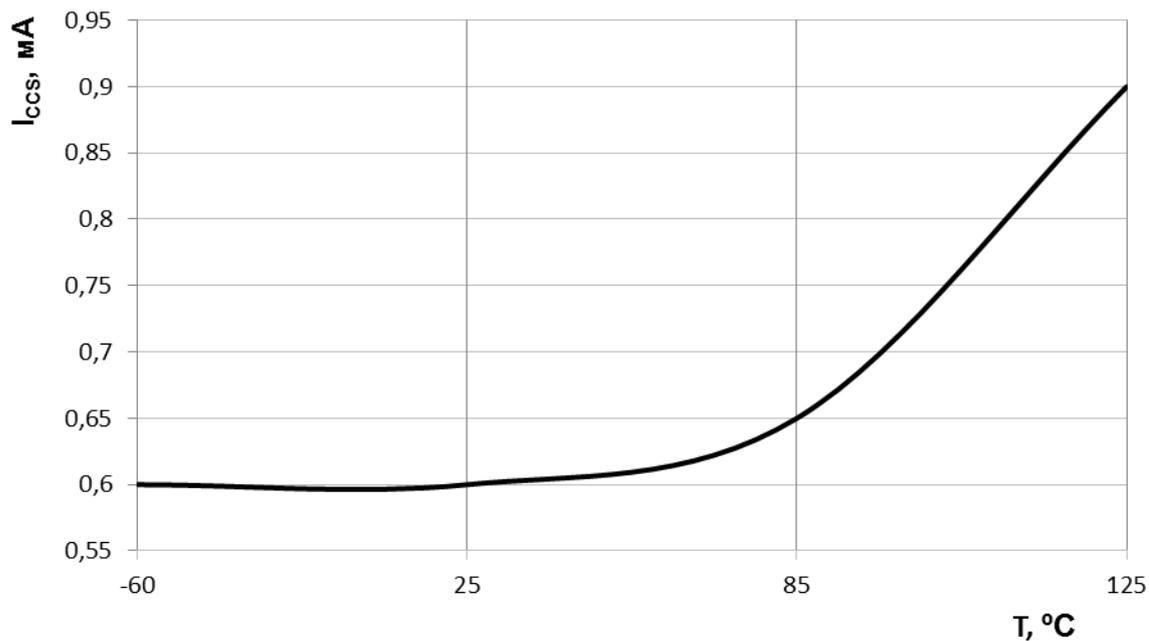


Рисунок 171 – Зависимость статического тока потребления I_{CCS} микросхемы 1986BE8Т от температуры при напряжении питания 5,5 В

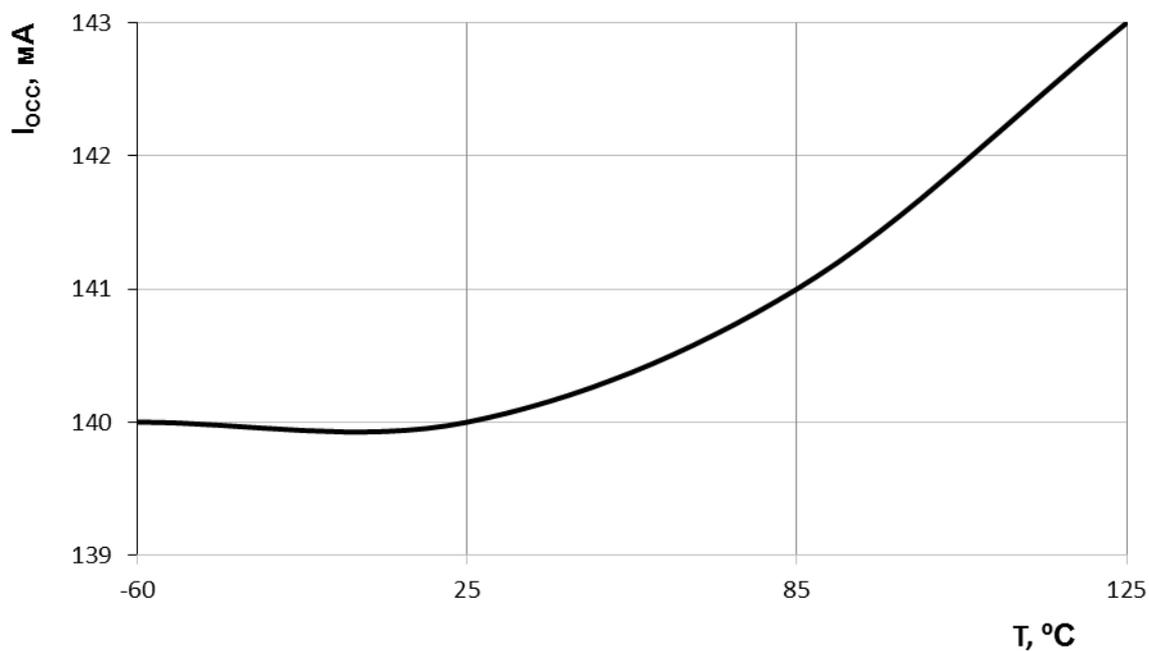


Рисунок 172 – Зависимость динамического тока потребления I_{ODC} микросхемы 1986BE8Т от температуры при напряжении питания 5,5 В

10 Предельно-допустимые характеристики микросхемы

Таблица 153 – Предельно-допустимые режимы эксплуатации и предельные режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В, - на выводах U _{CC} ; - на выводах ADC_U _{CC} ; - на выводах DAC_U _{CC} ; - на выводах SPW_U _{CC} ; - на выводах ETH_U _{CC}	U _{CC} **	3,0	5,5	- 0,3	6,0
Входное напряжение встроенного регулятора, В, на выводах U _{CC1}	U _{CC1}	2,2	U _{CC}	- 0,3	U _{CC} + 0,3
Напряжение питания батарейного домена, В, на выводе BU _{CC}	U _{CCB}	2,0	5,5	- 0,3	6,0
Напряжение питания цифровой части, В, на выводах DU _{CC0} , DU _{CC1} , DU _{CC2} , DU _{CC_PLL} , DU _{CC_V}	U _{CCD}	1,62	1,98	- 0,3	2,5
Входное напряжение высокого уровня, В, - на 5 В толерантных выводах: PA[31-0], PB[31-0], PC[5-0], PC[31], PC[30], PD[31-0], PE[31-0]	U _{IH}	2,0	5,5	-	6,0
- на 5 В не толерантных выводах: PC[29-6], JTAG_EN, SHDN		2,0	U _{CC}	-	U _{CC} + 0,3
Входное напряжение низкого уровня, В, - на выводах: PA[31-0], PB[31-0], PC[31-0], PD[31-0], PE[31-0], JTAG_EN, SHDN, RSTn0, RSTn1, WAKEUP	U _{IL}	0	0,8	- 0,3***	-
Выходной ток высокого уровня, мА	I _{OH}	- 6	-	- 15	-
Выходной ток низкого уровня, мА	I _{OL}	-	6	-	15
Частота следования импульсов АЦП, МГц	f _{C_ADC}	-	11	-	-
Длительность тактового импульса АЦП, нс	t _{W_ADC}	45	-	-	-
Частота следования импульсов CLK, МГц	f _{C_CLK}	-	100	-	-
Длительность тактового импульса CLK, нс	t _{W_CLK}	5	-	-	-
Частота следования импульсов CPU CLK, МГц	f _{C_CPUCLK}	-	64	-	-
Частота следования импульсов тактовых сигналов PLL, МГц	f _{C_PLL}	2	16	-	-
Емкость нагрузки на выводах, пФ	C _L	-	50	-	100
Параметры генератора тактовых сигналов					
Входное напряжение высокого уровня, В, на входах генератора тактового сигнала в режиме BYPASS: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U _{IH1}	0,8·U _{CC}	U _{CC}	-	U _{CC} + 0,3
Входное напряжение низкого уровня, В, на входах генератора тактового сигнала в режиме BYPASS: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U _{IL1}	0	0,2·U _{CC}	- 0,3	-

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Амплитуда входного сигнала, В, на входах генератора тактового сигнала в режиме OSC: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U _{A_OSC}	0,05	U _{CC}	–	–
Частота следования импульсов тактовых сигналов HSE, МГц, - в режиме BYPASS - в режиме OSC	f _{C_HSE}	–	100	–	–
		1	30	–	–
Частота следования импульсов тактовых сигналов LSE, - в режиме BYPASS, МГц - в режиме OSC, кГц	f _{C_LSE}	–	1	–	–
		20	40	–	–
Параметры АЦП					
Напряжение верхней границы внешнего опорного напряжения АЦП, В	U _{ADC_REFp}	2,4	U _{CC}	–	–
Напряжение нижней границы внешнего опорного напряжения АЦП, В	U _{ADC_REFn}	0	U _{CC} – 2,4	–	–
Диапазон напряжения внешнего опорного источника АЦП, В, U _{REF(ADC)} = U _{ADC_REFp} – U _{ADC_REFn}	U _{REF(ADC)}	2,4	U _{CC}	–	–
Параметры ЦАП					
Напряжение верхней границы внешнего опорного напряжения ЦАП, В	U _{DAC_REFp}	2,4	U _{CC}	–	–
Напряжение нижней границы внешнего опорного напряжения ЦАП, В	U _{DAC_REFn}	0	U _{CC} – 2,4	–	–
Диапазон напряжения внешнего опорного источника ЦАП, В, U _{REF(DAC)} = U _{DAC_REFp} – U _{DAC_REFn}	U _{REF(DAC)}	2,4	U _{CC}	–	–
Сопrotивление нагрузки ЦАП, кОм	R _{LOADDAC}	10	–	–	–
Емкость нагрузки ЦАП, пФ	C _{LOADDAC}	–	100	–	–
Параметры SpaceWire					
Входное напряжение на выводах приемника SpaceWire, В	U _{IN_RXSPW}	0	2,4	– 0,3	U _{CC} + 0,3
Входное напряжение дифференциальное, В	U _{ID_SPW}	0,1	2,4	–	–
Входное напряжение синфазное, В	U _{IC_SPW}	0,05	2,35	–	–
Параметры EtherNet					
Входное напряжение на выводах приемника EtherNet, В	U _{IN_RXETH}	0	U _{CC}	– 0,3	U _{CC} + 0,3
Входное напряжение дифференциальное, мВ	U _{ID_ETH}	500	U _{CC}	–	–
<p>** Напряжения питания на выводах ADC_U_{CC}, DAC_U_{CC}, SPW_U_{CC}, ETH_U_{CC} должны отличаться от напряжения питания на выводах U_{CC} не более, чем на ±0,2 В.</p> <p>*** Режим без ограничения входного тока. Допускается вместо ограничения входного напряжения ограничивать входной вытекающий ток не более 1 мА.</p> <p>Примечание – Не допускается одновременное задание более одного предельного режима.</p>					

11 Электрические параметры микросхемы

Таблица 154 – Электрические параметры микросхем при приёмке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение высокого уровня, В	U_{OH}	2,4	–	25, 125, – 60
Выходное напряжение низкого уровня, В	U_{OL}	–	0,4	
Напряжение питания формирования сигнала «Сброс» встроенного регулятора, В, - при повышении U_{CC} ; - при понижении U_{CC}	U_{POR}	–	3,0	
		2,5	–	
Выходное напряжение встроенного регулятора, В, в режиме «Включено»	U_{O_D}	1,62	1,98	
Входной ток высокого уровня, мкА, на выводах SHDN, JTAG_EN	I_{IH1}	10	500	
Входной ток низкого уровня, мкА, на выводах WAKEUP, RSTn0, RSTn	I_{IL1}	– 500	– 10	
Выходной ток низкого уровня в состоянии «Выключено», мкА, на выводах FT_RSTn, FT_WRNn, при $U_O = 0$ В	I_{OZL}	– 500	– 10	
Входной ток высокого уровня, мкА, на выводах генератора тактового сигнала	I_{IH2}	– 100	100	
Входной ток низкого уровня, мкА, на выводах генератора тактового сигнала	I_{IL2}	– 100	100	
Входной ток утечки высокого уровня, мкА, на входах/выходах портов А – Е в цифровом режиме; на выводах WAKEUP, RSTn0, RSTn1, при $U_I = U_{CC}$	I_{ILH}	– 10	10	
Выходной ток высокого уровня в состоянии «Выключено», мкА, на выводах FT_RSTn, FT_WRNn, при $U_O = U_{CC}$	I_{OZH}	– 10	10	
Входной ток утечки низкого уровня, мкА, на входах/выходах портов А – Е в цифровом режиме; на выводах SHDN, JTAG_EN, при $U_I = 0$ В	I_{ILL}	– 10	10	
Входной ток, мкА на входах/выходах портов А – Е в аналоговом режиме, при $U_I = 2,5$ В	I_I	– 500	500	
Статический ток потребления, мА	I_{CC}	–	5*	
Статический ток потребления в режиме пониженного энергопотребления, мА	I_{CCS}	–	1	
Динамический ток потребления процессорного ядра, мА, при $U_{CC} = U_{CC1} = U_{CCB} = 3,6$ В	I_{OCC1}	–	150	
Динамический ток потребления, мА, при $U_{CC} = U_{CC1} = U_{CCB} = 5,5$ В	I_{OCC2}	–	750*	
Выходная частота генератора HSE, МГц, в режиме резонатора -максимальная -минимальная	f_{O_HSE}	30	–	
		–	1	
Максимальная выходная частота генератора HSE, МГц, в режиме генератора	f_{O_HSEmax}	100	–	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходная частота генератора LSE, кГц, в режиме резонатора -максимальная	f _{O_LSE}	40	–	25, 125, – 60
-минимальная		–	20	
Максимальная выходная частота генератора LSE, МГц, в режиме генератора	f _{O_LSEmax}	1	–	
Выходная частота генератора HSI, МГц	f _{O_HSI}	6	10	
Выходная частота генератора LSI, кГц	f _{O_LSI}	20	60	
Выходная частота PLL, МГц - максимальная	f _{O_PLL}	100	–	
- минимальная		–	40	
Коэффициент деления входной частоты, 1	K _{QPLL}	1	16	
Коэффициент деления синтезируемой частоты, 1	K _{NPLL}	4	75	
Параметры АЦП				
Разрядность АЦП, бит	E _{NADC}	12	–	25, 125, – 60
Дифференциальная нелинейность АЦП, ЕМР	E _{DLADC}	– 1	2	
Интегральная нелинейность АЦП, ЕМР	E _{ILADC}	– 3	3	
Ошибка смещения, ЕМР	E _{OFFADC}	– 25	25	
Ошибка усиления, %	E _{GAINADC}	– 10	10	
Параметры ЦАП				
Разрядность ЦАП, бит	E _{NDAC}	12	–	25, 125, – 60
Дифференциальная нелинейность ЦАП, ЕМР	E _{DLDAC}	– 1	2	
Интегральная нелинейность ЦАП, ЕМР	E _{ILDAC}	– 10	10	
Ошибка смещения, мВ	E _{OFFDAC}	– 50	50	
Ошибка усиления, %	E _{GAINDAC}	– 10	10	
Параметры SpaceWire				
Выходное дифференциальное напряжение, мВ	U _{OD_TXSPW}	250	450	25, 125, – 60
Выходное синфазное напряжение, В	U _{OS_TXSPW}	1,125	1,375	
Входной ток на выводах приемника SpaceWire, мкА	I _{IN_RXSPW}	– 20	20	
Параметры EtherNet				
Выходное дифференциальное напряжение, В - при U _{CC} = 3,6 В; - при U _{CC} = (3,0 – 5,5) В	U _{OD_TXETH}	2,5	3,5	25, 125, – 60
		2,0	4,5	
Входной ток на выводах приемника EtherNet, мкА - приемник выключен	I _{IN_RXETH}	– 1	1	
- приемник включен		– 2000	2000	

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

Коэффициент программируемости микросхем 1986BE8Т, N_{PR}, не менее 0,7*. После программирования ЭТТ не требуется.

* Значение уточняется в ходе проведения испытаний.

12 Справочные данные

Таблица 155 – Предельно-допустимые значения ОИН

Тип вывода	Длительность ОИН, мкс					
	Предельно-допустимое напряжение ОИН, В (мДж)					
	0,1		1,0		10,0	
1986BE8Т						
Входы	- 286,5 (30)	428,6 (63)	- 67,9 (25)	107,9 (44)	- 11,1 (14)	15,8 (30)
Выходы	- 287 (30)	428,2 (63)	- 68,1 (24)	108,5 (40)	- 12,9 (14)	15,7 (31)
Цепь питания	- 284,7 (33)	427 (63)	- 71,3 (28)	106,9 (50)	- 11,7 (50)	16,5 (52)
1986BE81Т						
Входы						
Выходы						
Цепь питания						

Таблица 156 – Справочные параметры микросхемы

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Питание				
Напряжение питания U_{CC} разрешения работы блока сброса POVR, В	U_{POVR_max}	5,5	–	25, 125, – 60
Напряжение питания U_{CC} запрещения работы блока сброса POVR, В	U_{POVR_min}	–	5,7	
Ток потребления по выводу BU $_{CC}$, мА	I_{CC1}	–	20	
Ток потребления по выводам E T HU $_{CC}$, мА	I_{CC2}	–	20	
Ток потребления по выводам E T HU $_{CC}$, мА, в режиме замыкания	I_{CC3}	–	150	
Ток потребления по выводам SPWU $_{CC}$, мА	I_{CC4}	–	40	
Ток потребления по выводам SPWU $_{CC}$, мА, в режиме замыкания	I_{CC5}	–	40	
Ток потребления по выводам DACU $_{CC}$, мА	I_{CC6}	–	10	
Ток потребления по выводам DACREF, мА	I_{CC7}	–	10	
Ток потребления по выводам ADCU $_{CC}$, мА	I_{CC8}	–	10	
Ток потребления по выводам ADCREF, мА	I_{CC9}	–	10	
Ток потребления по выводам HV в режиме программирования, мА	I_{HV}	-	10	
Время задержки сигнала сброса от момента появления питания, мс	$T_{PORRSTn}$	2	8	
Порты ввода-вывода				
Входной гистерезис переключения приемника, В	U_{HES}	0,15	0,3	25, 125, – 60
Выходной ток высокого уровня срабатывания схемы защиты от перегрузки, мА	I_{HLOAD}	– 100	– 10	
Выходной ток низкого уровня срабатывания схемы защиты от перегрузки, мА	I_{LOAD}	10	100	
Сопrotивление внутреннего резистора подтяжки, кОм	R_{PULL}	40	60	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Тактовые генераторы				
Время запуска генератора HSE, нс	t _{HSEON}	–	100+512 тактов	25, 125, – 60
Длительность фильтруемых иголок в HSE, нс	t _{HSEFILTER}	–	3	
Время запуска генератора LSE, нс, в режиме резонатора	t _{LSEON}	–	100+512 тактов	
Длительность фильтруемых иголок в LSE, нс	t _{LSEFILTER}	–	3	
Время запуска PLL, мкс	t _{PLLON}	–	60	
Емкость нагрузки, пФ, на выводах генератора тактового сигнала в режиме OSC: HSE0_OSC_IN и HSE0_OSC_OUT; HSE1_OSC_IN и HSE1_OSC_OUT	C _{L_HSE}	15	30	
Емкость нагрузки, пФ, на выводах генератора тактового сигнала в режиме OSC: LSE_OSC_IN и LSE_OSC_OUT	C _{L_LSE}	40	60	
АЦП				
Число тактов преобразования	N	43	43	25, 125, – 60
Время включения АЦП, мкс	t _{ADCON}	–	10	
Опорное напряжение 2,5 В	U _{REF_ADC}	2,3	2,7	
Время зарядки от ref_2,5 В, мкс	t _{REF}	–	2	
Частота преобразования, Мвыб./с	f _{CONV}	–	0,25	
Входное напряжение АЦП, В, при опоре от вывода ADC_Ucc	U _{IN_ADC}	0	U _{CC_ADC}	
Входное напряжение АЦП, В, при опоре от выводов ADCREF	U _{IN_ADC1}	U _{ADC_REFn}	U _{ADC_REFp}	
Входное напряжение АЦП, В, при опоре от источника опорного напр. 2,4 В	U _{IN_ADC24}	0	2,2	
Сопротивление внешнего токозадающего резистора на выводе ADC_EXTR, кОм	R _{L_ADC_EXTR}	114	126	
SpaceWire				
Ток короткого замыкания, мА	I _{OS_TX}	5,9	20,0	25, 125, – 60
Дифференциальный ток короткого замыкания, мА	I _{OSD_TX}	4,1	8,5	
Входное сопротивление дифференциальное на выводах приемника SpaceWire, кОм	R _{RX_IN_SPW}	500	1000	
Выходное сопротивление дифференциальное на выводах приемника SpaceWire, Ом	R _{TX_OUT_SPW}	98	224	
Время нарастания/спада выходного сигнала, нс, от 20% до 80%	t _{RISE_TX} t _{FALL_TX}	–	0,8	
Сквозность выходного сигнала, %	CK _{TX}	45,0	55,0	
Дифференциальное входное напряжение перехода в “1”, мВ	U _{TH_RX}	-	100	
Дифференциальное входное напряжение перехода в “0”, мВ	U _{TL_RX}	– 100	–	
EtherNet				
Изменение выходного дифференциального напряжения, В, на выводах TXP и TXM	ΔU _{OD_TX}	–	0,8	25, 125, – 60
Выходное синфазное напряжение передатчика, В, выводы TXP и TXM	U _{OS_TX}	1,2	3,5	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Изменение выходного синфазного напряжение передатчика, В	ΔU_{OS_TX}	–	2,2	
Дифференциальное пороговое напряжение приемника, В	U_{OD}	– 0,15	0,15	
Гистерезис приемника, мВ	U_h	5	15	
Время включения передатчика по сигналу TXEN, нс	$t_{TXEN-TX}$	–	12	
Время задержки передатчика от сигнала TXD, нс	t_{TXD-TX}	–	10	
Длительность фронта ТХР и ТХН, нс	t_{FTXP} t_{FTXN}	–	8	
Источник опорного напряжения				
Сопротивление внешнего токозадающего резистора на выводе EXTR, кОм	R_{L_EXTR}	118,8	121,2	25, 125, – 60



Размеры КП 90 × 90 мкм

- 1 Номера контактным площадкам (КП), кроме первой, присвоены условно. Расположение КП соответствует топологическому чертежу.
- 2 М – маркировка кристалла:
MLDR85 для микросхем К1986BE8Н4;
MLDR138 для микросхем К1986BE81Н4.
- 3 Координаты КП см. в таблице ниже.
- 4 Толщина кристалла (0,475 ± 0,025) мм.
- 5 Материал КП – сплав AlCu толщиной 0,88 мкм с содержанием Cu 0,5 %.

Рисунок 174 – Кристалл (бескорпусное исполнение)

Таблица 157 – Координаты КП кристалла

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		Х	У			Х	У
1	RESETn0	-6303	4496	48	PD<27>	-6303	-2411
2	RESETn1	-6303	4352	49	PD<24>	-6303	-2555
3	FT_RESET	-6303	4208	50	PD<25>	-6303	-2699
4	PE<31>	-6303	4064	51	PD<22>	-6303	-2843
5	FT_WARNG	-6303	3920	52	PD<23>	-6303	-2987
6	PE<29>	-6303	3776	53	PD<20>	-6303	-3131
7	PE<30>	-6303	3632	54	PD<21>	-6303	-3275
8	PE<27>	-6303	3488	55	PD<18>	-6303	-3419
9	PE<28>	-6303	3344	56	PD<19>	-6303	-3563
10	PE<25>	-6303	3200	57	PD<16>	-6303	-3707
11	PE<26>	-6303	3056	58	PD<17>	-6303	-3851
12	PE<23>	-6303	2912	59	PD<15>	-6303	-3995
13	PE<24>	-6303	2768	60	SHDN_LDO	-6303	-4139
14	PE<21>	-6303	2624	61	BDUcc	-6303	-4278
15	PE<22>	-6303	2480	62	DUcc_B	-6303	-4491
16	PE<19>	-6303	2336	63	BUcc	-5741	-5803
17	PE<20>	-6303	2192	64	OSC_IN_2	-5491	-5803
18	PE<17>	-6303	2048	65	OSC_OUT_2	-5346	-5803
19	PE<18>	-6303	1904	66	STANDBY	-5201	-5803
20	Ucc	-6303	1759	67	WAKEUP	-5056	-5803
21	Ucc	-6303	1551	68	Ucc	-4906	-5803
22	GND	-6303	1407	69	Ucc	-4706	-5803
23	GND	-6303	1263	70	PD<14>	-4561	-5803
24	Ucc1	-6303	1119	71	PD<13>	-4416	-5803
25	Ucc1	-6303	901	72	PD<12>	-4271	-5803
26	PE<16>	-6303	757	73	PD<11>	-4126	-5803
27	PE<15>	-6303	613	74	GND	-3980	-5803
28	PE<14>	-6303	469	75	GND	-3836	-5803
29	PE<13>	-6303	325	76	GND	-3691	-5803
30	PE<12>	-6303	181	77	PD<10>	-3546	-5803
31	PE<11>	-6303	37	78	PD<9>	-3401	-5803
32	PE<10>	-6303	-107	79	PD<8>	-3256	-5803
33	PE<9>	-6303	-251	80	PD<7>	-3111	-5803
34	PE<8>	-6303	-395	81	PD<6>	-2966	-5803
35	PE<7>	-6303	-539	82	Ucc1	-2821	-5803
36	PE<6>	-6303	-683	83	Ucc1	-2676	-5803
37	PE<5>	-6303	-827	84	Ucc1	-2531	-5803
38	PE<4>	-6303	-971	85	PD<5>	-2386	-5803
39	PE<3>	-6303	-1115	86	PD<4>	-2241	-5803
40	PE<2>	-6303	-1259	87	PD<3>	-2096	-5803
41	PE<1>	-6303	-1403	88	PD<2>	-1951	-5803
42	PE<0>	-6303	-1547	89	PD<1>	-1806	-5803
43	PD<31>	-6303	-1691	90	DUcc0	-1661	-5803
44	PD<30>	-6303	-1835	91	DUcc0	-1451	-5803
45	PD<28>	-6303	-1979	92	PD<0>	-1306	-5803
46	PD<29>	-6303	-2123	93	PC<31>	-1161	-5803
47	PD<26>	-6303	-2267	94	PC<30>	-1016	-5803

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		Х	У			Х	У
95	DAC0_IREF	-871	-5803	142	HV	6303	-4123
96	DUcc0	-726	-5803	143	DUcc0	6303	-3973
97	DUcc0	-516	-5803	144	PC<5>	6303	-3823
98	DAC0_EXTR	-371	-5803	145	HV	6303	-3673
99	DAC1_EXTR	-226	-5803	146	PC<3>	6303	-3518
100	PC<29>	-81	-5803	147	PC<4>	6303	-3364
101	PC<28>	64	-5803	148	PC<2>	6303	-3210
102	GND	215	-5803	149	PC<0>	6303	-3056
103	GND	361	-5803	150	PC<1>	6303	-2902
104	PC<27>	522	-5803	151	PB<30>	6303	-2748
105	PC<26>	669	-5803	152	PB<31>	6303	-2594
106	PC<25>	816	-5803	153	PB<29>	6303	-2440
107	PC<24>	963	-5803	154	PB<27>	6303	-2286
108	PC<23>	1110	-5803	155	PB<28>	6303	-2132
109	Ucc1	1254	-5803	156	PB<25>	6303	-1978
110	Ucc1	1399	-5803	157	PB<26>	6303	-1824
111	PC<22>	1543	-5803	158	PB<23>	6303	-1670
112	PC<21>	1690	-5803	159	PB<24>	6303	-1516
113	PC<20>	1837	-5803	160	PB<22>	6303	-1362
114	PC<19>	1984	-5803	161	PB<21>	6303	-1208
115	DAC_Ucc	2128	-5803	162	PB<20>	6303	-1054
116	DAC_Ucc	2273	-5803	163	Ucc	6303	-909
117	PC<18>	2417	-5803	164	PB<19>	6303	-759
118	PC<17>	2564	-5803	165	Ucc	6303	-504
119	PC<16>	2711	-5803	166	GND	6303	-359
120	PC<15>	2858	-5803	167	Ucc1	6303	-214
121	DAC_GND	3002,93	-5803	168	GND	6303	-64
122	DAC_GND	3147,93	-5803	169	PB<18>	6303	91
123	PC<14>	3291	-5803	170	Ucc1	6303	246
124	PC<13>	3438	-5803	171	PB<17>	6303	396
125	PC<12>	3585	-5803	172	PB<16>	6303	550
126	PC<11>	3732	-5803	173	PB<15>	6303	704
127	ADC_EXTR	3879	-5803	174	PB<14>	6303	858
128	ADC_GND	4023,93	-5803	175	PB<13>	6303	1012
129	ADC_GND	4168,93	-5803	176	PB<12>	6303	1166
130	PC<10>	4312	-5803	177	PB<11>	6303	1320
131	PC<9>	4459	-5803	178	PB<10>	6303	1474
132	PC<8>	4606	-5803	179	PB<9>	6303	1628
133	PC<7>	4753	-5803	180	PB<8>	6303	1782
134	Ucc1	4897	-5803	181	PB<7>	6303	1936
135	Ucc1	5042	-5803	182	PB<6>	6303	2090
136	Ucc1	5187	-5803	183	PB<4>	6303	2244
137	ADC_Ucc	5332	-5803	184	PB<5>	6303	2398
138	GND	5477,93	-5803	185	PB<2>	6303	2552
139	ADC_Ucc	5622	-5803	186	PB<3>	6303	2706
140	PC<6>	5769	-5803	187	PB<0>	6303	2860
141	DUcc0	6303	-4391	188	PB<1>	6303	3014

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		Х	У			Х	У
189	PA<30>	6303	3168	230	ETH0_R	683	5803
190	PA<31>	6303	3322	231	ETH0_IREF	538	5803
191	PA<28>	6303	3476	232	ETH_GND	387,07	5803
192	PA<29>	6303	3630	233	ETH_GND	237,07	5803
193	PA<26>	6303	3784	234	ETH0_TXP	88	5803
194	PA<27>	6303	3938	235	ETH0_TXN	-52	5803
195	PA<25>	6303	4092	236	ETH0_RXP	-192	5803
196	GND	6303	4252	237	ETH0_RXN	-332	5803
197	GND	6303	4407	238	ETH_Ucc	-482	5803
198	Ucc1	5785	5803	239	ETH_Ucc	-632	5803
199	Ucc1	5635	5803	240	PA<16>	-782	5803
200	Ucc1	5405	5803	241	PA<15>	-927	5803
201	SPW0_IREF	5261	5803	242	PA<14>	-1072	5803
202	SPW0_EXTR	5116	5803	243	Ucc1	-1222	5803
203	SPW_Ucc	4972	5803	244	Ucc1	-1372	5803
204	GND	4827	5803	245	Ucc1	-1602	5803
205	GND	4683	5803	246	PA<13>	-1752	5803
206	SPW_GND	4537,07	5803	247	PA<12>	-1907	5803
207	SPW0_RXDP	4388	5803	248	PA<11>	-2062	5803
208	SPW0_RXDN	4248	5803	249	GND	-2212,93	5803
209	SPW0_RXSP	4108	5803	250	GND	-2357	5803
210	SPW0_RXSN	3968	5803	251	GND	-2507	5803
211	SPW0_TXSP	3548	5803	252	PA<10>	-2666	5803
212	SPW0_TXSN	3408	5803	253	PA<9>	-2821	5803
213	SPW0_TXDP	3268	5803	254	PA<8>	-2976	5803
214	SPW0_TXDN	3128	5803	255	PA<7>	-3136	5803
215	GND	2983,07	5803	256	DUcc0	-3286	5803
216	PA<24>	2840	5803	257	DUcc0	-3516	5803
217	PA<23>	2695	5803	258	PA<6>	-3666	5803
218	PA<22>	2550	5803	259	PA<5>	-3821	5803
219	PA<21>	2405	5803	260	PA<4>	-3976	5803
220	Ucc1	2255	5803	261	PA<3>	-4131	5803
221	Ucc1	2025	5803	262	PA<2>	-4286	5803
222	PA<20>	1881	5803	263	DUcc_PLL	-4551	5803
223	PA<19>	1736	5803	264	PA<1>	-4696	5803
224	PA<18>	1591	5803	265	PA<0>	-4851	5803
225	PA<17>	1446	5803	266	OSC_IN_0	-5006	5803
226	GND	1287	5803	267	OSC_OUT_0	-5146	5803
227	GND	1127	5803	268	Ucc	-5291	5803
228	OSC_IN_1	968	5803	269	JTAG_EN	-5436	5803
229	OSC_OUT_1	828	5803	270	Ucc	-5696	5803

14 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
1986BE8Т	1986BE8Т	4244.256-3	минус 60 – 125 °С
К1986BE8Т	К1986BE8Т	4244.256-3	минус 60 – 125 °С
К1986BE8ТК	К1986BE8Т•	4244.256-3	0 – 70 °С
1986BE81Т	1986BE81Т	4244.256-3	минус 60 – 125 °С
К1986BE81Т	К1986BE81Т	4244.256-3	минус 60 – 125 °С
К1986BE81ТК	К1986BE81Т•	4244.256-3	0 – 70 °С

Микросхемы с приемкой «ВП» маркируются ромбом.

Микросхемы с приемкой «ОТК» маркируются буквой «К».

Примечание – Микросхемы в бескорпусном исполнении поставляются в виде отдельных кристаллов, получаемых разделением пластины. Микросхемы поставляются в таре (кейсах) без потери ориентации. Маркировка микросхемы в бескорпусном исполнении – К1986BE8Н4, К1986BE81Н4 – наносится на тару.

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	27.01.2015	1.0.0	Введена впервые	
2	15.05.2015	1.1.0	Добавлены разделы описания функционирования. Обновлены электрические и предельно-допустимые параметры.	16–758
3	08.09.2015	2.0.0	Внесение изменений и дополнений конструктора. Корректировка в соответствии с ТУ и КД лит. А	По тексту
4	14.09.2015	2.1.0	Исправления в описании регистров EPHYn_CLK, SPHYn_CLK, UARTn_CLK, SSPn_CLK, USB_CLK Исправление в таблице справочных параметров	131, 133, 134, 135, 136 591
5	26.10.2015	2.2.0	Исправление в основных характеристиках Исправлено значение U _{CC1} в таблице 151	2 584
6	28.10.2015	2.3.0	Введение бескорпусного исполнения микросхемы К1986BE8Н4	По тексту
7	30.10.2015	2.4.0	В таблицы описания выводов (таблицы 1, 2) добавлена информация о КП кристалла	8 – 20
8	12.12.2015	2.5.0	Исправлена размерность параметров I _{OH} и I _{OL} в таблице 151 Внесены исправления в описания битов регистров Исправлено УГО рисунке 1	587 402, 403, 405, 408, 410, 420, 435 21
9	16.03.2017	2.6.0	Введение типономиналов 1986BE81Т, К1986BE81Т, К1986BE81ТК, К1986BE81Н4. Внесение исправлений и дополнений разработчика. Исправлены таблицы параметров в соответствии с ТУ и КД. Добавлены разделы Указания по применению и эксплуатации, Типовая схема включения, Основные зависимости.	По тексту По тексту 594 – 600 30, 590 – 593
10	12.05.2017	2.7.0	Добавлены диаграммы чтения и записи на внешней шине Добавлено значение коэффициента программируемости	187, 188 598
11	28.06.2017	2.8.0	Добавлено описание бита MDR_MIL1_EN регистра Регистр PER1_CLK	88
12	15.08.2017	2.9.0	Исправление ошибок и опечаток	2, 34, 43, 47, 78, 79, 84, 85, 87, 88, 90, 94, 106, 111, 113, 114, 117, 181, 182, 557
13	07.05.2018	2.10.0	Внесены исправления и добавления в описание регистров PHY_CNTR, CONFIG2, CONFIG1. Добавлен подраздел Скорость интерфейса. Внесены корректировки разработчика Внесены дополнения на рис.154 (добавлена возможность работы с микросхемами 1586ИН2У и 1586ИН4У)	377, 568, 582 356 По тексту 506
14	16.11.2018	2.11.0	Плановая корректировка по замечаниям	По тексту
15	12.12.2018	2.12.0	Плановая корректировка по замечаниям	36, 53, 83, 121, 127, 140, 152-154, 155, 156, 161-163, 202- 207, 234, 326, 344, 350, 407, 428
16	16.01.2019	2.13.0	Плановая корректировка по замечаниям	По тексту
17	01.03.2019	2.14.0	Плановая корректировка по замечаниям	79, 80, 126, 343, 392, 542
18	10.10.2019	2.15.0	Плановая корректировка по замечаниям	По тексту

Спецификация 1986BE8(81)Т, К1986BE8(81)Т, К1986BE8(81)ТК, К1986BE8(81)Н4

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
19	27.12.2019	2.16.0	Дополнен раздел Указания по применению и эксплуатации. Исправлен номинал резистора R2 на типовой схеме включения. Корректировка функций выводов питания (питание 1,8 В при SHDN = 1. Выход LDO при SHDN = 0)	29 560 8 – 19