

МИКРОСХЕМА ИНТЕГРАЛЬНАЯ

1921BK028

**Руководство пользователя**

**от 04.03.2019**

## Содержание

Введение .....	6
1 Область применения и особенности микроконтроллера .....	7
2 Краткое техническое описание микроконтроллера .....	8
2.1 Функциональные параметры .....	8
2.2 Электрические параметры .....	38
3 Архитектура изделия .....	41
3.1 Блок коммутации микроконтроллера .....	41
4 Блок управления сбросом и синхронизацией RCU .....	42
4.1 Общая система тактирования .....	42
4.2 Синтезатор частоты на основе ФАПЧ (PLL) .....	43
4.3 Система слежения за тактовыми сигналами .....	44
4.4 Сигналы сброса .....	45
4.5 Прерывания .....	46
4.6 Тактирование и сброс периферийных блоков .....	46
5 Блок управления энергопотреблением PMU .....	48
5.1 Режим Sleep .....	48
5.2 Режим Deepsleep .....	48
6 Организация памяти .....	50
7 Контроллер Flash-памяти .....	53
7.1 Основная Flash-память .....	53
7.2 Загрузочная Flash-память .....	54
7.3 Сервисный сброс всей Flash-памяти .....	56
8 Контроллер внешней памяти .....	57
9 Контроллер прямого доступа к памяти DMA .....	60
9.1 Программное управление контроллером DMA .....	61
9.2 Правила обмена данными .....	66
9.3 Правила арбитража .....	67
9.4 Типы циклов .....	69
9.5 Циклический режим .....	80
9.6 Индикация ошибок .....	80
10 Система прерываний .....	81
10.1 Контроллер прерываний NVIC .....	86
10.2 Обработка прерываний по уровню и по фронту .....	86
11 Порты ввода-вывода .....	88
11.1 Функционирование порта .....	88
11.2 Режим альтернативных функций .....	89
11.3 Входные фильтры .....	89
11.4 Прерывания .....	90
11.5 Генерация аппаратных запросов .....	91
11.6 Механизм блокировки конфигурации .....	91
11.7 Механизм маскирования .....	92
12 Таймер TMR .....	94
13 Расширенный таймер ETMR .....	95
13.1 Функционирование таймера .....	95
13.2 Режимы счёта .....	96
13.3 Сопряжение с блоком DMA .....	96
14 Блок программируемых логических ячеек LAU .....	97
14.1 Логический блок LUT .....	97
14.2 Логическая макроячейка LM .....	101

14.3	Логический массив LAU .....	103
15	Блок тригонометрических вычислений TMU .....	104
15.1	Вычисление синуса и косинуса .....	105
15.2	Перевод из прямоугольной системы координат в полярную .....	105
15.3	Перевод из полярной системы координат в прямоугольную .....	106
15.4	Поворот вектора на заданный угол .....	106
15.5	Генерация запросов к DMA .....	106
15.6	Генерация исключений .....	106
15.7	Генерация прерываний .....	107
16	Часы реального времени RTC .....	108
16.1	Рекомендации по подключению и трассировке сигналов на печатной плате .....	108
16.2	Функционирование .....	108
16.3	Генерация прерываний .....	109
16.4	Сопряжение с таймерами .....	109
17	Сторожевой таймер WDT .....	110
18	Блок АЦП .....	111
18.1	Секвенсор .....	112
18.2	Модуль АЦП .....	119
18.3	Цифровой компаратор .....	122
18.4	Прерывания .....	125
18.5	Примеры работы блока АЦП .....	127
19	Блоки захвата .....	136
19.1	Режим захвата времени .....	137
19.2	Режим работы «генератор ШИМ» .....	138
19.3	Прерывания .....	140
20	Блоки ШИМ .....	141
20.1	Таймер .....	142
20.2	Компаратор .....	145
20.3	Обработчик событий .....	147
20.4	Пороговый выключатель .....	150
20.5	Генератор задержки ШИМ .....	151
20.6	Фильтр коротких импульсов .....	153
20.7	Модулятор .....	153
20.8	Детектор сигнала аварии .....	155
20.9	Триггер событий .....	156
20.10	ШИМ высокого разрешения .....	160
21	Модуль квадратурного декодера QEP .....	162
21.1	Обработчик сигналов входов .....	162
21.2	Квадратурный преобразователь .....	163
21.3	Счетчик позиции .....	165
21.4	Таймер временных отсчетов .....	170
21.5	Модуль захвата времени .....	170
21.6	Сторожевой таймер блока QEP .....	172
21.7	Прерывания .....	173
22	Сигма-дельта демодулятор SDFM .....	174
22.1	Структура демодулятора .....	174
22.2	Входной декодер .....	175
22.3	Фильтр данных .....	176
22.4	Компаратор .....	179
22.5	Скорость работы и задержка Sinc-фильтра .....	181
22.6	Генерация прерываний .....	182
22.7	Генерация запросов DMA .....	184

23	Приемопередатчик UART .....	185
23.1	Функционирование приемопередатчика .....	185
23.2	Прерывания .....	185
24	Приемопередатчик UART.....	186
24.1	Функционирование блока UART .....	187
24.2	Функционирование ИК передатчика .....	190
24.3	Режим модема .....	191
24.4	Интерфейс прямого доступа к памяти .....	192
24.5	Прерывания .....	193
24.6	Программирование .....	195
25	Контроллер интерфейса CAN 2.0b.....	196
25.1	Протокол CAN .....	196
25.2	Структура и функционирование контроллера CAN.....	202
25.3	Узел CAN.....	208
25.4	Объекты сообщений .....	214
25.5	Прием и передача сообщений.....	217
25.6	Фильтрация сообщений.....	220
25.7	Удаленные запросы .....	221
25.8	Дополнительные режимы передачи .....	222
25.9	FIFO структура объектов сообщений .....	223
25.10	Режим шлюза.....	226
25.11	Прерывания объектов сообщений.....	229
25.12	Программирование контроллера CAN .....	232
26	Контроллер интерфейса Ethernet 10/100 .....	233
26.1	Компоненты дескриптора .....	234
26.2	Операции передачи.....	236
26.3	Операции приема .....	236
27	Контроллер МПИ (по ГОСТ Р 52070–2003) .....	238
27.1	Режимы работы .....	239
27.2	Контроллер шины (КШ).....	239
27.3	Удаленный терминал (УТ).....	248
27.4	Монитор шины (МШ).....	255
28	Контроллер интерфейса I2C .....	257
28.1	Протокол шины.....	257
28.2	Функциональное описание .....	264
28.3	Инициализация и функционирование .....	267
29	Контроллер интерфейса SPI .....	281
29.1	Структура контроллера SPI .....	281
29.2	Интерфейс прямого доступа к памяти .....	283
29.3	Функционирование .....	283
29.4	Прерывания .....	287
30	Контроллер интерфейса SpaceWire .....	289
30.1	Общие сведения .....	289
30.2	Принцип работы.....	289
30.3	Интерфейс связи .....	290
30.4	Распространение системного времени (тайм-кодов) .....	292
30.5	Приёмный канал DMA .....	293
30.6	Передающий канал DMA .....	297
30.7	Аппаратные особенности.....	301
31	Контроллер интерфейса OWI.....	302
31.1	Функционирование интерфейса .....	302
32	Программно-аппаратные средства отладки .....	305



Заключение.....	306
Приложение А (обязательное) Регистры микроконтроллера.....	307
А.1 Регистры блока управления системой SIU.....	307
А.2 Регистры блока управления сбросом и синхронизацией RCU.....	317
А.3 Регистры блока управления энергопотреблением PMU.....	339
А.4 Регистры контроллера основной Flash-памяти.....	343
А.5 Регистры контроллера загрузочной Flash-памяти.....	348
А.6 Регистры контроллера внешней памяти.....	352
А.7 Регистры контроллера прямого доступа к памяти DMA.....	353
А.8 Регистры портов ввода-вывода.....	364
А.9 Регистры таймера TMR.....	387
А.10 Регистры расширенного таймера ETMR.....	391
А.11 Регистры блока программируемых логических ячеек LAU.....	396
А.12 Регистры блока тригонометрических вычислений TMU.....	407
А.13 Регистры часов реального времени RTC.....	414
А.14 Регистры сторожевого таймера WDT.....	419
А.15 Регистры блока АЦП.....	423
А.16 Регистры блоков захвата.....	446
А.17 Регистры блоков ШИМ.....	457
А.18 Регистры модуля квадратурного декодера QEP.....	486
А.19 Регистры сигма-дельта демодулятора SDFM.....	503
А.20 Регистры приемопередатчика TUART.....	515
А.21 Регистры приемопередатчика UART.....	519
А.22 Регистры контроллера интерфейса CAN 2.0b.....	532
А.23 Регистры контроллера интерфейса Ethernet 10/100.....	559
А.24 Регистры контроллера МПИ (по ГОСТ Р 52070–2003).....	581
А.25 Регистры контроллера интерфейса I2C.....	604
А.26 Регистры контроллера интерфейса SPI.....	612
А.27 Регистры контроллера интерфейса SpaceWire.....	618
А.28 Регистры контроллера интерфейса OWI.....	628
Приложение Б (обязательное) Коды состояний функционирования блока I2C.....	631
Приложение В (обязательное) Регистры прерываний.....	639
Лист регистрации изменений.....	641

## **Введение**

32-разрядные микроконтроллеры с каждым днем набирают все большую популярность среди разработчиков различного оборудования и программистов. Они применяются при разработке и изготовлении электронной техники. Высокая вычислительная мощность и при этом относительно низкая стоимость делают эти устройства привлекательными для самого широкого круга разработчиков.

Микросхема 1921BK028 представляет собой СБИС 32-разрядного микроконтроллера на базе ядра ARM Cortex-M4F, предназначенного для промышленных и потребительских приложений, включая системы дистанционного мониторинга, контрольно-измерительные приборы, сетевые устройства, системы автоматизации производственных процессов, автомобильную электронику, системы управления электродвигателями.

В состав микроконтроллера входит широкий набор цифровой и аналоговой периферии, в связи с чем, он может применяться в различных системах цифровой обработки сигналов, в том числе, требующих точных аналогово-цифровых преобразований, в системах управления и сбора информации.

В настоящем техническом описании приведено описание архитектуры, функционального построения и периферии микроконтроллера 1921BK028. Техническое описание может служить практическим руководством по применению микроконтроллера для разработчиков систем на его основе и программистов.

## **1 Область применения и особенности микроконтроллера**

Сфера применения микросхемы 1921BK028 довольно широка: средства измерений, связи, наблюдения, безопасности, автоматизация производства, медицины, энергетики, промышленности, различных систем управления и системы электропривода.

Для эффективного управления в электромеханических системах была разработана дополнительная периферия: блоки ШИМ, блок АЦП с интерфейсом к контроллеру прямого доступа к памяти, модуль захвата/сравнения, блок импульсного квадратурного декодера, используемого для обработки сигналов датчиков положения ротора в высокопроизводительных системах для определения положения, направления и скорости вращения.

Разработанный микроконтроллер имеет встроенную Flash-память программ объемом 2 Мбайт, которую можно использовать для хранения и загрузки пользовательского программного обеспечения.

Также существует дополнительная область размером 512 Кбайт, которая может быть использована для хранения начального загрузчика.

Система тактирования микроконтроллера позволяет использовать различные источники тактового сигнала, что позволяет расширить набор применений и решаемых задач пользователя. Микроконтроллер может тактироваться от внутреннего RC-генератора с частотой 16 МГц, внутреннего осциллятора с внешним кварцевым резонатором, а также сигналом встроенного генератора PLL. Существует возможность гибкой настройки тактовых сигналов для блоков периферии.

Для снижения энергопотребления микросхемы предусмотрена возможность отключения тактовых сигналов отдельных блоков периферии в случае, если они не используются пользователем. При переходе процессора в режим пониженного энергопотребления возможно отключение тактового сигнала ядра (команда WFI или WFE).

## 2 Краткое техническое описание микроконтроллера

### 2.1 Функциональные параметры

Структурная схема микроконтроллера показана на рисунке 2.1.

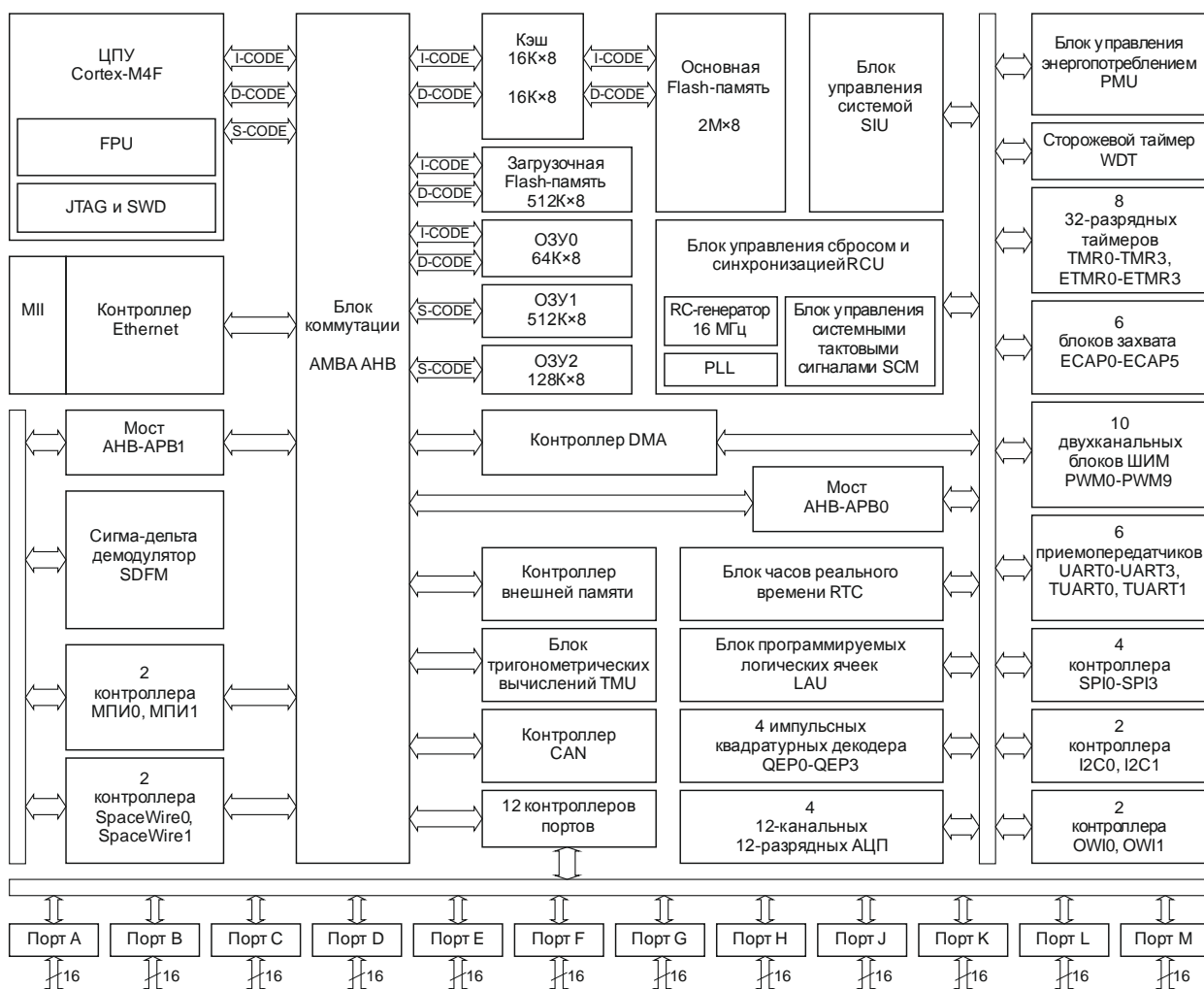


Рисунок 2.1 – Структурная схема микроконтроллера

В состав микроконтроллера входят функциональные элементы:

- 32-разрядное ЦПУ с поддержкой набора одноцикловых команд умножения с накоплением, команд централизованного управления потоком данных, арифметических и логических команд, встроенным модулем обработки команд с плавающей запятой с одинарной точностью FPU, поддержкой отладочных интерфейсов JTAG и SWD;
- блок управления сбросом и синхронизацией RCU, имеющий в своем составе RC-генератор (16 МГц), синтезатор частоты PLL и блок управления системными тактовыми сигналами SCM;
- блок управления системой SIU;
- блок управления энергопотреблением PMU;
- блок коммутации AMBA AHB;
- основная Flash-память объемом 2 Мбайт;
- загрузочная Flash-память объемом 512 Кбайт;
- кэш команд и кэш данных объемом по 16 Кбайт каждый;
- три ОЗУ с объемами: 64 (ОЗУ0), 128 (ОЗУ2) и 512 (ОЗУ1) Кбайт;
- 32-канальный контроллер прямого доступа к памяти DMA;

- контролер внешней памяти;
- блок программируемых логических ячеек LAU;
- блок тригонометрических вычислений TMU;
- сигма-дельта демодулятор SDFM;
- блок часов реального времени RTC;
- сторожевой таймер WDT;
- четыре 12-канальных 12-разрядных АЦП с режимами цифрового компаратора для каждого из каналов и функцией автоматического запуска модулей ШИМ по событию «окончание преобразования»;
- двенадцать 16-разрядных портов ввода-вывода A, B, C, D, E, F, G, H, J, K, L, M;
- восемь 32-разрядных таймеров TMR0 – TMR3, ETMR0 – ETMR3;
- десять двухканальных блоков ШИМ: PWM0 – PWM9;
- шесть блоков захвата ECAP0 – ECAP2;
- четыре импульсных квадратурных декодера для обработки сигналов датчиков положения ротора QEP0 – QEP3;
- шесть приемопередатчиков UART0 – UART3, TUART0, TUART1;
- контроллеры интерфейсов:
  - CAN 2.0b;
  - Ethernet с МП
  - два контроллера МПИ по ГОСТ Р 52070–2003 (МПИ0, МПИ1);
  - два контроллера I2C (I2C0, I2C1);
  - четыре контроллера SPI (SPI0 – SPI3);
  - два контроллера SpaceWire (SpaceWire0, SpaceWire1);
  - два контроллера OWI (OWI0, OWI1).

### **Особенности выводов микроконтроллера**

Все выводы микроконтроллера по их функциональному назначению, организованы в группы:

- 16-разрядные порты ввода-вывода A, B, C, D, E, F, G, H, J, K, L, M;
- порт тестирования JTAG, совмещенный с интерфейсом отладки SWD;
- входные каналы АЦП;
- питание АЦП;
- батарейное питание для блока часов реального времени RTC;
- питание микроконтроллера.

Каждый из выводов портов ввода-вывода является двунаправленным выводом общего назначения с набором альтернативных функций. Режим работы, альтернативная функция, нагрузочная способность и быстродействие, а также подтяжка (к высокому или низкому уровню) и функционирование в режиме выхода с открытым стоком/источком могут быть заданы для каждого вывода независимо от других.

Примечание – После сброса микроконтроллера выводы портов ввода-вывода конфигурируются как выводы общего назначения и находятся в третьем состоянии.

Порт JTAG, предназначенный для внутрисхемного программирования микроконтроллера, тестирования и отладки программ пользователя, включает в свой состав пять выводов TCK, TMS, TDI, TDO, TRST для подключения JTAG эмулятора JEM-NT-CM4.

В случае использования интерфейса SWD его выводы SWCLK, SWDIO и SWO соединяются с выводами TCK, TMS и TDO, соответственно. При этом выводы TDI и TRST удерживаются в состоянии слабой единицы за счет внутренних схем подтяжки.

Выводы ADC\_CH0 – ADC\_CH47 (входные каналы АЦП), не подключенные к внешним источникам напряжения, находятся в плавающем состоянии.

Вывод RESET# является входом сигнала сброса микроконтроллера и должен находиться в состоянии логической единицы. Внешний сброс осуществляется подачей на

вывод логического ноля в течение мс. При внутреннем сбросе вывод удерживается в состоянии логического ноля микроконтроллером.

Микроконтроллер имеет четыре конфигурационных вывода. Назначение этих выводов во время сброса микроконтроллера:

- SERVEN – запрещает (в состоянии логической единицы) любые операции со всей Flash-памятью, кроме полного его стирания;

- BOOT0, BOOT1 – задают режимы загрузки (старта);

- CPE – выбирает источник тактового сигнала (внешний – на выводах XI\_OSC, XO\_OSC или внутренний генератор).

Выводы XI\_RTC и XO\_RTC предназначены для подключения внешнего источника тактового сигнала (с частотой 32 кГц) блока RTC.

Выводы XI\_OSC и XO\_OSC предназначены для подключения внешнего источника тактового сигнала микроконтроллера с частотой (8 – 24) МГц.

Выводы AVDD33 и AGND предназначены для подключения внешнего источника питания АЦП с номинальным значением напряжения 3,3 В.

Вывод AREF предназначен для подключения внешнего источника опорного напряжения АЦП.

Выводы RTC\_BAT и RTC\_GND предназначены для подключения внешнего источника питания (батарейки) блока RTC. Номинальное значение напряжения должно находиться в диапазоне от 1,6 до 3,6 В.

Выводы VDD12, VDD33 и GND предназначены для подключения внешних источников питания ядра и периферии микроконтроллера. Номинальные значения напряжений 1,2 В и 3,3 В.

Примечание – На плате вывод питания AVDD33 может быть объединен с выводами питания VDD33 (при этом должны быть приняты меры для снижения помех по питанию).

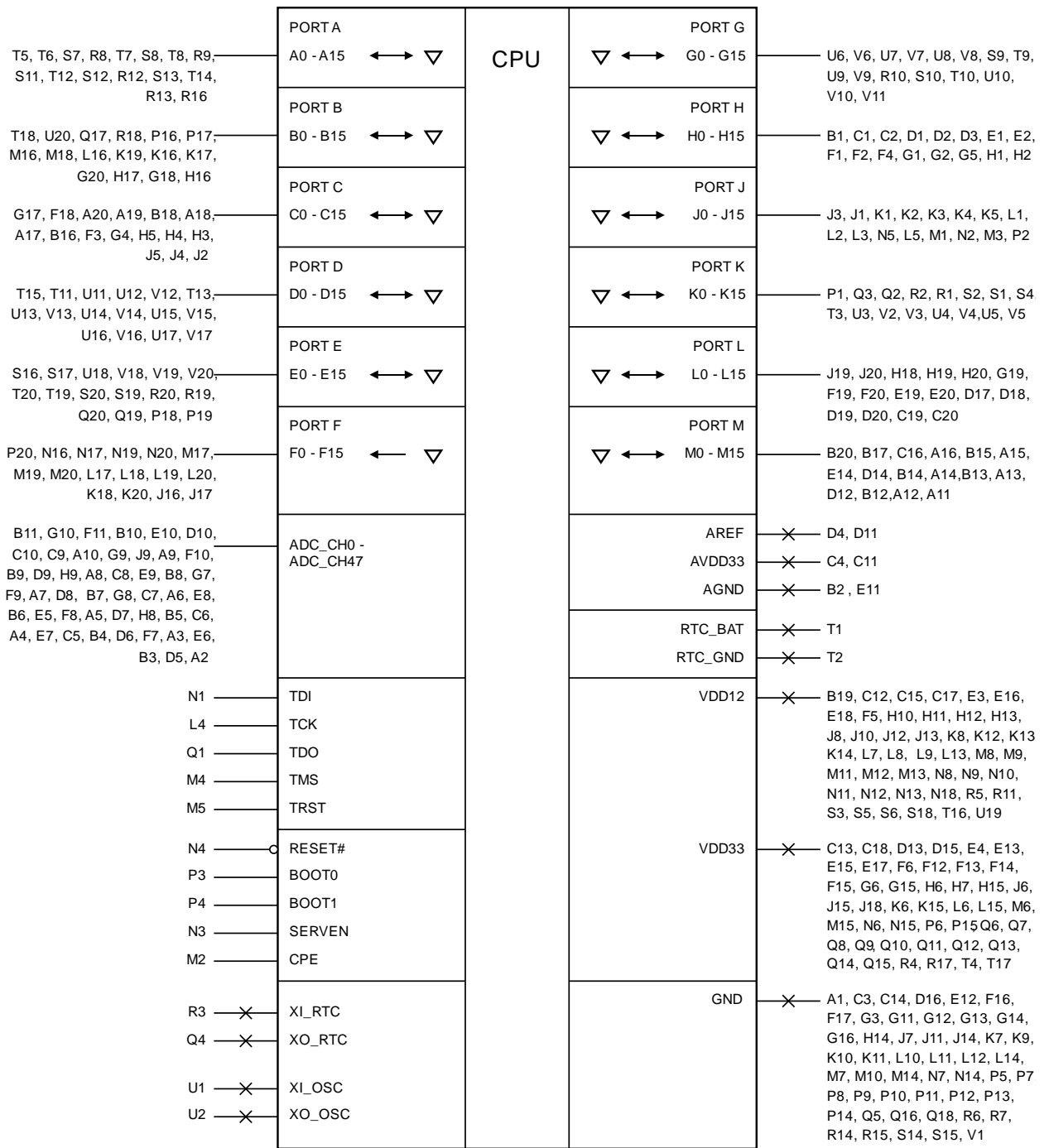
Выключение питания микроконтроллера подразумевает полное снятие напряжения как с выводов питания VDD12, VDD33, AVDD33, так и со всех остальных выводов микроконтроллера.

Примечание – Запрещено подавать напряжение на функциональные выводы при выключенном питании микроконтроллера.

### **УГО и назначение выводов микроконтроллера**

Условное графическое обозначение микроконтроллера приведено на рисунке 2.2.

Функциональное назначение выводов микроконтроллера приведено в таблицах 2.1 – 2.3, в которых приняты обозначения: I – вход, O – выход, I/O – вход/выход, Z – третье состояние.



Примечание – Альтернативные функции выводов портов А, В, С, D, E, F, G, H, J, К, L, М указаны в таблице 2.1.

Рисунок 2.2 – Условное графическое обозначение микросхемы

Таблица 2.1 – Функциональное назначение выводов, имеющих альтернативные функции

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
A0		T5	I/O/Z	Вход/выход порта А, разряд 0
	SPI0_CLK		I/O	Вход/выход синхросигнала SPI0
	I2C1_SCL		I/O	Вход/выход синхронизации I2C1
	QEP3_A		I	Вход А блока QEP3
	LAU_LM0_0		I/O	Вход/выход макроячейки LM0, нулевой
A1		T6	I/O/Z	Вход/выход порта А, разряд 1
	SPI0_RX		I	Вход данных SPI0
	I2C1_SDA		I/O	Вход/выход данных I2C1
	QEP3_B		I	Вход В блока QEP3
	LAU_LM0_1		I/O	Вход/выход макроячейки LM0, первый
A2		S7	I/O/Z	Вход/выход порта А, разряд 2
	SPI0_TX		O	Выход данных SPI0
	TUART0_TX		O	Выход данных TUART0
	QEP3_I		I/O	Вход/выход индексный блока QEP3
	LAU_LM0_2		I/O	Вход/выход макроячейки LM0, второй
A3		R8	I/O/Z	Вход/выход порта А, разряд 3
	SPI0_FSS		I/O	Вход/выход выбора устройства SPI0
	TUART0_RX		I	Вход данных TUART0
	QEP3_S		I/O	Вход/выход стробирования блока QEP3
	LAU_LM0_3		I/O	Вход/выход макроячейки LM0, третий
A4		T7	I/O/Z	Вход/выход порта А, разряд 4
	SDFM0_DATA		I	Вход нулевого бита данных сигма-дельта демодулятора
	TMR2_EXTIN		I	Вход синхронизации таймера 2
	QEP2_A		I	Вход А блока QEP2
	LAU_LM0_4		I/O	Вход/выход макроячейки LM0, четвертый
A5		S8	I/O/Z	Вход/выход порта А, разряд 5
	SDFM0_CLK		I/O	Вход/выход нулевого сигнала синхронизации сигма-дельта демодулятора
	CAP_PWM3		I/O	Вход/выход третьего блока захвата
	QEP2_B		I	Вход В блока QEP2
	LAU_LM0_5		I/O	Вход/выход макроячейки LM0, пятый
A6		T8	I/O/Z	Вход/выход порта А, разряд 6
	SDFM1_DATA		I	Вход первого бита данных сигма-дельта демодулятора
	CAP_PWM4		I/O	Вход/выход четвертого блока захвата
	QEP2_I		I/O	Вход/выход индексный блока QEP2
	LAU_LM0_6		I/O	Вход/выход макроячейки LM0, шестой
A7		R9	I/O/Z	Вход/выход порта А, разряд 7
	SDFM1_CLK		I/O	Вход/выход первого сигнала синхронизации сигма-дельта демодулятора
	CAP_PWM5		I/O	Вход/выход пятого блока захвата
	QEP2_S		I/O	Вход/выход стробирования блока QEP2
	LAU_LM0_7		I/O	Вход/выход макроячейки LM0, седьмой



Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
A8		S11	I/O/Z	Вход/выход порта А, разряд 8
	SDFM2_DATA		I	Вход второго бита данных сигма-дельта демодулятора
	QEP1_A		I	Вход А блока QEP1
	SW1_DIN		I	Вход данных SpaceWire1
	LAU_LM1_0		I/O	Вход/выход макроячейки LM1, нулевой
A9		T12	I/O/Z	Вход/выход порта А, разряд 9
	SDFM2_CLK		I/O	Вход/выход второго сигнала синхронизации сигма-дельта демодулятора
	QEP1_B		I	Вход В блока QEP1
	SW1_SIN		I	Вход синхронизации SpaceWire1
	LAU_LM1_1		I/O	Вход/выход макроячейки LM1, первый
A10		S12	I/O/Z	Вход/выход порта А, разряд 10
	SDFM3_DATA		I	Вход третьего бита данных сигма-дельта демодулятора
	QEP1_I		I/O	Вход/выход индексный блока QEP1
	SW1_DOUT		O	Выход данных SpaceWire1
	LAU_LM1_2		I/O	Вход/выход макроячейки LM1, второй
A11		R12	I/O/Z	Вход/выход порта А, разряд 11
	SDFM3_CLK		I/O	Вход/выход третьего сигнала синхронизации сигма-дельта демодулятора
	QEP1_S		I/O	Вход/выход стробирования блока QEP1
	SW1_SOUT		O	Выход синхронизации SpaceWire1
	LAU_LM1_3		I/O	Вход/выход макроячейки LM1, третий
A12		S13	I/O/Z	Вход/выход порта А, разряд 12
	QEP0_A		I	Вход А блока QEP0
	CAP_PWM0		I/O	Вход/выход нулевого блока захвата
	TMR0_EXTIN		I	Вход синхронизации таймера 0
	LAU_LM1_4		I/O	Вход/выход макроячейки LM1, четвертый
A13		T14	I/O/Z	Вход/выход порта А, разряд 13
	QEP0_B		I	Вход В блока QEP0
	CAP_PWM1		I/O	Вход/выход первого блока захвата
	TMR1_EXTIN		I	Вход синхронизации таймера 1
	LAU_LM1_5		I/O	Вход/выход макроячейки LM1, пятый
A14		R13	I/O/Z	Вход/выход порта А, разряд 14
	QEP0_I		I/O	Вход/выход индексный блока QEP0
	CAP_PWM2		I/O	Вход/выход второго блока захвата
	MII_MDIO		I/O	Вход/выход управляющих данных Ethernet
	LAU_LM1_6		I/O	Вход/выход макроячейки LM1, шестой
A15		R16	I/O/Z	Вход/выход порта А, разряд 15
	QEP0_S		I/O	Вход/выход стробирования блока QEP0
	CAP_PWM3		I/O	Вход/выход третьего блока захвата
	MII_MDC		O	Выход синхросигнала управляющих данных Ethernet
	LAU_LM1_7		I/O	Вход/выход макроячейки LM1, седьмой

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
B0		T18	I/O/Z	Вход/выход порта В, разряд 0
	CAP_PWM0		I/O	Вход/выход нулевого блока захвата
	M0_PARITY		I	Вход четности адреса удаленного терминала МПИО
	MII_RXCLK		I	Вход синхросигнала приема Ethernet
	LAU_LM2_0		I/O	Вход/выход макроячейки LM2, нулевой
B1		U20	I/O/Z	Вход/выход порта В, разряд 1
	CAP_PWM1		I/O	Вход/выход первого блока захвата
	M0_ADDR0		I	Вход адреса удаленного терминала МПИО
	MII_RXDV		I	Вход валидности принимаемых данных Ethernet
	LAU_LM2_1		I/O	Вход/выход макроячейки LM2, первый
B2		Q17	I/O/Z	Вход/выход порта В, разряд 2
	CAP_PWM2		I/O	Вход/выход второго блока захвата
	M0_ADDR1		I	Вход адреса удаленного терминала МПИО
	MII_RXER		I	Вход ошибки приема Ethernet
	LAU_LM2_2		I/O	Вход/выход макроячейки LM2, второй
B3		R18	I/O/Z	Вход/выход порта В, разряд 3
	PWM_TZ0		I	Вход сигнала аварии, нулевой
	M0_ADDR2		I	Вход адреса удаленного терминала МПИО
	MII_TXCLK		I	Вход синхросигнала передачи Ethernet
	LAU_LM2_3		I/O	Вход/выход макроячейки LM2, третий
B4		P16	I/O/Z	Вход/выход порта В, разряд 4
	PWM_TZ1		I	Вход сигнала аварии, первый
	M0_ADDR3		I	Вход адреса удаленного терминала МПИО
	MII_TXEN		O	Выход разрешения передачи Ethernet
	LAU_LM2_4		I/O	Вход/выход макроячейки LM2, четвертый
B5		P17	I/O/Z	Вход/выход порта В, разряд 5
	PWM_TZ2		I	Вход сигнала аварии, второй
	M0_ADDR4		I	Вход адреса удаленного терминала МПИО
	MII_TXER		O	Выход ошибки передачи Ethernet
	LAU_LM2_5		I/O	Вход/выход макроячейки LM2, пятый
B6		M16	I/O/Z	Вход/выход порта В, разряд 6
	PWM0_A		O	Выход А блока ШИМО
	M0_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине А
	MII_CRS		I	Вход контроля носителя Ethernet
	LAU_LM2_6		I/O	Вход/выход макроячейки LM2, шестой
B7		M18	I/O/Z	Вход/выход порта В, разряд 7
	PWM0_B		O	Выход В блока ШИМО
	M0_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине В
	MII_COL		I	Вход обнаружения конфликта Ethernet
	LAU_LM2_7		I/O	Вход/выход макроячейки LM2, седьмой

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
B8		L16	I/O/Z	Вход/выход порта В, разряд 8
	PWM1_A		O	Выход А блока ШИМ1
	M0_A_TX		O	Выход передатчика МПИО, шина А
	MII_TXD0		O	Выход нулевого бита данных Ethernet
	LAU_LM3_0		I/O	Вход/выход макроячейки LM3, нулевой
B9		K19	I/O/Z	Вход/выход порта В, разряд 9
	PWM1_B		O	Выход В блока ШИМ1
	M0_A_TXN		O	Выход передатчика МПИО инверсный, шина А
	MII_TXD1		O	Выход первого бита данных Ethernet
	LAU_LM3_1		I/O	Вход/выход макроячейки LM3, первый
B10		K16	I/O/Z	Вход/выход порта В, разряд 10
	PWM2_A		O	Выход А блока ШИМ2
	M0_A_RX		I	Вход приемника МПИО, шина А
	MII_TXD2		O	Выход второго бита данных Ethernet
	LAU_LM3_2		I/O	Вход/выход макроячейки LM3, второй
B11		K17	I/O/Z	Вход/выход порта В, разряд 11
	PWM2_B		O	Выход В блока ШИМ2
	M0_A_RXN		I	Вход приемника МПИО инверсный, шина А
	MII_TXD3		O	Выход третьего бита данных Ethernet
	LAU_LM3_3		I/O	Вход/выход макроячейки LM3, третий
B12		G20	I/O/Z	Вход/выход порта В, разряд 12
	PWM3_A		O	Выход А блока ШИМ3
	M0_B_TX		O	Выход передатчика МПИО, шина В
	MII_RXD0		I	Вход нулевого бита данных Ethernet
	LAU_LM3_4		I/O	Вход/выход макроячейки LM3, четвертый
B13		H17	I/O/Z	Вход/выход порта В, разряд 13
	PWM3_B		O	Выход В блока ШИМ3
	M0_B_TXN		O	Выход передатчика МПИО инверсный, шина В
	MII_RXD1		I	Вход первого бита данных Ethernet
	LAU_LM3_5		I/O	Вход/выход макроячейки LM3, пятый
B14		G18	I/O/Z	Вход/выход порта В, разряд 14
	PWM4_A		O	Выход А блока ШИМ4
	M0_B_RX		I	Вход приемника МПИО, шина В
	MII_RXD2		I	Вход второго бита данных Ethernet
	LAU_LM3_6		I/O	Вход/выход макроячейки LM3, шестой
B15		H16	I/O/Z	Вход/выход порта В, разряд 15
	PWM4_B		O	Выход В блока ШИМ4
	M0_B_RXN		I	Вход приемника МПИО инверсный, шина В
	MII_RXD3		I	Вход третьего бита данных Ethernet
	LAU_LM3_7		I/O	Вход/выход макроячейки LM3, седьмой

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
C0		G17	I/O/Z	Вход/выход порта C, разряд 0
	PWM5_A		O	Выход А блока ШИМ5
	SW0_DIN		I	Вход данных SpaceWire0
	SPI2_CLK		I/O	Вход/выход синхросигнала SPI2
	M1_A_TX		O	Выход передатчика МПИ1, шина А
C1		F18	I/O/Z	Вход/выход порта C, разряд 1
	PWM5_B		O	Выход В блока ШИМ5
	SW0_SIN		I	Вход синхронизации SpaceWire0
	SPI2_TX		O	Выход данных SPI2
	M1_A_TXN		O	Выход передатчика МПИ1 инверсный, шина А
C2		A20	I/O/Z	Вход/выход порта C, разряд 2
	PWM6_A		O	Выход А блока ШИМ6
	SW0_DOUT		O	Выход данных SpaceWire0
	SPI2_RX		I	Вход данных SPI2
	M1_A_RX		I	Вход приемника МПИ1, шина А
C3		A19	I/O/Z	Вход/выход порта C, разряд 3
	PWM6_B		O	Выход В блока ШИМ6
	SW0_SOUT		O	Выход синхронизации SpaceWire0
	SPI2_FSS		I/O	Вход/выход выбора устройства SPI2
	M1_A_RXN		I	Вход приемника МПИ1 инверсный, шина А
C4		B18	I/O/Z	Вход/выход порта C, разряд 4
	PWM7_A		O	Выход А блока ШИМ7
	SPI1_FSS		I/O	Вход/выход выбора устройства SPI1
	UART0_RTS		O	Выход сигнала запроса на передачу UART0
	M1_B_TX		O	Выход передатчика МПИ1, шина В
C5		A18	I/O/Z	Вход/выход порта C, разряд 5
	PWM7_B		O	Выход В блока ШИМ7
	SPI1_CLK		I/O	Вход/выход синхронизации SPI1
	UART0_DTR		O	Выход сигнала готовности приемника данных UART0
	M1_B_TXN		O	Выход передатчика МПИ1 инверсный, шина В
C6		A17	I/O/Z	Вход/выход порта C, разряд 6
	PWM8_A		O	Выход А блока ШИМ8
	SPI1_TX		O	Выход данных SPI1
	UART0_CTS		I	Вход сигнала готовности к приему UART0
	M1_B_RX		I	Вход приемника МПИ1, шина В
C7		B16	I/O/Z	Вход/выход порта C, разряд 7
	PWM8_B		O	Выход В блока ШИМ8
	SPI1_RX		I	Вход данных SPI1
	UART0_DSR		I	Вход сигнала готовности источника данных UART0
	M1_B_RXN		I	Вход приемника МПИ1 инверсный, шина В

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
C8		F3	I/O/Z	Вход/выход порта C, разряд 8
	I2C0_SCL		I/O	Вход/выход синхронизации I2C0
	UART2_TX		O	Выход данных UART2
	UART0_DCD		I	Вход отклика при обнаружении информационного сигнала UART0
	M1_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине А
C9		G4	I/O/Z	Вход/выход порта C, разряд 9
	I2C0_SDA		I/O	Вход/выход данных I2C0
	UART2_RX		I	Вход данных UART2
	UART0_RI		I	Вход сигнала-индикатора вызова UART0
	M1_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине В
C10		H5	I/O/Z	Вход/выход порта C, разряд 10
	UART0_TX		O	Выход данных UART0
	TMR2_EXTIN		I	Вход синхронизации таймера 2
	NMI		I	Вход внешнего немаскируемого прерывания (активная единица)
	M1_PARITY		I	Вход четности адреса удаленного терминала МПИ1
C11		H4	I/O/Z	Вход/выход порта C, разряд 11
	UART0_RX		I	Вход данных UART0
	TMR3_EXTIN		I	Вход синхронизации таймера 3
	PLL_EXTSRC		I	Вход внешнего синхросигнала для PLL
	M1_ADDR0		I	Вход адреса удаленного терминала МПИ1
C12		H3	I/O/Z	Вход/выход порта C, разряд 12
	CAN0_TX		O	Выход данных CAN0
	TUART1_TX		O	Выход данных TUART1
	SPI3_CLK		I/O	Вход/выход синхронизации SPI3
	M1_ADDR1		I	Вход адреса удаленного терминала МПИ1
C13		J5	I/O/Z	Вход/выход порта C, разряд 13
	CAN0_RX		I	Вход данных CAN0
	TUART1_RX		I	Вход данных TUART1
	SPI3_RX		I	Вход данных SPI3
	M1_ADDR2		I	Вход адреса удаленного терминала МПИ1
C14		J4	I/O/Z	Вход/выход порта C, разряд 14
	CAN1_TX		O	Выход данных CAN1
	UART1_TX		O	Выход данных UART1
	SPI3_TX		O	Выход данных SPI3
	M1_ADDR3		I	Вход адреса удаленного терминала МПИ1
C15		J2	I/O/Z	Вход/выход порта C, разряд 15
	CAN1_RX		I	Вход данных CAN1
	UART1_RX		I	Вход данных UART1
	SPI3_FSS		I/O	Вход/выход выбора устройства SPI3
	M1_ADDR4		I	Вход адреса удаленного терминала МПИ1

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
D0		T15	I/O/Z	Вход/выход порта D, разряд 0
	OWI0_DATA		I/O	Вход/выход данных OneWire0
	CLKOUT		O	Выход системного тактового сигнала
	UART0_SIR		O	Выход данных в режиме ИК-порта
	LAU_CLK0		I	Вход нулевого синхросигнала блока LAU
D1		T11	I/O/Z	Вход/выход порта D, разряд 1
	PWM9_A		O	Выход А блока ШИМ9
	I2C1_SCL		I/O	Вход/выход синхронизации I2C1
	QEP3_A		I	Вход А блока QEP3
	LAU_CLK1		I	Вход первого синхросигнала блока LAU
D2		U11	I/O/Z	Вход/выход порта D, разряд 2
	PWM9_B		O	Выход В блока ШИМ9
	I2C1_SDA		I/O	Вход/выход данных I2C1
	QEP3_B		I	Вход В блока QEP3
	LAU_CLK2		I	Вход второго синхросигнала блока LAU
D3		U12	I/O/Z	Вход/выход порта D, разряд 3
	PWM_SYNCI		I	Вход внешнего синхросигнала для блоков ШИМ
	OWI1_DATA		I/O	Вход/выход данных OneWire1
	QEP3_I		I/O	Вход/выход индексный блока QEP3
	LAU_CLK3		I	Вход третьего синхросигнала блока LAU
D4		V12	I/O/Z	Вход/выход порта D, разряд 4
	MEM_CE0		O	Выход нулевого сигнала выбора внешней памяти
	PWM_TZ3		I	Вход третьего сигнала аварии ШИМ
	QEP3_S		I/O	Вход/выход стробирования блока QEP3
	TRACE_EXTCLK		I	Вход внешнего синхросигнала блока ETM
D5		T13	I/O/Z	Вход/выход порта D, разряд 5
	MEM_CE1		O	Выход первого сигнала выбора внешней памяти
	PWM_TZ4		I	Вход четвертого сигнала аварии ШИМ
	UART1_RTS		O	Выход сигнала запроса на передачу UART1
	TRACE_CLK		O	Выход синхросигнала блока ETM
D6		U13	I/O/Z	Вход/выход порта D, разряд 6
	MEM_CE2		O	Выход второго сигнала выбора внешней памяти
	PWM_TZ5		I	Вход пятого сигнала аварии ШИМ
	UART1_DTR		O	Выход сигнала готовности приемника данных UART1
	TRACE_DATA0		O	Выход нулевого бита данных блока ETM
D7		V13	I/O/Z	Вход/выход порта D, разряд 7
	MEM_CE3		O	Выход третьего сигнала выбора внешней памяти
	CAN0_TX		O	Выход данных CAN0
	UART1_CTS		I	Вход сигнала готовности к приему UART1
	TRACE_DATA1		O	Выход первого бита данных блока ETM

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
D8		U14	I/O/Z	Вход/выход порта D, разряд 8
	MEM_CE4		O	Выход четвертого сигнала выбора внешней памяти
	CAN0_RX		I	Вход данных CAN0
	UART1_DSR		I	Вход сигнала готовности источника данных UART1
	TRACE_DATA2		O	Выход второго бита данных блока ETM
D9		V14	I/O/Z	Вход/выход порта D, разряд 9
	MEM_CE5		O	Выход пятого сигнала выбора внешней памяти
	CAN1_TX		O	Выход данных CAN1
	UART1_DCD		I	Вход отклика при обнаружении информационного сигнала UART1
	TRACE_DATA3		O	Выход третьего бита данных блока ETM
D10		U15	I/O/Z	Вход/выход порта D, разряд 10
	MEM_CE6		O	Выход шестого сигнала выбора внешней памяти
	CAN1_RX		I	Вход данных CAN1
	UART1_RI		I	Вход сигнала-индикатора вызова UART1
	CAP_PWM0		I/O	Вход/выход нулевого блока захвата
D11		V15	I/O/Z	Вход/выход порта D, разряд 11
	MEM_CE7		O	Выход седьмого сигнала выбора внешней памяти
	–		–	–
	UART1_SIR		O	Выход данных UART1 в режиме ИК-порта
	CAP_PWM1		I/O	Вход/выход первого блока захвата
D12		U16	I/O/Z	Вход/выход порта D, разряд 12
	MEM_LB		O	Выход сигнала записи младшего байта во внешнюю память
	TUART0_TX		O	Выход данных TUART0
	PWM0_A		O	Выход А блока ШИМ0
	CAP_PWM2		I/O	Вход/выход второго блока захвата
D13		V16	I/O/Z	Вход/выход порта D, разряд 13
	MEM_UB		O	Выход сигнала записи старшего байта во внешнюю память
	TUART0_RX		I	Вход данных TUART0
	PWM0_B		O	Выход В блока ШИМ0
	CAP_PWM3		I/O	Вход/выход третьего блока захвата
D14		U17	I/O/Z	Вход/выход порта D, разряд 14
	MEM_WE		O	Выход сигнала разрешения записи во внешнюю память
	MIIMDIO		I/O	Вход/выход управляющих данных Ethernet
	PWM1_A		O	Выход А блока ШИМ1
	CAP_PWM4		I/O	Вход/выход четвертого блока захвата
D15		V17	I/O/Z	Вход/выход порта D, разряд 15
	MEM_OE		O	Выход сигнала разрешения чтения из внешней памяти
	MIIMDC		O	Выход синхросигнала управляющих данных Ethernet
	PWM1_B		O	Выход В блока ШИМ1
	CAP_PWM5		I/O	Вход/выход пятого блока захвата

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
E0		S16	I/O/Z	Вход/выход порта E, разряд 0
	MEM_DATA0		I/O	Вход/выход шины данных внешней памяти, разряд 0
	MII_TXD0		O	Выход нулевого бита данных Ethernet
	M0_PARITY		I	Вход четности адреса удаленного терминала МПИО
	UART2_RTS		O	Выход сигнала запроса на передачу UART2
E1		S17	I/O/Z	Вход/выход порта E, разряд 1
	MEM_DATA1		I/O	Вход/выход шины данных внешней памяти, разряд 1
	MII_TXD1		O	Выход первого бита данных Ethernet
	M0_ADDR0		I	Вход адреса удаленного терминала МПИО
	UART2_DTR		O	Выход сигнала готовности приемника данных UART2
E2		U18	I/O/Z	Вход/выход порта E, разряд 2
	MEM_DATA2		I/O	Вход/выход шины данных внешней памяти, разряд 2
	MII_TXD2		O	Выход второго бита данных Ethernet
	M0_ADDR1		I	Вход адреса удаленного терминала МПИО
	UART2_CTS		I	Вход сигнала готовности к приему UART2
E3		V18	I/O/Z	Вход/выход порта E, разряд 3
	MEM_DATA3		I/O	Вход/выход шины данных внешней памяти, разряд 3
	MII_TXD3		O	Выход третьего бита данных Ethernet
	M0_ADDR2		I	Вход адреса удаленного терминала МПИО
	UART2_DSR		I	Вход сигнала готовности источника данных UART2
E4		V19	I/O/Z	Вход/выход порта E, разряд 4
	MEM_DATA4		I/O	Вход/выход шины данных внешней памяти, разряд 4
	MII_RXD0		I	Вход нулевого бита данных Ethernet
	M0_ADDR3		I	Вход адреса удаленного терминала МПИО
	UART2_DCD		I	Вход отклика при обнаружении информационного сигнала UART2
E5		V20	I/O/Z	Вход/выход порта E, разряд 5
	MEM_DATA5		I/O	Вход/выход шины данных внешней памяти, разряд 5
	MII_RXD1		I	Вход первого бита данных Ethernet
	M0_ADDR4		I	Вход адреса удаленного терминала МПИО
	UART2_RI		I	Вход сигнала-индикатора вызова UART2
E6		T20	I/O/Z	Вход/выход порта E, разряд 6
	MEM_DATA6		I/O	Вход/выход шины данных внешней памяти, разряд 6
	MII_RXD2		I	Вход второго бита данных Ethernet
	M0_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине А
	UART2_SIR		O	Выход данных UART2 в режиме ИК-порта
E7		T19	I/O/Z	Вход/выход порта E, разряд 7
	MEM_DATA7		I/O	Вход/выход шины данных внешней памяти, разряд 7
	MII_RXD3		I	Вход третьего бита данных Ethernet
	M0_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине В
	UART3_SIR		O	Выход данных UART3 в режиме ИК-порта



Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
E8		S20	I/O/Z	Вход/выход порта E, разряд 8
	MEM_DATA8		I/O	Вход/выход шины данных внешней памяти, разряд 8
	MII_RXCLK		I	Вход синхросигнала приема Ethernet
	M0_A_TX		O	Выход передатчика МППО, шина А
	UART3_RX		I	Вход данных UART3
E9		S19	I/O/Z	Вход/выход порта E, разряд 9
	MEM_DATA9		I/O	Вход/выход шины данных внешней памяти, разряд 9
	MII_RXDV		I	Вход валидности принимаемых данных Ethernet
	M0_A_TXN		O	Выход передатчика МППО инверсный, шина А
	UART3_TX		O	Выход данных UART3
E10		R20	I/O/Z	Вход/выход порта E, разряд 10
	MEM_DATA10		I/O	Вход/выход шины данных внешней памяти, разряд 10
	MII_RXER		I	Вход ошибки приема Ethernet
	M0_A_RX		I	Вход приемника МППО, шина А
	UART3_RTS		O	Выход сигнала запроса на передачу UART3
E11		R19	I/O/Z	Вход/выход порта E, разряд 11
	MEM_DATA11		I/O	Вход/выход шины данных внешней памяти, разряд 11
	MII_TXCLK		I	Вход синхросигнала передачи Ethernet
	M0_A_RXN		I	Вход приемника МППО инверсный, шина А
	UART3_DTR		O	Выход сигнала готовности приемника данных UART3
E12		Q20	I/O/Z	Вход/выход порта E, разряд 12
	MEM_DATA12		I/O	Вход/выход шины данных внешней памяти, разряд 12
	MII_TXEN		O	Выход разрешения передачи Ethernet
	M0_B_TX		O	Выход передатчика МППО на шине В
	UART3_CTS		I	Вход сигнала готовности к приему UART3
E13		Q19	I/O/Z	Вход/выход порта E, разряд 13
	MEM_DATA13		I/O	Вход/выход шины данных внешней памяти, разряд 13
	MII_TXER		O	Выход ошибки передачи Ethernet
	M0_B_TXN		O	Выход передатчика МППО инверсный, шина В
	UART3_DSR		I	Вход сигнала готовности источника данных UART3
E14		P18	I/O/Z	Вход/выход порта E, разряд 14
	MEM_DATA14		I/O	Вход/выход шины данных внешней памяти, разряд 14
	MII_CRS		I	Вход контроля носителя Ethernet
	M0_B_RX		I	Вход приемника МППО, шина В
	UART3_DCD		I	Вход отклика при обнаружении информационного сигнала UART3
E15		P19	I/O/Z	Вход/выход порта E, разряд 15
	MEM_DATA15		I/O	Вход/выход шины данных внешней памяти, разряд 15
	MII_COL		I	Вход обнаружения конфликта Ethernet
	M0_B_RXN		I	Вход приемника МППО инверсный, шина В
	UART3_RI		I	Вход сигнала-индикатора вызова UART3

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
F0		P20	I/O/Z	Вход/выход порта F, разряд 0
	MEM_ADDR0		O	Выход шины адреса внешней памяти, разряд 0
	M1_A_TX		O	Выход передатчика МПИ1, шина А
	SPI0_CLK		I/O	Вход/выход синхросигнала SPI0
	UART0_RTS		O	Выход сигнала запроса на передачу UART0
F1		N16	I/O/Z	Вход/выход порта F, разряд 1
	MEM_ADDR1		O	Выход шины адреса внешней памяти, разряд 1
	M1_A_TXN		O	Выход передатчика МПИ1 инверсный, шина А
	SPI0_TX		O	Выход данных SPI0
	UART0_DTR		O	Выход сигнала готовности приемника данных UART0
F2		N17	I/O/Z	Вход/выход порта F, разряд 2
	MEM_ADDR2		O	Выход шины адреса внешней памяти, разряд 2
	M1_A_RX		I	Вход приемника МПИ1, шина А
	SPI0_RX		I	Вход данных SPI0
	UART0_CTS		I	Вход сигнала готовности к приему UART0
F3		N19	I/O/Z	Вход/выход порта F, разряд 3
	MEM_ADDR3		O	Выход шины адреса внешней памяти, разряд 3
	M1_A_RXN		I	Вход приемника МПИ1 инверсный, шина А
	SPI0_FSS		I/O	Вход/выход выбора устройства SPI0
	UART0_DSR		I	Вход сигнала готовности источника данных UART0
F4		N20	I/O/Z	Вход/выход порта F, разряд 4
	MEM_ADDR4		O	Выход шины адреса внешней памяти, разряд 4
	M1_B_TX		O	Выход передатчика МПИ1, шина В
	SPI1_CLK		I/O	Вход/выход синхросигнала SPI1
	UART0_DCD		I	Вход отклика при обнаружении информационного сигнала UART0
F5		M17	I/O/Z	Вход/выход порта F, разряд 5
	MEM_ADDR5		O	Выход шины адреса внешней памяти, разряд 5
	M1_B_TXN		O	Выход передатчика МПИ1 инверсный, шина В
	SPI1_TX		O	Выход данных SPI1
	UART0_RI		I	Вход сигнала-индикатора вызова UART0
F6		M19	I/O/Z	Вход/выход порта F, разряд 6
	MEM_ADDR6		O	Выход шины адреса внешней памяти, разряд 6
	M1_B_RX		I	Вход приемника МПИ1, шина В
	SPI1_RX		I	Вход данных SPI1
	UART0_RX		I	Вход данных UART0
F7		M20	I/O/Z	Вход/выход порта F, разряд 7
	MEM_ADDR7		O	Выход шины адреса внешней памяти, разряд 7
	M1_B_RXN		I	Вход приемника МПИ1 инверсный, шина В
	SPI1_FSS		I/O	Вход/выход выбора устройства SPI1
	UART0_TX		O	Выход данных UART0

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
F8		L17	I/O/Z	Вход/выход порта F, разряд 8
	MEM_ADDR8		O	Выход шины адреса внешней памяти, разряд 8
	M1_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине А
	SPI2_CLK		I/O	Вход/выход синхросигнала SPI2
	UART0_SIR		O	Выход данных UART0 в режиме ИК-порта
F9		L18	I/O/Z	Вход/выход порта F, разряд 9
	MEM_ADDR9		O	Выход шины адреса внешней памяти, разряд 9
	M1_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине В
	SPI2_TX		O	Выход данных SPI2
	UART1_SIR		O	Выход данных UART1 в режиме ИК-порта
F10		L19	I/O/Z	Вход/выход порта F, разряд 10
	MEM_ADDR10		O	Выход шины адреса внешней памяти, разряд 10
	M1_PARITY		I	Вход четности адреса удаленного терминала МПИ1
	SPI2_RX		I	Вход данных SPI2
	TUART1_TX		O	Выход данных TUART1
F11		L20	I/O/Z	Вход/выход порта F, разряд 11
	MEM_ADDR11		O	Выход шины адреса внешней памяти, разряд 11
	M1_ADDR0		I	Вход адреса удаленного терминала МПИ1
	SPI2_FSS		I/O	Вход/выход выбора устройства SPI2
	TUART1_RX		I	Вход данных TUART1
F12		K18	I/O/Z	Вход/выход порта F, разряд 12
	MEM_ADDR12		O	Выход шины адреса внешней памяти, разряд 12
	M1_ADDR1		I	Вход адреса удаленного терминала МПИ1
	SPI3_CLK		I/O	Вход/выход синхросигнала SPI3
	QEP2_A		I	Вход А блока QEP2
F13		K20	I/O/Z	Вход/выход порта F, разряд 13
	MEM_ADDR13		O	Выход шины адреса внешней памяти, разряд 13
	M1_ADDR2		I	Вход адреса удаленного терминала МПИ1
	SPI3_TX		O	Выход данных SPI3
	QEP2_B		I	Вход В блока QEP2
F14		J16	I/O/Z	Вход/выход порта F, разряд 14
	MEM_ADDR14		O	Выход шины адреса внешней памяти, разряд 14
	M1_ADDR3		I	Вход адреса удаленного терминала МПИ1
	SPI3_RX		I	Вход данных SPI3
	QEP2_I		I/O	Вход/выход индексный блока QEP2
F15		J17	I/O/Z	Вход/выход порта F, разряд 15
	MEM_ADDR15		O	Выход шины адреса внешней памяти, разряд 15
	M1_ADDR4		I	Вход адреса удаленного терминала МПИ1
	SPI3_FSS		I/O	Вход/выход выбора устройства SPI3
	QEP2_S		I/O	Вход/выход стробирования блока QEP2

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
G0		U6	I/O/Z	Вход/выход порта G, разряд 0
	MEM_ADDR16		O	Выход шины адреса внешней памяти, разряд 16
	CAP_PWM0		I/O	Вход/выход нулевого блока захвата
	QEP0_A		I	Вход А блока QEP0
	SDFM0_DATA		I	Вход нулевого бита данных сигма-дельта демодулятора
G1		V6	I/O/Z	Вход/выход порта G, разряд 1
	MEM_ADDR17		O	Выход шины адреса внешней памяти, разряд 17
	CAP_PWM1		I/O	Вход/выход первого блока захвата
	QEP0_B		I	Вход В блока QEP0
	SDFM0_CLK		I/O	Вход/выход нулевого сигнала синхронизации сигма-дельта демодулятора
G2		U7	I/O/Z	Вход/выход порта G, разряд 2
	MEM_ADDR18		O	Выход шины адреса внешней памяти, разряд 18
	CAP_PWM2		I/O	Вход/выход второго блока захвата
	QEP0_I		I/O	Вход/выход индексный блока QEP0
	SDFM1_DATA		I	Вход первого бита данных сигма-дельта демодулятора
G3		V7	I/O/Z	Вход/выход порта G, разряд 3
	MEM_ADDR19		O	Выход шины адреса внешней памяти, разряд 19
	CAP_PWM3		I/O	Вход/выход третьего блока захвата
	QEP0_S		I/O	Вход/выход стробирования блока QEP0
	SDFM1_CLK		I/O	Вход/выход первого сигнала синхронизации сигма-дельта демодулятора
G4		U8	I/O/Z	Вход/выход порта G, разряд 4
	MEM_ADDR20		O	Выход шины адреса внешней памяти, разряд 20
	CAP_PWM4		I/O	Вход/выход четвертого блока захвата
	QEP1_A		I	Вход А блока QEP1
	SDFM2_DATA		I	Вход второго бита данных сигма-дельта демодулятора
G5		V8	I/O/Z	Вход/выход порта G, разряд 5
	MEM_ADDR21		O	Выход шины адреса внешней памяти, разряд 21
	CAP_PWM5		I/O	Вход/выход пятого блока захвата
	QEP1_B		I	Вход В блока QEP1
	SDFM2_CLK		I/O	Вход/выход второго сигнала синхронизации сигма-дельта демодулятора
G6		S9	I/O/Z	Вход/выход порта G, разряд 6
	MEM_ADDR22		O	Выход шины адреса внешней памяти, разряд 22
	I2C0_SCL		I/O	Вход/выход синхронизации I2C0
	QEP1_I		I/O	Вход/выход индексный блока QEP1
	SDFM3_DATA		I	Вход третьего бита данных сигма-дельта демодулятора
G7		T9	I/O/Z	Вход/выход порта G, разряд 7
	MEM_ADDR23		O	Выход шины адреса внешней памяти, разряд 23
	I2C0_SDA		I/O	Вход/выход данных I2C0
	QEP1_S		I/O	Вход/выход стробирования блока QEP1
	SDFM3_CLK		I/O	Вход/выход третьего сигнала синхронизации сигма-дельта демодулятора

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
G8		U9	I/O/Z	Вход/выход порта G, разряд 8
	TUART0_TX		O	Выход данных TUART0
	MEM_CE0		O	Выход нулевого сигнала выбора внешней памяти
	SW0_DIN		I	Вход данных SpaceWire0
	LAU_CLK0		I	Вход нулевого синхросигнала блока LAU
G9		V9	I/O/Z	Вход/выход порта G, разряд 9
	TUART0_RX		I	Вход данных TUART0
	MEM_CE1		O	Выход первого сигнала выбора внешней памяти
	SW0_SIN		I	Вход синхронизации SpaceWire0
	LAU_CLK1		I	Вход первого синхросигнала блока LAU
G10		R10	I/O/Z	Вход/выход порта G, разряд 10
	I2C1_SCL		I/O	Вход/выход синхронизации I2C1
	MEM_CE2		O	Выход второго сигнала выбора внешней памяти
	SW0_DOUT		O	Выход данных SpaceWire0
	LAU_CLK2		I	Вход второго синхросигнала блока LAU
G11		S10	I/O/Z	Вход/выход порта G, разряд 11
	I2C1_SDA		I/O	Вход/выход данных I2C1
	MEM_CE3		O	Выход третьего сигнала выбора внешней памяти
	SW0_SOUT		O	Выход синхронизации SpaceWire0
	LAU_CLK3		I	Вход третьего синхросигнала блока LAU
G12		T10	I/O/Z	Вход/выход порта G, разряд 12
	TMR0_EXTIN		I	Вход синхронизации таймера 0
	MEM_CE4		O	Выход четвертого сигнала выбора внешней памяти
	SW1_DIN		I	Вход данных SpaceWire1
	CLKOUT		O	Выход системного тактового сигнала
G13		U10	I/O/Z	Вход/выход порта G, разряд 13
	TMR1_EXTIN		I	Вход синхронизации таймера 1
	MEM_CE5		O	Выход пятого сигнала выбора внешней памяти
	SW1_SIN		I	Вход синхронизации SpaceWire1
	NMI		I	Вход внешнего немаскируемого прерывания (активная единица)
G14		V10	I/O/Z	Вход/выход порта G, разряд 14
	MII_MDIO		I/O	Вход/выход управляющих данных Ethernet
	MEM_CE6		O	Выход шестого сигнала выбора внешней памяти
	SW1_DOUT		O	Выход данных SpaceWire1
	PLL_EXTSRC		I	Вход внешнего синхросигнала для PLL
G15		V11	I/O/Z	Вход/выход порта G, разряд 15
	MII_MDC		O	Выход синхросигнала управляющих данных Ethernet
	MEM_CE7		O	Выход седьмого сигнала выбора внешней памяти
	SW1_SOUT		O	Выход синхронизации SpaceWire1
	PWM_SYNCI		I	Вход внешнего синхросигнала для блоков ШИМ

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
Н0		В1	I/O/Z	Вход/выход порта Н, разряд 0
	MI1_TXD0		O	Выход нулевого бита данных Ethernet
	MEM_DATA0		I/O	Вход/выход шины данных внешней памяти, разряд 0
	SPI0_CLK		I/O	Вход/выход синхросигнала SPI0
	PWM0_A		O	Выход А блока ШИМ0
Н1		С1	I/O/Z	Вход/выход порта Н, разряд 1
	MI1_TXD1		O	Выход первого бита данных Ethernet
	MEM_DATA1		I/O	Вход/выход шины данных внешней памяти, разряд 1
	SPI0_TX		O	Выход данных SPI0
	PWM0_B		O	Выход В блока ШИМ0
Н2		С2	I/O/Z	Вход/выход порта Н, разряд 2
	MI1_TXD2		O	Выход второго бита данных Ethernet
	MEM_DATA2		I/O	Вход/выход шины данных внешней памяти, разряд 2
	SPI0_RX		I	Вход данных SPI0
	PWM1_A		O	Выход А блока ШИМ1
Н3		D1	I/O/Z	Вход/выход порта Н, разряд 3
	MI1_TXD3		O	Выход третьего бита данных Ethernet
	MEM_DATA3		I/O	Вход/выход шины данных внешней памяти, разряд 3
	SPI0_FSS		I/O	Вход/выход выбора устройства SPI0
	PWM1_B		O	Выход В блока ШИМ1
Н4		D2	I/O/Z	Вход/выход порта Н, разряд 4
	MI1_RXD0		I	Вход нулевого бита данных Ethernet
	MEM_DATA4		I/O	Вход/выход шины данных внешней памяти, разряд 4
	SPI1_CLK		I/O	Вход/выход синхросигнала SPI1
	PWM2_A		O	Выход А блока ШИМ2
Н5		D3	I/O/Z	Вход/выход порта Н, разряд 5
	MI1_RXD1		I	Вход первого бита данных Ethernet
	MEM_DATA5		I/O	Вход/выход шины данных внешней памяти, разряд 5
	SPI1_TX		O	Выход данных SPI1
	PWM2_B		O	Выход В блока ШИМ2
Н6		E1	I/O/Z	Вход/выход порта Н, разряд 6
	MI1_RXD2		I	Вход второго бита данных Ethernet
	MEM_DATA6		I/O	Вход/выход шины данных внешней памяти, разряд 6
	SPI1_RX		I	Вход данных SPI1
	PWM3_A		O	Выход А блока ШИМ3
Н7		E2	I/O/Z	Вход/выход порта Н, разряд 7
	MI1_RXD3		I	Вход третьего бита данных Ethernet
	MEM_DATA7		I/O	Вход/выход шины данных внешней памяти, разряд 7
	SPI1_FSS		I/O	Вход/выход выбора устройства SPI1
	PWM3_B		O	Выход В блока ШИМ3

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
H8		F1	I/O/Z	Вход/выход порта H, разряд 8
	MII_RXCLK		I	Вход синхросигнала приема Ethernet
	MEM_DATA8		I/O	Вход/выход шины данных внешней памяти, разряд 8
	SPI2_CLK		I/O	Вход/выход синхросигнала SPI2
	PWM4_A		O	Выход А блока ШИМ4
H9		F2	I/O/Z	Вход/выход порта H, разряд 9
	MII_RXDV		I	Вход валидности принимаемых данных Ethernet
	MEM_DATA9		I/O	Вход/выход шины данных внешней памяти, разряд 9
	SPI2_TX		O	Выход данных SPI2
	PWM4_B		O	Выход В блока ШИМ4
H10		F4	I/O/Z	Вход/выход порта H, разряд 10
	MII_RXER		I	Вход ошибки приема Ethernet
	MEM_DATA10		I/O	Вход/выход шины данных внешней памяти, разряд 10
	SPI2_RX		I	Вход данных SPI2
	PWM5_A		O	Выход А блока ШИМ5
H11		G1	I/O/Z	Вход/выход порта H, разряд 11
	MII_TXCLK		I	Вход синхросигнала передачи Ethernet
	MEM_DATA11		I/O	Вход/выход шины данных внешней памяти, разряд 11
	SPI2_FSS		I/O	Вход/выход выбора устройства SPI2
	PWM5_B		O	Выход В блока ШИМ5
H12		G2	I/O/Z	Вход/выход порта H, разряд 12
	MII_TXEN		O	Выход разрешения передачи Ethernet
	MEM_DATA12		I/O	Вход/выход шины данных внешней памяти, разряд 12
	CAN0_TX		O	Выход данных CAN0
	PWM6_A		O	Выход А блока ШИМ6
H13		G5	I/O/Z	Вход/выход порта H, разряд 13
	MII_TXER		O	Выход ошибки передачи Ethernet
	MEM_DATA13		I/O	Вход/выход шины данных внешней памяти, разряд 13
	CAN0_RX		I	Вход данных CAN0
	PWM6_B		O	Выход В блока ШИМ6
H14		H1	I/O/Z	Вход/выход порта H, разряд 14
	MII_CRIS		I	Вход контроля носителя Ethernet
	MEM_DATA14		I/O	Вход/выход шины данных внешней памяти, разряд 14
	CAN1_TX		O	Выход данных CAN1
	PWM7_A		O	Выход А блока ШИМ7
H15		H2	I/O/Z	Вход/выход порта H, разряд 15
	MII_COL		I	Вход обнаружения конфликта Ethernet
	MEM_DATA15		I/O	Вход/выход шины данных внешней памяти, разряд 15
	CAN1_RX		I	Вход данных CAN1
	PWM7_B		O	Выход В блока ШИМ7

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
J0		J3	I/O/Z	Вход/выход порта J, разряд 0
	MO_PARITY		I	Вход четности адреса удаленного терминала МПИО
	MEM_ADDR0		O	Выход шины адреса внешней памяти, разряд 0
	LAU_LM0_0		I/O	Вход/выход макроячейки LM0, нулевой
	PWM8_A		O	Выход А блока ШИМ8
J1		J1	I/O/Z	Вход/выход порта J, разряд 1
	MO_ADDR0		I	Вход адреса удаленного терминала МПИО
	MEM_ADDR1		O	Выход шины адреса внешней памяти, разряд 1
	LAU_LM0_1		I/O	Вход/выход макроячейки LM0, первый
	PWM8_B		O	Выход В блока ШИМ8
J2		K1	I/O/Z	Вход/выход порта J, разряд 2
	MO_ADDR1		I	Вход адреса удаленного терминала МПИО
	MEM_ADDR2		O	Выход шины адреса внешней памяти, разряд 2
	LAU_LM0_2		I/O	Вход/выход макроячейки LM0, второй
	PWM9_A		O	Выход А блока ШИМ9
J3		K2	I/O/Z	Вход/выход порта J, разряд 3
	MO_ADDR2		I	Вход адреса удаленного терминала МПИО
	MEM_ADDR3		O	Выход шины адреса внешней памяти, разряд 3
	LAU_LM0_3		I/O	Вход/выход макроячейки LM0, третий
	PWM9_B		O	Выход В блока ШИМ9
J4		K3	I/O/Z	Вход/выход порта J, разряд 4
	MO_ADDR3		I	Вход адреса удаленного терминала МПИО
	MEM_ADDR4		O	Выход шины адреса внешней памяти, разряд 4
	LAU_LM0_4		I/O	Вход/выход макроячейки LM0, четвертый
	OW11_DATA		I/O	Вход/выход данных OneWire1
J5		K4	I/O/Z	Вход/выход порта J, разряд 5
	MO_ADDR4		I	Вход адреса удаленного терминала МПИО
	MEM_ADDR5		O	Выход шины адреса внешней памяти, разряд 5
	LAU_LM0_5		I/O	Вход/выход макроячейки LM0, пятый
	TMR3_EXTIN		I	Вход синхронизации таймера 3
J6		K5	I/O/Z	Вход/выход порта J, разряд 6
	MO_A_TX		O	Выход передатчика МПИО, шина А
	MEM_ADDR6		O	Выход шины адреса внешней памяти, разряд 6
	LAU_LM0_6		I/O	Вход/выход макроячейки LM0, шестой
	PWM_TZ0		I	Вход нулевого сигнала аварии ШИМ
J7		L1	I/O/Z	Вход/выход порта J, разряд 7
	MO_A_TXN		O	Выход передатчика МПИО инверсный, шина А
	MEM_ADDR7		O	Выход шины адреса внешней памяти, разряд 7
	LAU_LM0_7		I/O	Вход/выход макроячейки LM0, седьмой
	PWM_TZ1		I	Вход первого сигнала аварии ШИМ



Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
J8		L2	I/O/Z	Вход/выход порта J, разряд 8
	M0_A_RX		I	Вход приемника МПИО, шина А
	MEM_ADDR8		O	Выход шины адреса внешней памяти, разряд 8
	LAU_LM1_0		I/O	Вход/выход макроячейки LM1, нулевой
	PWM_TZ2		I	Вход второго сигнала аварии ШИМ
J9		L3	I/O/Z	Вход/выход порта J, разряд 9
	M0_A_RXN		I	Вход приемника МПИО инверсный, шина А
	MEM_ADDR9		O	Выход шины адреса внешней памяти, разряд 9
	LAU_LM1_1		I/O	Вход/выход макроячейки LM1, первый
	PWM_TZ3		I	Вход третьего сигнала аварии ШИМ
J10		N5	I/O/Z	Вход/выход порта J, разряд 10
	M0_B_TX		O	Выход передатчика МПИО, шина В
	MEM_ADDR10		O	Выход шины адреса внешней памяти, разряд 10
	LAU_LM1_2		I/O	Вход/выход макроячейки LM1, второй
	PWM_TZ4		I	Вход четвертого сигнала аварии ШИМ
J11		L5	I/O/Z	Вход/выход порта J, разряд 11
	M0_B_TXN		O	Выход передатчика МПИО инверсный, шина В
	MEM_ADDR11		O	Выход шины адреса внешней памяти, разряд 11
	LAU_LM1_3		I/O	Вход/выход макроячейки LM1, третий
	PWM_TZ5		I	Вход пятого сигнала аварии ШИМ
J12		M1	I/O/Z	Вход/выход порта J, разряд 12
	M0_B_RX		I	Вход приемника МПИО, шина В
	MEM_ADDR12		O	Выход шины адреса внешней памяти, разряд 12
	LAU_LM1_4		I/O	Вход/выход макроячейки LM1, четвертый
	UART1_RTS		O	Выход сигнала запроса на передачу UART1
J13		N2	I/O/Z	Вход/выход порта J, разряд 13
	M0_B_RXN		I	Вход приемника МПИО инверсный, шина В
	MEM_ADDR13		O	Выход шины адреса внешней памяти, разряд 13
	LAU_LM1_5		I/O	Вход/выход макроячейки LM1, пятый
	UART1_DTR		O	Выход сигнала готовности приемника данных UART1
J14		M3	I/O/Z	Вход/выход порта J, разряд 14
	M0_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине А
	MEM_ADDR14		O	Выход шины адреса внешней памяти, разряд 14
	LAU_LM1_6		I/O	Вход/выход макроячейки LM1, шестой
	UART1_CTS		I	Вход сигнала готовности к приему UART1
J15		P2	I/O/Z	Вход/выход порта J, разряд 15
	M0_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИО на шине В
	MEM_ADDR15		O	Выход шины адреса внешней памяти, разряд 15
	LAU_LM1_7		I/O	Вход/выход макроячейки LM1, седьмой
	UART1_DSR		I	Вход сигнала готовности источника данных UART1

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
K0		P1	I/O/Z	Вход/выход порта К, разряд 0
	M1_A_TX		O	Выход передатчика МПИ1, шина А
	MEM_ADDR16		O	Выход шины адреса внешней памяти, разряд 16
	LAU_LM2_0		I/O	Вход/выход макроячейки LM2, нулевой
	UART1_DCD		I	Вход отклика при обнаружении информационного сигнала UART1
K1		Q3	I/O/Z	Вход/выход порта К, разряд 1
	M1_A_TXN		O	Выход передатчика МПИ1 инверсный, шина А
	MEM_ADDR17		O	Выход шины адреса внешней памяти, разряд 17
	LAU_LM2_1		I/O	Вход/выход макроячейки LM2, первый
	UART1_RI		I	Вход сигнала-индикатора вызова UART1
K2		Q2	I/O/Z	Вход/выход порта К, разряд 2
	M1_A_RX		I	Вход приемника МПИ1, шина А
	MEM_ADDR18		O	Выход шины адреса внешней памяти, разряд 18
	LAU_LM2_2		I/O	Вход/выход макроячейки LM2, второй
	UART2_RTS		O	Выход сигнала запроса на передачу UART2
K3		R2	I/O/Z	Вход/выход порта К, разряд 3
	M1_A_RXN		I	Вход приемника МПИ1 инверсный, шина А
	MEM_ADDR19		O	Выход шины адреса внешней памяти, разряд 19
	LAU_LM2_3		I/O	Вход/выход макроячейки LM2, третий
	UART2_DTR		O	Выход сигнала готовности приемника данных UART2
K4		R1	I/O/Z	Вход/выход порта К, разряд 4
	M1_B_TX		O	Выход передатчика МПИ1, шина В
	MEM_ADDR20		O	Выход шины адреса внешней памяти, разряд 20
	LAU_LM2_4		I/O	Вход/выход макроячейки LM2, четвертый
	UART2_CTS		I	Вход сигнала готовности к приему UART2
K5		S2	I/O/Z	Вход/выход порта К, разряд 5
	M1_B_TXN		O	Выход передатчика МПИ1 инверсный, шина В
	MEM_ADDR21		O	Выход шины адреса внешней памяти, разряд 21
	LAU_LM2_5		I/O	Вход/выход макроячейки LM2, пятый
	UART2_DSR		I	Вход сигнала готовности источника данных UART2
K6		S1	I/O/Z	Вход/выход порта К, разряд 6
	M1_B_RX		I	Вход приемника МПИ1, шина В
	MEM_ADDR22		O	Выход шины адреса внешней памяти, разряд 22
	LAU_LM2_6		I/O	Вход/выход макроячейки LM2, шестой
	UART2_DCD		I	Вход отклика при обнаружении информационного сигнала UART2
K7		S4	I/O/Z	Вход/выход порта К, разряд 7
	M1_B_RXN		I	Вход приемника МПИ1 инверсный, шина В
	MEM_ADDR23		O	Выход шины адреса внешней памяти, разряд 23
	LAU_LM2_7		I/O	Вход/выход макроячейки LM2, седьмой
	UART2_RI		I	Вход сигнала-индикатора вызова UART2

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
K8		T3	I/O/Z	Вход/выход порта K, разряд 8
	M1_ADDR0		I	Вход адреса удаленного терминала МПИ1
	MEM_WE		O	Выход сигнала разрешения записи во внешнюю память
	LAU_LM3_0		I/O	Вход/выход макроячейки LM3, нулевой
	UART1_RX		I	Вход данных UART1
K9		U3	I/O/Z	Вход/выход порта K, разряд 9
	M1_ADDR1		I	Вход адреса удаленного терминала МПИ1
	MEM_OE		O	Выход сигнала разрешения чтения из внешней памяти
	LAU_LM3_1		I/O	Вход/выход макроячейки LM3, первый
	UART1_TX		O	Выход данных UART1
K10		V2	I/O/Z	Вход/выход порта K, разряд 10
	M1_ADDR2		I	Вход адреса удаленного терминала МПИ1
	MEM_LB		O	Выход сигнала записи младшего байта во внешнюю память
	LAU_LM3_2		I/O	Вход/выход макроячейки LM3, второй
	UART2_RX		I	Вход данных UART2
K11		V3	I/O/Z	Вход/выход порта K, разряд 11
	M1_ADDR3		I	Вход адреса удаленного терминала МПИ1
	MEM_UB		O	Выход сигнала записи старшего байта во внешнюю память
	LAU_LM3_3		I/O	Вход/выход макроячейки LM3, третий
	UART2_TX		O	Выход данных UART2
K12		U4	I/O/Z	Вход/выход порта K, разряд 12
	M1_ADDR4		I	Вход адреса удаленного терминала МПИ1
	NMI		I	Вход внешнего немаскируемого прерывания (активная единица)
	LAU_LM3_4		I/O	Вход/выход макроячейки LM3, четвертый
	UART2_SIR		O	Выход данных UART2 в режиме ИК-порта
K13		V4	I/O/Z	Вход/выход порта K, разряд 13
	M1_A_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине А
	OWI0_DATA		I/O	Вход/выход данных OneWire0
	LAU_LM3_5		I/O	Вход/выход макроячейки LM3, пятый
	UART3_SIR		O	Выход данных UART3 в режиме ИК-порта
K14		U5	I/O/Z	Вход/выход порта K, разряд 14
	M1_B_TXEN		O	Выход сигнала разрешения работы передатчика МПИ1 на шине В
	CLKOUT		O	Выход системного тактового сигнала
	LAU_LM3_6		I/O	Вход/выход макроячейки LM3, шестой
	UART3_RX		I	Вход данных UART3
K15		V5	I/O/Z	Вход/выход порта K, разряд 15
	M1_PARITY		I	Вход четности адреса удаленного терминала МПИ1
	PLL_EXTSRC		I	Вход внешнего синхросигнала для PLL
	LAU_LM3_7		I/O	Вход/выход макроячейки LM3, седьмой
	UART3_TX		O	Выход данных UART3

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
L0		J19	I/O/Z	Вход/выход порта L, разряд 0
	TRACE_CLK		O	Выход синхросигнала блока ETM
	PWM0_A		O	Выход А блока ШИМ0
	LAU_LM4_0		I/O	Вход/выход макроячейки LM4, нулевой
	UART3_RTS		O	Выход сигнала запроса на передачу UART3
L1		J20	I/O/Z	Вход/выход порта L, разряд 1
	TRACE_DATA0		O	Выход нулевого бита данных блока ETM
	PWM0_B		O	Выход В блока ШИМ0
	LAU_LM4_1		I/O	Вход/выход макроячейки LM4, первый
	UART3_DTR		O	Выход сигнала готовности приемника данных UART3
L2		H18	I/O/Z	Вход/выход порта L, разряд 2
	TRACE_DATA1		O	Выход первого бита данных блока ETM
	PWM1_A		O	Выход А блока ШИМ1
	LAU_LM4_2		I/O	Вход/выход макроячейки LM4, второй
	UART3_CTS		I	Вход сигнала готовности к приему UART3
L3		H19	I/O/Z	Вход/выход порта L, разряд 3
	TRACE_DATA2		O	Выход второго бита данных блока ETM
	PWM1_B		O	Выход В блока ШИМ1
	LAU_LM4_3		I/O	Вход/выход макроячейки LM4, третий
	UART3_DSR		I	Вход сигнала готовности источника данных UART3
L4		H20	I/O/Z	Вход/выход порта L, разряд 4
	TRACE_DATA3		O	Выход третьего бита данных блока ETM
	PWM2_A		O	Выход А блока ШИМ2
	LAU_LM4_4		I/O	Вход/выход макроячейки LM4, четвертый
	UART3_DCD		I	Вход отклика при обнаружении информационного сигнала UART3
L5		G19	I/O/Z	Вход/выход порта L, разряд 5
	TRACE_EXTCLK		I	Вход внешнего синхросигнала блока ETM
	PWM2_B		O	Выход В блока ШИМ2
	LAU_LM4_5		I/O	Вход/выход макроячейки LM4, пятый
	UART3_RI		I	Вход сигнала-индикатора вызова UART3
L6		F19	I/O/Z	Вход/выход порта L, разряд 6
	TMR0_EXTIN		I	Вход синхронизации таймера 0
	PWM3_A		O	Выход А блока ШИМ3
	LAU_LM4_6		I/O	Вход/выход макроячейки LM4, шестой
	I2C0_SCL		I/O	Вход/выход синхронизации I2C0
L7		F20	I/O/Z	Вход/выход порта L, разряд 7
	TMR1_EXTIN		I	Вход синхронизации таймера 1
	PWM3_B		O	Выход В блока ШИМ3
	LAU_LM4_7		I/O	Вход/выход макроячейки LM4, седьмой
	I2C0_SDA		I/O	Вход/выход данных I2C0

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
L8		E19	I/O/Z	Вход/выход порта L, разряд 8
	TUART0_TX		O	Выход данных TUART0
	PWM4_A		O	Выход А блока ШИМ4
	LAU_LM5_0		I/O	Вход/выход макроячейки LM5, нулевой
	I2C1_SCL		I/O	Вход/выход синхронизации I2C1
L9		E20	I/O/Z	Вход/выход порта L, разряд 9
	TUART0_RX		I	Вход данных TUART0
	PWM4_B		O	Выход В блока ШИМ4
	LAU_LM5_1		I/O	Вход/выход макроячейки LM5, первый
	I2C1_SDA		I/O	Вход/выход данных I2C1
L10		D17	I/O/Z	Вход/выход порта L, разряд 10
	TMR2_EXTIN		I	Вход синхронизации таймера 2
	PWM5_A		O	Выход А блока ШИМ5
	LAU_LM5_2		I/O	Вход/выход макроячейки LM5, второй
	TUART1_TX		O	Выход данных TUART1
L11		D18	I/O/Z	Вход/выход порта L, разряд 11
	TMR3_EXTIN		I	Вход синхронизации таймера 3
	PWM5_B		O	Выход В блока ШИМ5
	LAU_LM5_3		I/O	Вход/выход макроячейки LM5, третий
	TUART1_RX		I	Вход данных TUART1
L12		D19	I/O/Z	Вход/выход порта L, разряд 12
	QEP3_A		I	Вход А блока QEP3
	PWM6_A		O	Выход А блока ШИМ6
	LAU_LM5_4		I/O	Вход/выход макроячейки LM5, четвертый
	SW0_DIN		I	Вход данных SpaceWire0
L13		D20	I/O/Z	Вход/выход порта L, разряд 13
	QEP3_B		I	Вход В блока QEP3
	PWM6_B		O	Выход В блока ШИМ6
	LAU_LM5_5		I/O	Вход/выход макроячейки LM5, пятый
	SW0_SIN		I	Вход синхронизации SpaceWire0
L14		C19	I/O/Z	Вход/выход порта L, разряд 14
	QEP3_I		I/O	Вход/выход индексный блока QEP3
	PWM7_A		O	Выход А блока ШИМ7
	LAU_LM5_6		I/O	Вход/выход макроячейки LM5, шестой
	SW0_DOUT		O	Выход данных SpaceWire0
L15		C20	I/O/Z	Вход/выход порта L, разряд 15
	QEP3_S		I/O	Вход/выход стробирования блока QEP3
	PWM7_B		O	Выход В блока ШИМ7
	LAU_LM5_7		I/O	Вход/выход макроячейки LM5, седьмой
	SW0_SOUT		O	Выход синхронизации SpaceWire0

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
M0		B20	I/O/Z	Вход/выход порта M, разряд 0
	QEP0_A		I	Вход А блока QEP0
	PWM8_A		O	Выход А блока ШИМ8
	LAU_LM6_0		I/O	Вход/выход макроячейки LM6, нулевой
	SDFM0_DATA		I	Вход нулевого бита данных сигма-дельта демодулятора
M1		B17	I/O/Z	Вход/выход порта M, разряд 1
	QEP0_B		I	Вход В блока QEP0
	PWM8_B		O	Выход В блока ШИМ8
	LAU_LM6_1		I/O	Вход/выход макроячейки LM6, первый
	SDFM0_CLK		I/O	Вход/выход нулевого сигнала синхронизации сигма-дельта демодулятора
M2		C16	I/O/Z	Вход/выход порта M, разряд 2
	QEP0_I		I/O	Вход/выход индексный блока QEP0
	PWM9_A		O	Выход А блока ШИМ9
	LAU_LM6_2		I/O	Вход/выход макроячейки LM6, второй
	SDFM1_DATA		I	Вход первого бита данных сигма-дельта демодулятора
M3		A16	I/O/Z	Вход/выход порта M, разряд 3
	QEP0_S		I/O	Вход/выход стробирования блока QEP0
	PWM9_B		O	Выход В блока ШИМ9
	LAU_LM6_3		I/O	Вход/выход макроячейки LM6, третий
	SDFM1_CLK		I/O	Вход/выход первого сигнала синхронизации сигма-дельта демодулятора
M4		B15	I/O/Z	Вход/выход порта M, разряд 4
	QEP1_A		I	Вход А блока QEP1
	PWM_SYNCI		I	Вход внешнего синхросигнала для блоков ШИМ
	LAU_LM6_4		I/O	Вход/выход макроячейки LM6, четвертый
	SDFM2_DATA		I	Вход второго бита данных сигма-дельта демодулятора
M5		A15	I/O/Z	Вход/выход порта M, разряд 5
	QEP1_B		I	Вход В блока QEP1
	CAP_PWM0		I/O	Вход/выход нулевого блока захвата
	LAU_LM6_5		I/O	Вход/выход макроячейки LM6, пятый
	SDFM2_CLK		I/O	Вход/выход второго сигнала синхронизации сигма-дельта демодулятора
M6		E14	I/O/Z	Вход/выход порта M, разряд 6
	QEP1_I		I/O	Вход/выход индексный блока QEP1
	CAP_PWM1		I/O	Вход/выход первого блока захвата
	LAU_LM6_6		I/O	Вход/выход макроячейки LM6, шестой
	SDFM3_DATA		I	Вход третьего бита данных сигма-дельта демодулятора
M7		D14	I/O/Z	Вход/выход порта M, разряд 7
	QEP1_S		I/O	Вход/выход стробирования блока QEP1
	CAP_PWM2		I/O	Вход/выход второго блока захвата
	LAU_LM6_7		I/O	Вход/выход макроячейки LM6, седьмой
	SDFM3_CLK		I/O	Вход/выход третьего сигнала синхронизации сигма-дельта демодулятора

Продолжение таблицы 2.1

Вывод корпуса				Функциональное назначение вывода
Обозначение	Альтернативная функция	Номер	Тип	
M8		B14	I/O/Z	Вход/выход порта M, разряд 8
	QEP2_A		I	Вход А блока QEP2
	CAP_PWM3		I/O	Вход/выход третьего блока захвата
	LAU_LM7_0		I/O	Вход/выход макроячейки LM7, нулевой
	CLKOUT		O	Выход системного тактового сигнала
M9		A14	I/O/Z	Вход/выход порта M, разряд 9
	QEP2_B		I	Вход В блока QEP2
	CAP_PWM4		I/O	Вход/выход четвертого блока захвата
	LAU_LM7_1		I/O	Вход/выход макроячейки LM7, первый
	PLL_EXTSRC		I	Вход внешнего синхросигнала для PLL
M10		B13	I/O/Z	Вход/выход порта M, разряд 10
	QEP2_I		I/O	Вход/выход индексный блока QEP2
	CAP_PWM5		I/O	Вход/выход пятого блока захвата
	LAU_LM7_2		I/O	Вход/выход макроячейки LM7, второй
	PWM_TZ0		I	Вход нулевого сигнала аварии ШИМ
M11		A13	I/O/Z	Вход/выход порта M, разряд 11
	QEP2_S		I/O	Вход/выход стробирования блока QEP2
	–		–	–
	LAU_LM7_3		I/O	Вход/выход макроячейки LM7, третий
	PWM_TZ1		I	Вход первого сигнала аварии ШИМ
M12		D12	I/O/Z	Вход/выход порта M, разряд 12
	SW1_DIN		I	Вход данных SpaceWire1
	CAN0_TX		O	Выход данных CAN0
	LAU_LM7_4		I/O	Вход/выход макроячейки LM7, четвертый
	PWM_TZ2		I	Вход второго сигнала аварии ШИМ
M13		B12	I/O/Z	Вход/выход порта M, разряд 13
	SW1_SIN		I	Вход синхронизации SpaceWire1
	CAN0_RX		I	Вход данных CAN0
	LAU_LM7_5		I/O	Вход/выход макроячейки LM7, пятый
	PWM_TZ3		I	Вход третьего сигнала аварии ШИМ
M14		A12	I/O/Z	Вход/выход порта M, разряд 14
	SW1_DOUT		O	Выход данных SpaceWire1
	CAN1_TX		O	Выход данных CAN1
	LAU_LM7_6		I/O	Вход/выход макроячейки LM7, шестой
	PWM_TZ4		I	Вход четвертого сигнала аварии ШИМ
M15		A11	I/O/Z	Вход/выход порта M, разряд 15
	SW1_SOUT		O	Выход синхронизации SpaceWire1
	CAN1_RX		I	Вход данных CAN1
	LAU_LM7_7		I/O	Вход/выход макроячейки LM7, седьмой
	PWM_TZ5		I	Вход пятого сигнала аварии ШИМ

Таблица 2.2 – Функциональное назначение выводов, не имеющих альтернативных функций

Вывод корпуса			Функциональное назначение вывода
Обозначение	Номер	Тип	
ADC_CH0	B11	I	Вход АЦПО, канал 0
ADC_CH1	G10	I	Вход АЦПО, канал 1
ADC_CH2	F11	I	Вход АЦПО, канал 2
ADC_CH3	B10	I	Вход АЦПО, канал 3
ADC_CH4	E10	I	Вход АЦПО, канал 4
ADC_CH5	D10	I	Вход АЦПО, канал 5
ADC_CH6	C10	I	Вход АЦПО, канал 6
ADC_CH7	C9	I	Вход АЦПО, канал 7
ADC_CH8	A10	I	Вход АЦПО, канал 8
ADC_CH9	G9	I	Вход АЦПО, канал 9
ADC_CH10	J9	I	Вход АЦПО, канал 10
ADC_CH11	A9	I	Вход АЦПО, канал 11
ADC_CH12	F10	I	Вход АЦП1, канал 12
ADC_CH13	B9	I	Вход АЦП1, канал 13
ADC_CH14	D9	I	Вход АЦП1, канал 14
ADC_CH15	H9	I	Вход АЦП1, канал 15
ADC_CH16	A8	I	Вход АЦП1, канал 16
ADC_CH17	C8	I	Вход АЦП1, канал 17
ADC_CH18	E9	I	Вход АЦП1, канал 18
ADC_CH19	B8	I	Вход АЦП1, канал 19
ADC_CH20	G7	I	Вход АЦП1, канал 20
ADC_CH21	F9	I	Вход АЦП1, канал 21
ADC_CH22	A7	I	Вход АЦП1, канал 22
ADC_CH23	D8	I	Вход АЦП1, канал 23
ADC_CH24	B7	I	Вход АЦП2, канал 24
ADC_CH25	G8	I	Вход АЦП2, канал 25
ADC_CH26	C7	I	Вход АЦП2, канал 26
ADC_CH27	A6	I	Вход АЦП2, канал 27
ADC_CH28	E8	I	Вход АЦП2, канал 28
ADC_CH29	B6	I	Вход АЦП2, канал 29
ADC_CH30	E5	I	Вход АЦП2, канал 30
ADC_CH31	F8	I	Вход АЦП2, канал 31
ADC_CH32	A5	I	Вход АЦП2, канал 32
ADC_CH33	D7	I	Вход АЦП2, канал 33
ADC_CH34	H8	I	Вход АЦП2, канал 34
ADC_CH35	B5	I	Вход АЦП2, канал 35
ADC_CH36	C6	I	Вход АЦП3, канал 36
ADC_CH37	A4	I	Вход АЦП3, канал 37
ADC_CH38	E7	I	Вход АЦП3, канал 38
ADC_CH39	C5	I	Вход АЦП3, канал 39
ADC_CH40	B4	I	Вход АЦП3, канал 40
ADC_CH41	D6	I	Вход АЦП3, канал 41
ADC_CH42	F7	I	Вход АЦП3, канал 42
ADC_CH43	A3	I	Вход АЦП3, канал 43
ADC_CH44	E6	I	Вход АЦП3, канал 44
ADC_CH45	B3	I	Вход АЦП3, канал 45
ADC_CH46	D5	I	Вход АЦП3, канал 46
ADC_CH47	A2	I	Вход АЦП3, канал 47



Окончание таблицы 2

Вывод корпуса			Функциональное назначение вывода
Обозначение	Номер	Тип	
BOOT0	P3	I	Вход 0 выбора режима загрузки
BOOT1	P4	I	Вход 1 выбора режима загрузки
CPE	M2	I	Вход выбора основного источника тактирования
AREF	D4, D11	–	Вывод опорного напряжения АЦП
RESET#	N4	I	Вход сигнала сброса
XI_RTC	R3	–	Вывод для подключения кварцевого резонатора (32 кГц)
XO_RTC	Q4	–	Вывод для подключения кварцевого резонатора (32 кГц)
SERVEN	N3	I	Вход сигнала активации сервисного режима
TCK	L4	I	Вход тактового сигнала порта JTAG
TDI	N1	I	Вход данных порта JTAG
TDO	Q1	O	Выход данных порта JTAG
TMS	M4	I	Вход режима порта JTAG
TRST	M5	I	Вход сброса порта JTAG
XI_OSC	U1	–	Вывод внешнего тактового сигнала/ Вывод для подключения кварцевого резонатора
XO_OSC	U2	–	Вывод для подключения кварцевого резонатора

Таблица 2.3 – Выводы питания и земли

Вывод корпуса		Функциональное назначение вывода
Обозначение	Номер	
VDD12	B19, C12, C15, C17, E3, E16, E18, F5, H10, H11, H12, H13, J8, J10, J12, J13, K8, K12, K13, K14, L7, L8, L9, L13, M8, M9, M11, M12, M13, N8, N9, N10, N11, N12, N13, N18, R5, R11, S3, S5, S6, S18, T16, U19	Питание ядра 1,2 В
VDD33	C13, C18, D13, D15, E4, E13, E15, E17, F6, F12, F13, F14, F15, G6, G15, H6, H7, H15, J6, J15, J18, K6, K15, L6, L15, M6, M15, N6, N15, P6, P15, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, R4, R17, T4, T17	Питание буферов ввода-вывода 3,3 В
AVDD33	C4, C11	Аналоговое питание 3,3 В
RTC_BAT	T1	Батарейное питание RTC
RTC_GND	T2	Земля RTC
AGND	B2, E11	Аналоговая земля
GND	A1, C3, C14, D16, E12, F16, F17, G3, G11, G12, G13, G14, G16, H14, J7, J11, J14, K7, K9, K10, K11, L10, L11, L12, L14, M7, M10, M14, N7, N14, P5, P7, P8, P9, P10, P11, P12, P13, P14, Q5, Q16, Q18, R6, R7, R14, R15, S14, S15, V1	Земля

Микросхемы выполнены в керамическом корпусе 8115.400-1.

Масса микросхемы – не более 5,0 г.

## 2.2 Электрические параметры

Номинальное значение напряжения питания по выводам:

- VDD12 должно быть  $1,2 \text{ В} \pm 10 \%$  с допустимой разностью значений между выводами не более 50 мВ;

- VDD33, AVDD33 должно быть  $3,3 \text{ В} \pm 10 \%$  с допустимой разностью значений между выводами не более 50 мВ.

Амплитудное значение пульсации напряжения питания должно быть не более 50 мВ.

Значение суммарного максимального тока по выводам портов ввода-вывода А, В, С, D, E, F, G, H, J, K, L, M **не должно превышать 50 мА.**

Электрические параметры микросхем при приемке и поставке соответствуют нормам, приведенным в таблице 2.4.

Таблица 2.4 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпера- тура среды, °С
		не менее	не более	
1	2	3	4	5
1 Выходное напряжение низкого уровня по выводам А0 – А15, В0 – В15, С0 – С15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15, TDO, В, $U_{CC1} = 3,0 \text{ В}, U_{CC2} = 1,2 \text{ В}, I_{OL} = 6,0 \text{ мА}$	$U_{OL}$	–	0,4	–60 ± 3 25 ± 10 85 ± 3
2 Выходное напряжение высокого уровня по выводам А0 – А15, В0 – В15, С0 – С15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15, TDO, В, $U_{CC1} = 3,0 \text{ В}, U_{CC2} = 1,2 \text{ В}, I_{OH} = -6,0 \text{ мА}$	$U_{OH}$	$U_{CC1} - 0,9$	–	
3 Ток утечки низкого уровня по входам А0 – А15, В0 – В15, С0 – С15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15 с отключенными «pull-up» и «pull-down», мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IL} = 0 \text{ В}$	$I_{LL}$	–10	–	
4 Ток утечки высокого уровня по входам А0 – А15, В0 – В15, С0 – С15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15 с отключенными «pull-up» и «pull-down», мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IH} = U_{CC1}$	$I_{LH}$	–	10	
5 Входной ток низкого уровня по выводам «pull-up» <sup>1), 2)</sup> , мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IL} = 0 \text{ В}$	$I_{IL1}$	–200	–	
6 Входной ток высокого уровня по выводам «pull-up» <sup>2)</sup> , мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IH} = U_{CC1}$	$I_{IH1}$	–	10	
7 Входной ток низкого уровня по выводам «pull-down» СРЕ, ТСК, BOOT0, BOOT1, SERVEN, мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IL} = 0 \text{ В}$	$I_{IL2}$	–10	–	

Окончание таблицы 2.4

1	2	3	4	5
8 Входной ток высокого уровня по выводам «pull-down» CPE, TCK, BOOT0, BOOT1, SERVEN, мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IH} = U_{CC1}$	$I_{IH2}$	–	200	–60 ± 3 25 ± 10 85 ± 3
9 Входной ток низкого уровня по выводу XI_OSC мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IL} = 0 \text{ В}$	$I_{IL3}$	–40	–	
10 Входной ток высокого уровня по выводу XI_OSC, мкА, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{IH} = U_{CC1}$	$I_{IH3}$	–	40	
11 Динамический ток потребления ядра в активном режиме, А, $U_{CC1} = 3,6 \text{ В}, U_{CC2} = 1,32 \text{ В}, U_{CC3} = 3,6 \text{ В}, U_{CC4} = 3,6 \text{ В}, f_{SYSCLK} = 200 \text{ МГц}$	$I_{OCC2}$	–	1	
12 Интегральная нелинейность АЦП, ЕМР, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = 1,2 \text{ В}, U_{CC3} = 3,3 \text{ В}, U_{CC4} = 3,3 \text{ В}$	$E_L$	–3	3	
13 Дифференциальная нелинейность АЦП, ЕМР, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = 1,2 \text{ В}, U_{CC3} = 3,3 \text{ В}, U_{CC4} = 3,3 \text{ В}$	$E_{LD}$	–2	2	
14 Функциональный контроль $U_{CC1} = (3,0; 3,6) \text{ В}, U_{CC2} = (1,08; 1,32) \text{ В}, U_{CC3} = (3,0; 3,6) \text{ В}, U_{CC4} = (1,6; 3,6) \text{ В}; f_{SYSCLK} = (0,001; 200) \text{ МГц}$	ФК	–	–	
Примечание – Параметры $I_{LL}, I_{LN}, I_{H1}, I_{L2}$ при температуре минус 60 °С не измеряются, а гарантируются нормами при температуре (25 ± 10) °С.				
<sup>1)</sup> Выводы с подключаемой схемой «pull-up»: A0 – A15, B0 – B15, C0 – C15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15. <sup>2)</sup> Выводы с постоянной схемой «pull-up»: RESET#, TRST, TDI, TMS.				

Необходимо соблюдать следующий порядок подачи напряжения питания: сначала подаются напряжения питания  $U_{CC1}, U_{CC3}, U_{CC4}$ , затем напряжение  $U_{CC2}$ , а затем входные напряжения низкого уровня  $U_{IL}$  и высокого уровня  $U_{IH}$ .

Порядок снятия напряжений с выводов микросхемы при выключении должен быть обратным подаче: первыми снимаются входные сигналы, а затем напряжения питания.

Значения предельно допустимых электрических режимов эксплуатации в диапазоне рабочих температур среды соответствуют нормам, приведенным в таблице 2.5.

Таблица 2.5 – Значения предельно допустимых электрических режимов эксплуатации

Наименование параметра режима, единица измерения	Буквенное обозначение параметра	Предельно допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
1	2	3	4	5	6
1 Напряжение питания буферов ввода-вывода, В	$U_{CC1}$	3,0	3,6	–	5,0
2 Напряжение питания ядра, В	$U_{CC2}$	1,08	1,32	–	1,68
3 Напряжение питания АЦП, В	$U_{CC3}$	3,0	3,6	–	5,0
4 Напряжение питания по выводу RTC_BAT, В	$U_{CC4}$	1,6	3,6	–	4,0
5 Входное напряжение низкого уровня, В	$U_{IL}$	–0,5	0,8	–0,6	–
6 Входное напряжение высокого уровня, В	$U_{IH}$	$0,2U_{CC1} + 1$	$U_{CC1}$	–	$U_{CC1} + 0,6$
7 Входное напряжение высокого уровня по выводу XI_OSC, В	$U_{IHCI}$	$0,7U_{CC1}$	$U_{CC1}$	–	$U_{CC1} + 0,6$
8 Выходной ток низкого уровня, мА	$I_{OL}$	–	6,0	–	10
9 Выходной ток высокого уровня, мА	$I_{OH}$	–6,0	–	–10	–
10 Частота следования импульсов тактового сигнала по выводу XI_OSC, МГц	$f_C$	см. <sup>1)</sup>	24	–	–
11 Частота системного тактового сигнала SYSCLK, МГц	$f_{SYSCLK}$	0,001	200	–	–
12 Емкость нагрузки, пФ	$C_L$	–	40	–	–
<p><b>Примечания</b></p> <p>1 Время работы в одном из предельных режимов должно быть не более 5 с (кроме параметра 12).</p> <p>2 Значение суммарного максимально допустимого тока по выводам A0 – A15, B0 – B15, C0 – C15, D0 – D15, E0 – E15, F0 – F15, G0 – G15, H0 – H15, J0 – J15, K0 – K15, L0 – L15, M0 – M15 – не более 50 мА на каждую группу.</p> <p><sup>1)</sup> При работе с внешним тактовым генератором <math>f_C</math> – не менее 0,001 МГц. При работе с кварцевым резонатором <math>f_C</math> – не менее 8 МГц.</p>					

### 3 Архитектура изделия

Микроконтроллер 1921BK028 структурно представляет собой мультистадийный RISC процессор. Ядро Cortex-M4F основано на архитектуре ARMv7-M и полностью реализует наборы команд Thumb и Thumb2. Поддержка DSP-инструкций и наличие модуля операций с плавающей запятой существенно ускоряет обработку потоковых данных, что в свою очередь делает микроконтроллер весьма привлекательным для использования в системах управления и обработки информации.

Микроконтроллер способен параллельно выполнять четыре операции сложения/вычитания с 8-разрядными операндами или две операции сложения/вычитания с 16-разрядными операндами. Также реализовано умножение за один цикл, при этом для 16-разрядных чисел возможно параллельное исполнение двух операций. Из особенностей следует отметить аппаратное умножение 32-разрядных чисел за 1 цикл, а также деление 32-разрядных чисел, занимающее от 2 до 12 циклов.

#### 3.1 Блок коммутации микроконтроллера

Все устройства микроконтроллера соединены между собой через блок коммутации. На рисунке 3.1 приведена схема соединения основных и периферийных блоков микроконтроллера внутри блока коммутации.

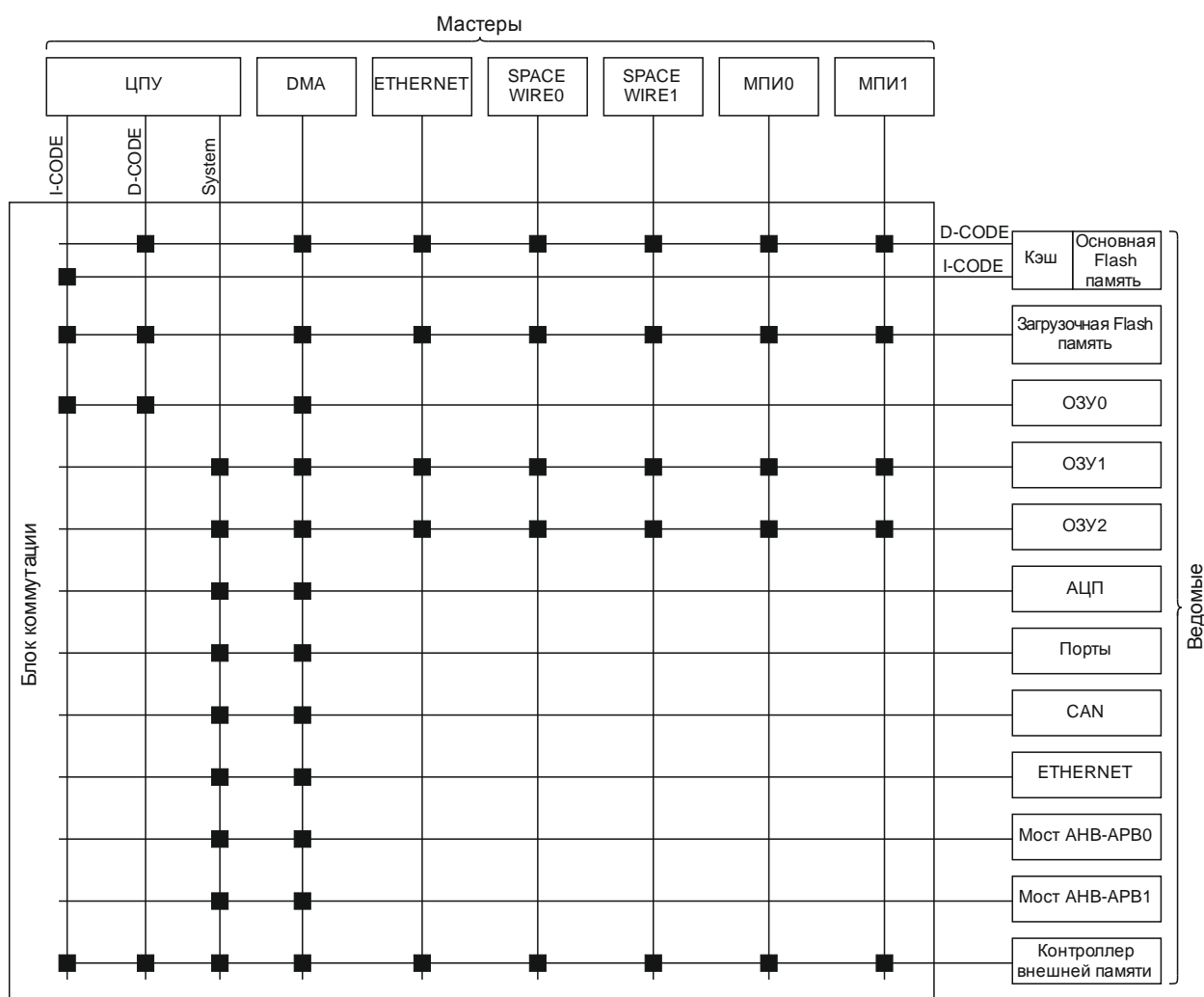


Рисунок 3.1 – Схема соединения блоков внутри блока коммутации микроконтроллера

## 4 Блок управления сбросом и синхронизацией RCU

### 4.1 Общая система тактирования

В микроконтроллере предусмотрена развитая подсистема управления тактовыми сигналами и сигналами сброса, см. рисунок 4.1.

Основными тактовыми сигналами, которые используются в микросхеме, являются:

- OSICLK – тактовый сигнал внутреннего RC-генератора;
- OSECLK – внешний тактовый сигнал с выводов XI\_OSC и XO\_OSC;
- RTCCLK – внешний низкочастотный тактовый сигнал частотой 36,768 кГц с выводов XI\_RTC и XO\_RTC;
- REFCLK – опорный тактовый сигнал (зависит от состояния вывода CPE);
- SRCCLK – исходный тактовый сигнал (зависит от состояния вывода CPE);
- PLLCLK – тактовый сигнал с выхода PLL;
- PLLDIVCLK – тактовый сигнал с выхода PLL, прошедший через дополнительный делитель;
- PLLEXTCLK – внешний тактовый сигнал с вывода PLL\_EXTSRC;
- TRACEEXTCLK – внешний тактовый сигнал с вывода TRACE\_EXTCLK;
- SYSCLK – системный тактовый сигнал, выбранный блоком SCM, см. на рисунке 4.1;
- PCLK – периферийный тактовый сигнал APB шины, получаемый путем деления HCLK (регистр APBCFG);
- HCLK – системный тактовый сигнал АНВ шины;
- FCLK – неотключаемый системный тактовый сигнал.

Важный момент в подсистеме тактирования – выбор частоты тактового сигнала APB шины PCLK. Она не должна превышать 100 МГц. В случае, когда частота тактового сигнала HCLK превышает это значение, необходимо включить делитель для PCLK, записав регистр APBCFG. Значение делителя задается как  $2^{DIV}$ . Соответственно можно задать коэффициент деления на 1, 2, 4, 8.

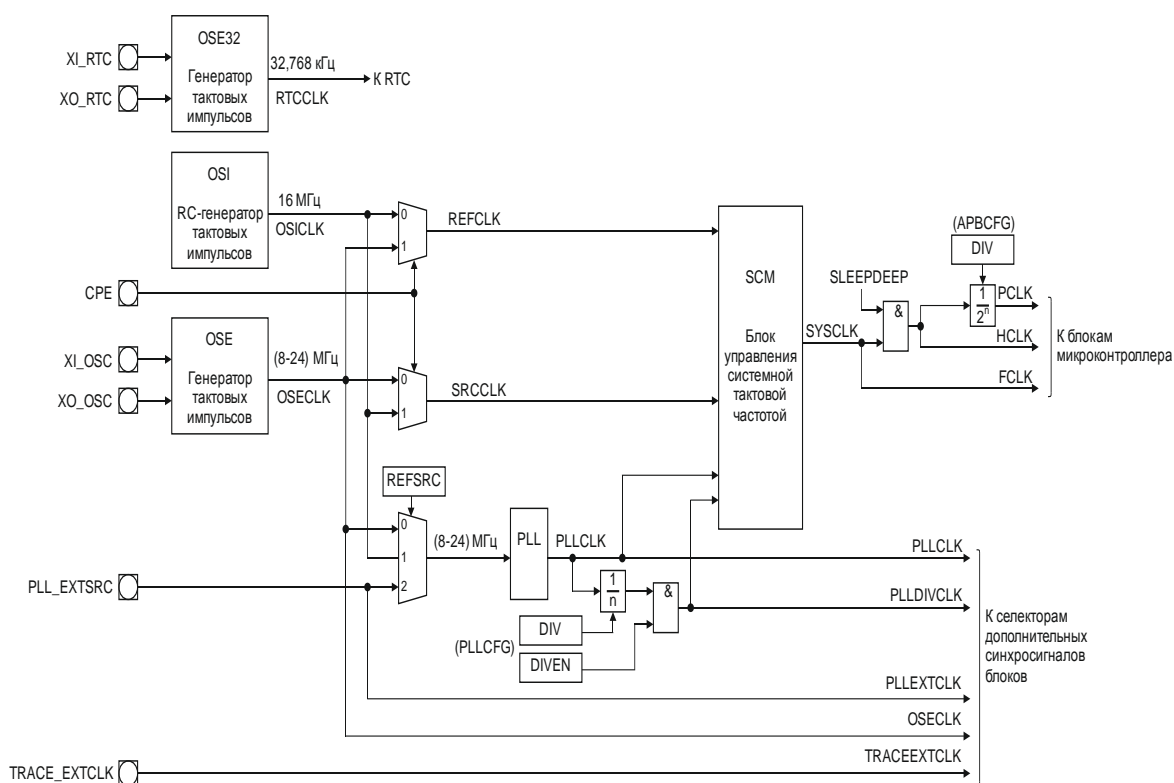


Рисунок 4.1 – Схема синхронизации

Выбор источников для REFCLK и SRCCLK осуществляется внешним выводом CPE, см. рисунок 4.1 и таблицу 4.1.

Таблица 4.1 – Влияние состояния вывода CPE на источники тактовых сигналов

CPE	REFCLK	SRCCLK
0	OSICLK	OSECLK
1	OSECLK	OSICLK

## 4.2 Синтезатор частоты на основе ФАПЧ (PLL)

Общая функциональная схема блока PLL показана на рисунке 4.2.

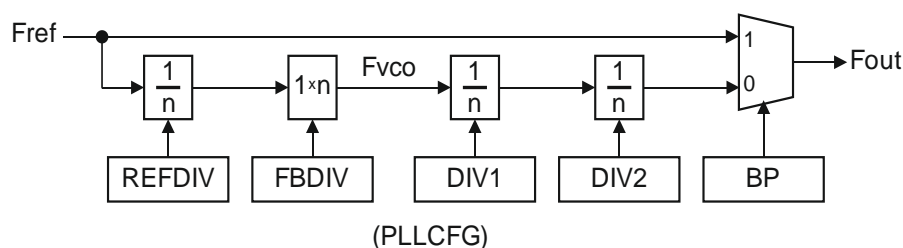


Рисунок 4.2 – Функциональная схема блока PLL

В качестве опорной частоты  $f_{REF}$  блока PLL используется источник, определенный полем REFSRC регистра PLLCFG.

Выходная частота блока PLL вычисляется согласно формуле

$$f_{OUT} = \frac{f_{REF}}{REFDIV} \times \frac{FB DIV}{DIV1 \times DIV2} \quad (4.1)$$

Коэффициенты REFDIV, FB DIV, DIV1, DIV2 задаются соответствующими полями регистра PLLCFG. Существуют следующие ограничения параметров:

- коэффициент деления входной частоты  $1 \leq REFDIV \leq 63$ ;
- коэффициент деления обратной связи  $16 \leq FB DIV \leq 200$ ;
- коэффициент деления  $1 \leq DIV1 \leq 7$ ;
- коэффициент деления  $1 \leq DIV2 \leq 7$ ;
- входная частота  $f_{REF}$  должна находиться в диапазоне (8 – 800) МГц;
- значение частоты  $f_{VCO}$  должно быть в диапазоне (400 – 1600) МГц;
- значение выходной частоты  $f_{OUT}$  должно быть в диапазоне (8 – 1600) МГц.

Настройку блока PLL необходимо делать до того, как его тактовый сигнал будет выбран в качестве рабочей частоты системы или одного из блоков. Сначала необходимо отключить прохождение входного сигнала сквозь блок PLL путём сброса BP бита в регистре PLLCFG. Затем установить делители REFDIV и FB DIV, соблюдая указанные выше ограничения. Следующим шагом будет вывод PLL из энергосберегающего режима (сброс бита PD регистра PLLCFG). Теперь следует установить желаемые коэффициенты DIV1 и DIV2, и разрешить генерацию выходной частоты путем сброса бита OUTDIS.

При правильной установке всех значений и выходе блока PLL на рабочий режим будет установлен бит LOCK в регистре PLLCFG.

Дополнительно можно использовать внешний делитель частоты, управляемый битами регистра PLLDIV. Для тактирования различных блоков и модулей микроконтроллера можно выбрать как сигнал PLLCLK, так и PLLDIVCLK, независимо друг от друга.

### 4.3 Система слежения за тактовыми сигналами

Система слежения осуществляет контроль источников тактовых сигналов и позволяет обрабатывать исключительные ситуации, связанные с их пропаданием (срыв генерации блока PLL, ненадежный контакт с внешним резонатором и т. п.).

Текущий статус тактовых сигналов можно установить, прочитав соответствующие биты xOK и xERR регистра SYCLKSTAT, где x – SRCCLK, PLLCLK, PLLDIVCLK. Бит xOK будет установлен, если соответствующий тактовый сигнал стабильно работает, и будет сброшен при его сбое. Бит xERR имеет обратное поведение – при сбое устанавливается, сбрасывается при нормальной работе тактового сигнала.

Время реакции системы слежения на исчезновение тактового сигнала настраивается с помощью полей регистра SECPRD, и по умолчанию равно 256 тактам сигнала REFCLK. Реакция на любое пропадание тактового сигнала на меньшее время будет отсутствовать.

Минимальная частота, за которой может осуществляться слежение – 40 кГц.

Значение минимально возможного времени реакции вычисляется по формуле (результат округляется до целого в большую сторону)

$$\text{SECPRD\_MIN} = (4 \times T_{\text{REF}} + 6 \times T_{\text{CLK}}) / T_{\text{REF}}, \quad (4.2)$$

где  $T_{\text{REF}}$  – период опорной тактовой частоты REFCLK;

$T_{\text{CLK}}$  – период контролируемой тактовой частоты.

Например, если необходима максимально быстрая реакция на пропадание частоты 100 МГц на выходе блока PLL ( $T_{\text{CLK}} = 10$  нс) при частоте сигнала REFCLK 16 МГц ( $T_{\text{REF}} = 62,5$  нс), то необходимо в поле PLLCLK регистра SECPRD записать значение SECPRD\_MIN = 5, вычисленное по формуле (4.2).

Стоит отметить, что SECPRD\_MIN является лишь временем детектирования сбоя. Сам переход в аварийный режим (переключение системной частоты на сигнал REFCLK) займет дополнительно не более 14 тактов сигнала REFCLK.

Кроме слежения за каждым из источников тактового сигнала по отдельности, система отслеживания позволяет контролировать текущий системный тактовый сигнал SYCLK.

Включение контроля сигнала SYCLK осуществляется установкой бита SECEN регистра SYCLKCFG.

При сбое будет осуществлен аварийный переход системной частоты на сигнал OSICLK и будет выработано немаскируемое прерывание NMI# совместно с установкой флага прерывания SYSFAIL в регистре INTSTAT (флаг сбрасывается записью единицы).

Текущий статус системного тактового сигнала доступен в бите SYSFAIL регистра SYCLKSTAT: при сбое этот бит установится и будет держаться установленным, пока пользователь вручную не перейдет на любой из доступных стабильных источников, записав в поле SYSSEL регистра SYCLKCFG новое значение (при сбое поле сохраняет старое значение источника системной частоты). В том числе возможен переход и на сигнал REFCLK, несмотря на то, что в аварийном режиме схема уже тактируется этим сигналом – это позволит сбросить бит SYSFAIL регистра SYCLKSTAT.

Прямой переход из аварийного состояния к тактированию от вызвавшего сбой, но уже восстановившегося источника – невозможен. Если это необходимо сделать, то сначала нужно перейти на один из стабильных источников (например, сигнал REFCLK), и лишь затем начать переход на бывший «сбойным», но восстановившийся источник.

При переходе микроконтроллера в аварийное состояние по тактированию требуется обязательно перейти на стабильный источник тактирования (бит xGOOD для него будет установлен), например, на сигнал REFCLK ( $\text{SYSSEL} = 0$ ). Если аварийный источник возобновил тактирование, то снова перейти на него можно, записав поле SYSSEL регистра SYCLKCFG на номер соответствующего источника.



#### 4.4 Сигналы сброса

Сброс может осуществляться как внешним сигналом с вывода RESET#, так и встроенным монитором питания POR (PowerOn Reset). Активный уровень сигнала сброса – низкий.

Помимо внешнего сброса существуют два внутренних источника сброса – по запросу процессора и по переполнению сторожевого таймера.

На рисунке 4.3 показана диаграмма срабатывания монитора питания POR при изменении питания. В таблице 4.2 указаны динамические и электрические параметры монитора питания POR во время сброса.

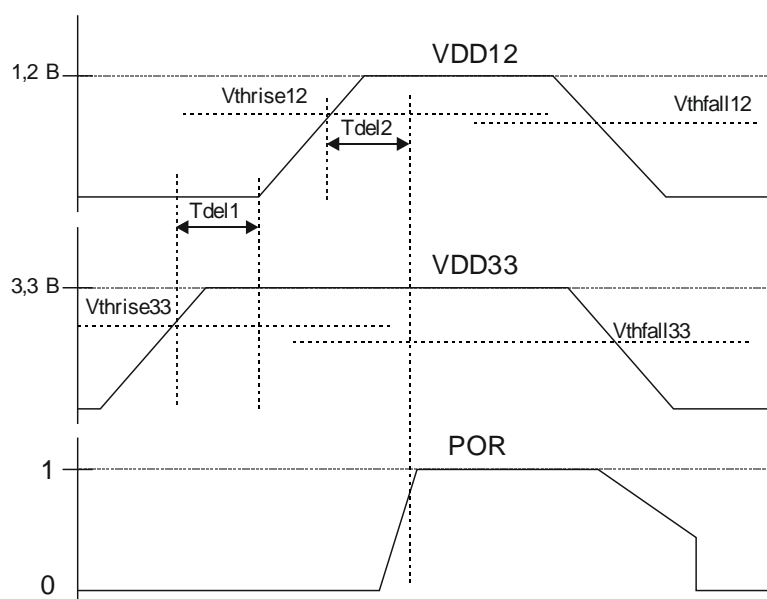


Рисунок 4.3 – Диаграмма сигнала сброса монитора питания POR при изменении линий питания

Как видно из рисунка выше, минимально возможное время задержки срабатывания POR определяется величиной Tdel1.

Таблица 4.2 – Динамические и электрические параметры монитора питания POR

Наименование параметра, единица измерения	Обозначение параметра	Миним.	Номин.	Макс.
Напряжение на выводах VDD33 для отключения схемы сброса, В	Vthrise33	2,70	2,80	2,90
Напряжение на выводах VDD33 для срабатывания схемы сброса, В	Vthfall33	2,53	2,62	2,72
Напряжение на выводах VDD12 для отключения схемы сброса, В	Vthrise12	0,986	1,0	1,02
Напряжение на выводах VDD12 для срабатывания схемы сброса, В	Vthfall12	0,890	0,9	0,92
Время задержки срабатывания монитора 3,3 В, мс	Tdel1	–	126	–
Время задержки срабатывания монитора 1,2 В, мкс	Tdel2	–	0,25	–

Также в микроконтроллере есть возможность сброса по получении сигнала LOCKUP от ядра. Данный сигнал возникает в случае невозможности восстановления системного исключения (Unrecoverable exception), если микроконтроллер не может из него выйти. Установка бита LOCKUPEN в регистре SYSRSTCFG разрешит автоматический сброс после получения от ядра сигнала LOCKUP.

В регистре SYSRSTSTAT содержится набор флагов, по которым можно установить причину последнего произошедшего сброса. В случае нормальной загрузки после подачи питания будет установлен флаг POR, так как сброс вызван срабатыванием схемы POR (PowerON Reset).

#### 4.5 Прерывания

В микроконтроллере есть два прерывания, связанных с работой блока RCU.

Первое прерывание – непосредственно RCU Interrupt. Выбрать источники, влияющие на него можно в регистре INTEN – события появления или пропадания одного из тактовых сигналов, выход блока PLL в рабочий режим. Все события сопровождаются флагами в регистре INTSTAT.

Второе прерывание связано с пропаданием текущего системного тактового сигнала SYSCLK. В случае если была включена система слежения за тактовым сигналом (бит SECEN регистра SYSCCLKCFG), возникнет прерывание, соответствующее вектору немаскируемого прерывания NMI#, и будет установлен соответствующий флаг в регистре INTSTAT.

#### 4.6 Тактирование и сброс периферийных блоков

Для некоторых периферийных блоков система тактирования предусматривает выбор независимого тактового источника. У блоков WDT, TRACE, UART, SPI, ADC, MILSTD, SPWR и блока формирования внешнего сигнала CLKOUT есть соответствующие регистры, в которых можно включать тактирование, выбирать его источник и настраивать отключаемый делитель. Источник тактирования выбирается независимо для каждого из блоков с помощью его собственного селектора, показанного на рисунке 4.4.

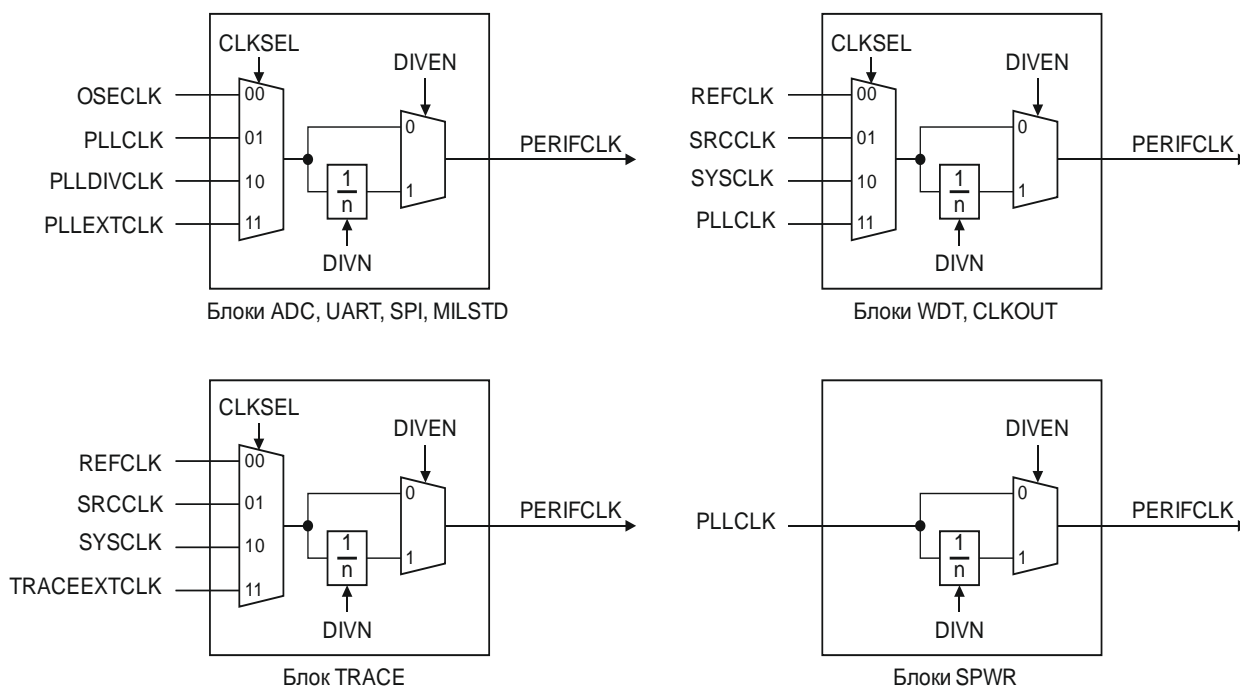


Рисунок 4.4 – Схема селектора дополнительного синхросигнала блока

Для остальных блоков, а именно CAP, TMR, ETMR, OWI, I2C, QEP, PWM, TUART, LAU и для блоков, относящихся к АНВ периферии (GPIO, CAN, SDFM, ETHERNET), тактирование подается установкой соответствующего бита в CLKCFG, см. таблицу 4.3.

Таблица 4.3 – Тактовые сигналы периферийных блоков

Периферийный блок	Тактовый сигнал	Делитель	Управляющий регистр
ADC	ACLK	1-63	ADCCFG
SPI	SPICLK	1-63	SPICFG
UART	UARTCLK	1-63	UARTCFG
WDT	WDTCLK	1-63	WDTCFG
SPWR	SPWRCLK	1-63	SPWRCFG
MILSTD	MILSTDCLK	1-63	MILSTDCFG
CLOCKOUT	CLKOUT	1-7	CLKOUTCFG
TRACE	TRACECLK	1-63	TRACECFG
TMR	PCLK	-	PCLKCFG, PRSTCFG
ETMR	PCLK	-	PCLKCFG, PRSTCFG
RTC	PCLK	-	PCLKCFG, PRSTCFG
LAU	PCLK	-	PCLKCFG, PRSTCFG
OWI	PCLK	-	PCLKCFG, PRSTCFG
PWM	PCLK	-	PCLKCFG, PRSTCFG
I2C	PCLK	-	PCLKCFG, PRSTCFG
QEP	PCLK	-	PCLKCFG, PRSTCFG
CAP	PCLK	-	PCLKCFG, PRSTCFG
GPIO	HCLK	-	HCLKCFG, HRSTCFG
CAN	HCLK	-	HCLKCFG, HRSTCFG
MILSTD	HCLK	-	HCLKCFG, HRSTCFG
SPWR	HCLK	-	HCLKCFG, HRSTCFG
SDFM	HCLK	-	HCLKCFG, HRSTCFG
EXTMEM	HCLK	-	HCLKCFG, HRSTCFG
ETHERNET	HCLK	-	HCLKCFG, HRSTCFG

У блоков MILSTD и SPWR тактовые сигналы для приемо-передающей части и для регистров управления разделены. Управление тактированием и коэффициентом деления передатчика SPWR и кодека MILSTD осуществляется через регистры SPWRCFG и MILSTDCGF (для последнего также доступен бит, отвечающий за сброс кодека). Включение тактирования и сброса регистровой части этих блоков осуществляются в регистрах HCLKCFG и HRSTCFG.

По умолчанию на всех периферийных блоках отключено тактирование, и все они находятся в состоянии сброса. Для начала работы нужно подать тактовый сигнал, а также вывести блок из состояния сброса, осуществив запись единиц в соответствующие биты регистров HRSTCFG и PRSTCFG.

## 5 Блок управления энергопотреблением PMU

В микроконтроллере предусмотрены несколько режимов функционирования для уменьшения потребляемой энергии в тех задачах, где ядро не должно постоянно работать, например, во время ожидания прихода внешнего события. К ним относятся:

- режим Sleep (ядро остановлено, периферия работает);
- режим Deepsleep (ядро остановлено, периферия переведена в режим Powerdown).

Контроль режимов энергопотребления осуществляется с помощью блока управления энергопотреблением PMU.

### 5.1 Режим Sleep

Вход в режим Sleep производится при выполнении инструкции WFI (Wait For Interrupt) или WFE (Wait For Event) при сброшенном бите SLEEPDEEP регистра SCR блока SCB ядра. В этом режиме прекращается тактирование ядра, но вся периферия продолжает полностью функционировать.

В зависимости от состояния бита SLEEPONEXIT (регистр SCR блока SCB ядра) возможны два режима входа в этот режим:

- SLEEPONEXIT = 0 – вход в режим Sleep производится, как только была выполнена инструкция WFI или WFE, при выходе продолжает исполняться программа, прерванная одной из инструкций выше;

- SLEEPONEXIT = 1 – вход в режим Sleep осуществляется не только при выполнении соответствующих инструкций, но и каждый раз при выходе из последнего активного обработчика прерываний (ядро просыпается только для обработки прерываний).

Выход из режима Sleep производится различными путями, в зависимости от того, какая инструкция была использована для входа.

Если для входа в режим Sleep была использована инструкция WFI, то приход любого разрешенного прерывания в NVIC вызовет выход из режима, одновременно с переходом в соответствующий обработчик.

Если была использована инструкция WFE, то выход произойдет либо по разрешенному прерыванию NVIC (аналогично WFI), либо по событию, которое может быть сгенерировано несколькими путями.

Если установлен бит SEVONPEND регистра SCR блока SCB ядра, то событием будет являться приход любого запрещенного прерывания в NVIC (разрешенного лишь в периферийном модуле, а не в NVIC). При этом по выходе необходимо будет сбрасывать соответствующие флаги прерывания, как в периферийном модуле, так и в регистрах NVIC (регистры ICPR).

Также событием для выхода из режима Sleep является сигнал RXEV, который может быть сгенерирован в любом из портов GPIO. Дополнительно сигнал RXEV от необходимого порта должен быть разрешен установкой бита GPIOxEV (где x = A, B, C, D, E, F, G, H, J, K, L, M) в регистре RXEVEN блока PMU. В этом случае выход из Sleep будет произведен наиболее быстро, так как время на вход/выход в прерывание тратиться не будет.

### 5.2 Режим Deepsleep

В режиме Deepsleep тактирование ядра останавливается, а также происходит переход в режим Powerdown следующих периферийных блоков (по умолчанию): MFLASH, BFLASH, PLL, EXTOSC.

Вход в режим Deepsleep производится при выполнении инструкции WFI (Wait For Interrupt) или WFE (Wait For Event) при установленном бите SLEEPDEEP регистра SCR блока SCB ядра.

Если для входа в режим DeepSleep была использована инструкция WFI, то приход любого разрешенного прерывания в NVIC вызовет выход из режима, одновременно с переходом в соответствующий обработчик.

Если для входа в режим DeepSleep была использована инструкция WFE, то приход любого разрешенного события RXEV от портов ввода-вывода вызовет выход из режима.

Перед тем как осуществить вход в этот режим необходимо провести некоторые подготовительные операции.

Необходимо включить PMU, установив бит EN в регистре CFG блока PMU. В противном случае переход в режим DeepSleep будет невозможен.

С помощью сброса битов PLLPD, MFLASHPD, BFLASHPD, EXTOSCPD в регистре PDEN блока PMU можно отключить перевод соответствующих блоков в режим Powerdown при входе в режим DeepSleep – они продолжат свое функционирование. Например, если оставить активным внешний осциллятор, периферия, использующая его в качестве тактового сигнала, продолжит функционировать.

Необходимо настроить значение длительности выхода из режима DeepSleep – выход периферийных блоков из режима Powerdown требует некоторого времени (для каждого блока разное), поэтому в регистр PUDEL блока PMU должно быть внесено значение большее либо равное самой большой величине задержки среди блоков, для которых активен режим Powerdown. Значение вносится в тактах тактового сигнала REFCLK. В таблице 5.1 представлены типовые величины задержек. По умолчанию PUDEL содержит значение 704, т. к. возможность активации режима Powerdown включена для блока PLL.

Таблица 5.1 – Настройка времени выхода из режима Powerdown для указанных блоков

Периферийный блок	Время выхода из режима Powerdown, нс	Значение для регистра PUDEL в тактах сигнала REFCLK ( $f_{REFCLK} = 16 \text{ МГц}$ )
PLL	44000	704
MFLASH	100	2
BFLASH	100	2
EXTOSC	0	0

## 6 Организация памяти

Память микроконтроллера имеет predetermined 32-разрядное адресное пространство с областями: программ, данных, периферии и внутренних ресурсов, жестко соединенных с процессором. Адресное пространство разбито на несколько областей, см. таблицу 6.1.

Таблица 6.1 – Общая организация памяти микроконтроллера

Адресное пространство	Описание
E000_0000h – FFFF_FFFFh	Системная область
8000_0000h – 87FF_FFFFh	Внешняя память (восемь последовательных окон по 16 байт)
4000_0000h – 400F_FFFFh	Регистры управления периферийными блоками
2000_8000h – 2009_FFFFh	Внутреннее ОЗУ2 128 Кбайт
2000_0000h – 2007_FFFFh	Внутреннее ОЗУ1 512 Кбайт
1300_0000h – 1300_FFFFh	Внутреннее ОЗУ0 64 Кбайт
1100_0000h – 1107_FFFFh	Загрузочная Flash-память (BFLASH) 512 Кбайт
1000_0000h – 101F_FFFFh	Основная Flash-память (MFLASH) 2 Мбайт
0000_0000h – 0FFF_FFFFh	Область переопределения памяти

Область переопределения памяти позволяет осуществлять загрузку и получать доступ к тому виду памяти, который определен состоянием выводов BOOT0, BOOT1, см. рисунок 6.1.

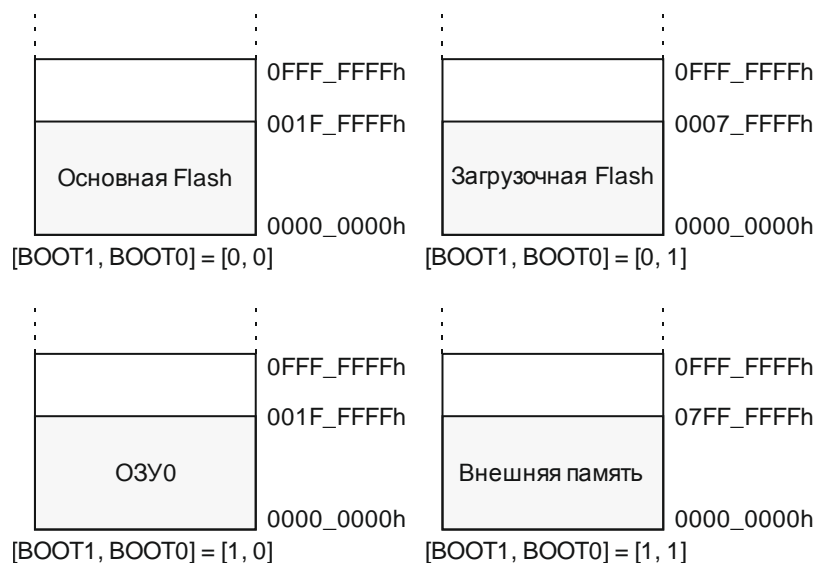


Рисунок 6.1 – Выбор режима загрузки микроконтроллера

Системная область делится на две части. Первая часть, объемом 1 Мбайт, занимает диапазон адресов E000\_0000h – E007\_FFFFh и зарезервирована для «личной» периферийной шины PPB (Private Peripheral Bus). Шина PPB используется для выборки/записи данных и отладочного доступа – для периферии. Эта область рассматривается как строго упорядоченная память. Часть ее адресов (E000\_E000h – E000\_EFFFh) занимает пространство управления системы SCS, в котором находятся регистры блока управления системой SCB, контроллера прерываний NVIC, системного таймера SysTick, средств отладки и другие. Байты регистров SCS всегда располагаются в порядке «младший – старший» независимо от того, какой формат представления данных принят для остальной памяти. Доступ к большинству регистров возможен только полными двойными словами (32 бита), все исключения из этого правила оговариваются специально. Попытка кода обратиться к области PPB обычно вызывает прерывание по

ошибке шины; исключениями являются возможность доступа к регистру STIR, если это разрешено соответствующим битом регистра CCR и доступность отладочных регистров. Вторая часть является памятью устройств, занимает область E010\_0000h – FFFF\_FFFFh и является системным регионом ядра.

Регистры периферийных блоков микроконтроллера доступны в адресном пространстве 4000\_0000h – 400F\_FFFFh. Карта размещения блоков микроконтроллера в памяти представлена в таблице 6.2.

Таблица 6.2 – Карта регистров периферийных блоков

Адресное пространство	Обозначение блока	Описание
400B_1000h – 400B_1FFFh	ADC	Блок АЦП
400B_0000h – 400B_0FFFh	OWI1	Контроллер однопроводного интерфейса 1
400A_F000h – 400A_FFFFh	OWI0	Контроллер однопроводного интерфейса 0
400A_E000h – 400A_EFFFh	LAU	Блок конфигурируемых логических ячеек
400A_D000h – 400A_DFFFh	QEP3	Квадратурный декодер 3
400A_C000h – 400A_CFFFh	QEP2	Квадратурный декодер 2
400A_B000h – 400A_BFFFh	QEP1	Квадратурный декодер 1
400A_A000h – 400A_AFFFh	QEP0	Квадратурный декодер 0
400A_9000h – 400A_9FFFh	PWM9	Блок ШИМ9
400A_8000h – 400A_8FFFh	PWM8	Блок ШИМ8
400A_7000h – 400A_7FFFh	PWM7	Блок ШИМ7
400A_6000h – 400A_6FFFh	PWM6	Блок ШИМ6
400A_5000h – 400A_6FFFh	PWM5	Блок ШИМ5
400A_4000h – 400A_5FFFh	PWM4	Блок ШИМ4
400A_3000h – 400A_4FFFh	PWM3	Блок ШИМ3
400A_2000h – 400A_2FFFh	PWM2	Блок ШИМ2
400A_1000h – 400A_1FFFh	PWM1	Блок ШИМ1
400A_0000h – 400A_0FFFh	PWM0	Блок ШИМ0
4009_F000h – 4009_FFFFh	ECAP5	Блок захвата 5
4009_E000h – 4009_EFFFh	ECAP4	Блок захвата 4
4009_D000h – 4009_DFFFh	ECAP3	Блок захвата 3
4009_C000h – 4009_CFFFh	ECAP2	Блок захвата 2
4009_B000h – 4009_BFFFh	ECAP1	Блок захвата 1
4009_A000h – 4009_AFFFh	ECAP0	Блок захвата 0
4009_9000h – 4009_9FFFh	RTC	Блок часов реального времени
4009_8000h – 4009_8FFFh	ETMR3	Блок расширенного таймера 3
4009_7000h – 4009_7FFFh	ETMR2	Блок расширенного таймера 2
4009_6000h – 4009_6FFFh	ETMR1	Блок расширенного таймера 1
4009_5000h – 4009_5FFFh	ETMR0	Блок расширенного таймера 0
4009_4000h – 4009_4FFFh	TMR3	Блок таймера 3
4009_3000h – 4009_3FFFh	TMR2	Блок таймера 2
4009_2000h – 4009_2FFFh	TMR1	Блок таймера 1
4009_1000h – 4009_1FFFh	TMR0	Блок таймера 0
4009_0000h – 4009_0FFFh	SPI3	Последовательный интерфейс SPI3
4008_F000h – 4008_FFFFh	SPI2	Последовательный интерфейс SPI2
4008_E000h – 4008_EFFFh	SPI1	Последовательный интерфейс SPI1
4008_D000h – 4008_DFFFh	SPI0	Последовательный интерфейс SPI0
4008_C000h – 4008_CFFFh	I2C1	Последовательный интерфейс I2C1
4008_B000h – 4008_BFFFh	I2C0	Последовательный интерфейс I2C0

Окончание таблицы 6.2

Адресное пространство	Обозначение блока	Описание
4008_A000h – 4008_AFFFh	TUART1	Последовательный приемопередатчик TUART1
4008_9000h – 4008_9FFFh	TUART0	Последовательный приемопередатчик TUART0
4008_8000h – 4008_8FFFh	UART3	Последовательный приемопередатчик UART3
4008_7000h – 4008_7FFFh	UART2	Последовательный приемопередатчик UART2
4008_6000h – 4008_6FFFh	UART1	Последовательный приемопередатчик UART1
4008_5000h – 4008_5FFFh	UART0	Последовательный приемопередатчик UART0
4008_4000h – 4008_4FFFh	DMA	Контроллер прямого доступа к памяти
4008_3000h – 4008_3FFFh	WDT	Сторожевой таймер
4008_2000h – 4008_2FFFh	PMU	Блок управления питанием
4008_1000h – 4008_1FFFh	RCU	Блок управления сбросом и синхронизацией
4008_0000h – 4008_0FFFh	SIU	Блок системных функций
4007_5000h – 4007_5FFFh	SDFM	Блок сигма-дельта демодулятора
4007_4000h – 4007_4FFFh	SPWRTMR	Таймер для интерфейса SpaceWire
4007_3000h – 4007_3FFFh	MILSTD1	Контроллер МПИ1
4007_2000h – 4007_2FFFh	MILSTD0	Контроллер МПИ0
4007_1000h – 4007_1FFFh	SPWR1	Контроллер SpaceWire1
4007_0000h – 4007_0FFFh	SPWR0	Контроллер SpaceWire0
4006_2000h – 4006_2FFFh	EXTMEM	Внешняя память
4006_1000h – 4006_1FFFh	BFLASH	Загрузочная Flash-память
4006_0000h – 4006_0FFFh	MFLASH	Основная Flash-память
4004_0000h – 4004_FFFFh	TMU	Блок тригонометрических вычислений
4003_0000h – 4003_FFFFh	ETHERNET	Контроллер Ethernet
4002_0000h – 4002_FFFFh	CAN	Контроллер CAN
4001_B000h – 4001_BFFFh	GPIOM	Порт M
4001_A000h – 4001_AFFFh	GPIOL	Порт L
4001_9000h – 4001_9FFFh	GPIOK	Порт K
4001_8000h – 4001_8FFFh	GPIOJ	Порт J
4001_7000h – 4001_7FFFh	GPIOH	Порт H
4001_6000h – 4001_6FFFh	GPIOG	Порт G
4001_5000h – 4001_5FFFh	GPIOF	Порт F
4001_4000h – 4001_4FFFh	GPIOE	Порт E
4001_3000h – 4001_3FFFh	GPIOD	Порт D
4001_2000h – 4001_2FFFh	GPIOC	Порт C
4001_1000h – 4001_1FFFh	GPIOB	Порт B
4001_0000h – 4001_0FFFh	GPIOA	Порт A



## 7 Контроллер Flash-памяти

### 7.1 Основная Flash-память

Основная Flash-память может использоваться для хранения программ и данных пользователя. Размер основной Flash-памяти составляет 2 Мбайт (128 страниц по 16 Кбайт), и в адресном пространстве она занимает диапазон с 1000\_0000h по 101F\_FFFFh.

Чтение Flash-памяти осуществляется через две шины АНВ: I-code (для команд) и D-code (для данных). Чтение D-code шины имеет приоритет. На обеих шинах при попытке записи в любую область, чтении из несуществующей области, чтении во время, когда Flash занята (стирание, запись), транзакция проходит успешно с неопределенными данными на выходе.

Память доступна для чтения, записи, полного и постраничного стирания через регистры данных DATA<sub>n</sub> (n от 0 до 15), адреса ADDR, команд CMD, статуса STAT блока MFLASH. Запись необходимо производить в предварительно очищенную ячейку памяти. Стирание памяти осуществляется полностью или постранично.

Минимальное время чтения данных из Flash-памяти составляет около 30 нс. Поэтому, исходя из выбранной рабочей частоты, следует задать определенное количество дополнительных тактов ожидания, необходимое для стабильного чтения из Flash-памяти. Данный параметр заносится в поле LAT регистра CTRL

Таблица 7.1 – Значения параметра (количество дополнительных тактов ожидания) в поле LAT в зависимости от частоты сигнала SYSCLK

f <sub>SYSCLK</sub> , не более, МГц	Количество дополнительных тактов ожидания при чтении Flash-памяти
200	6
180	5
150	4
120	3
90	2
60	1
30	0
Примечание – Значение параметра после сброса равно 1.	

Помимо основной области памяти (2 Мбайт), доступной как через глобальное адресное пространство, так и через регистры контроллера, существует дополнительная (NVR) область, доступная только для операций через регистры. Размер этой области 64 Кбайт (4 страницы по 16 Кбайт). Для выполнения операций с этой областью необходимо, чтобы вместе с командой был установлен бит NVRON в регистре CMD. Данную память можно использовать для хранения пользовательских данных.

#### Операция предвыборки

При запросе данных на шине по адресу, по которому не осуществлялась предвыборка, выполняются следующие действия:

1 Сигнал готовности на шине устанавливается в ноль и задерживает транзакцию.

2 По запрашиваемому адресу считываются 16 32-битных слов (512 бит) данных из Flash-памяти. Далее эти данные записываются во внутренний первый буфер.

3 Требуемое слово передается на шину АНВ, и сигнал готовности устанавливается в единицу.

4 Сразу после установки сигнала готовности из Flash-памяти считываются 512 бит данных по следующему адресу. Данные сохраняются во втором буфере. Если во время считывания этих данных появляются запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по другим адресам, то готовность на шине

устанавливается в ноль и происходит ожидание завершения считывания во второй буфер и далее возврат к действию 2.

5 Если приходят запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по адресам, находящимся во втором буфере, ответ также возникает мгновенно. Далее переписывается первый буфер значением второго и считывается следующий адрес из Flash-памяти. Если приходят запросы по адресам не из первого и второго буферов, то возврат к действию 1.

### **Операция кэширования**

Основная Flash-память дополнена блоками кэш-памяти по 16 Кбайт на каждую из шин I-code и D-code. Доступ к этим блокам осуществляется за 1 такт системной частоты. При первом обращении к адресу его данные параллельно заносятся и в кэш-память, и сохраняются там. Если в дальнейшем произойдет повторное обращение к тому же адресу, данные будут прочитаны уже из кэша, обращения к медленной Flash-памяти не будет. Но данные в кэше, которые были запрошены давно, могут быть вытеснены более свежими данными, обращение к которым было позже. Таким образом, кэшированная информация постоянно обновляется, но участки программного кода, вызываемые чаще всего, с большей вероятностью окажутся в кэше. За счет этого достигается повышенная скорость выполнения такого кода.

Предвыборка и кэш на каждую из шин I-code и D-code включается отдельно в регистре CTRL. Перед включением кэша необходимо сбросить его, установив соответствующий бит FLUSH регистра CTRL. Во время процедуры сброса бит BUSY соответствующего регистра статуса кэша будет держаться в 1. После успешного завершения сброса будет сброшен и данный бит, после чего можно продолжать выполнение основной программы.

## **7.2 Загрузочная Flash-память**

Загрузочная Flash-память может использоваться для хранения программ и данных пользователя. Размер этой Flash-памяти составляет 512 Кбайт (128 страниц по 4 Кбайт), и в адресном пространстве она занимает диапазон с 1100\_0000h по 1107\_FFFFh.

Чтение Flash-памяти осуществляется через две шины АНВ: I-code (для команд) и D-code (для данных). Чтение D-code шины имеет приоритет. На обеих шинах при попытке записи в любую область, чтении из несуществующей области, чтении во время, когда Flash занята (стирание, запись), транзакция проходит успешно с неопределенными данными на выходе.

Память доступна для чтения, записи, полного и постраничного стирания через регистры данных DATA<sub>n</sub> (n от 0 до 3), адреса ADDR, команд CMD, статуса STAT блока BFLASH. Запись необходимо производить в предварительно очищенную ячейку памяти. Стирание памяти осуществляется полностью или постранично.

Минимальное время чтения данных из Flash-памяти составляет около 30 нс. Поэтому, исходя из выбранной рабочей частоты, следует задать количество дополнительных тактов ожидания, необходимое для стабильного чтения из Flash-памяти. Данный параметр заносится в поле LAT регистра CTRL, см. таблицу 7.1

Помимо основной области памяти (512 Кбайт), доступной как через глобальное адресное пространство, так и через регистры контроллера, существует дополнительная (NVR) область, доступная только для операций через регистры. Размер этой области 16 Кбайт (4 страницы по 4 Кбайт). Для выполнения операций с этой областью необходимо, чтобы вместе с командой был установлен бит NVRON в регистре CMD. Данную память можно использовать для хранения пользовательских данных.

Операция предвыборки в загрузочной памяти включена всегда и функционирует аналогично механизму, используемому в основной памяти. Кэш память не используется.

### Конфигурационное слово CFGWORD

На последней странице дополнительной области загрузочной Flash-памяти расположено конфигурационное слово CFGWORD, состоящее из двух 32-битных слов CFGWORD0 и CFGWORD1 и содержащее параметры микроконтроллера. Описание битов представлено в таблицах 7.2 и 7.3. Чтение конфигурационного слова контроллером происходит каждый раз после сброса по включению (POR).

Таблица 7.2 – Конфигурационное слово CFGWORD0

CFGWORD0 +3FF0h (смещение относительно начального адреса NVR области BFLASH)		
Поле	Биты	Описание
MASK	31-8	Битовая маска адреса. Установленные биты соответствуют тем выводам, которые будут переведены в режим альтернативной функции при загрузке контроллера из внешней памяти. Например, FFFh – используются все 24 бита адреса, 00F – будут использоваться только выводы, соответствующие первым восьми битам адреса
WRC	7-4	Длительность цикла записи слова данных. Поле задает длительность цикла записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0h соответствует одному такту, значение Fh – шестнадцать
RDC	3-0	Длительность цикла чтения слова данных. Поле задает длительность цикла чтения на внешней шине в количестве тактов сигнала системной частоты. Значение 0h соответствует одному такту, значение Fh – шестнадцать

Таблица 7.3 – Конфигурационное слово CFGWORD1

CFGWORD1 +3FF4h (смещение относительно начального адреса NVR области BFLASH)		
Поле	Биты	Описание
DEBUGEN	11	Бит разрешения работы системы отладки ядра
		0   Отладка отключена
		1   Отладка включена (по умолчанию)
JTAGEN	10	Бит разрешения работы пинов JTAG/SWD
		0   Работа запрещена
		1   Работа разрешена (по умолчанию)
BNVRWE	9	Бит включения защиты NVR области загрузочной Flash-памяти
		0   Запрет записи и стирания NVR блока
		1   Защита выключена (по умолчанию)
BFLASHWE	8	Бит включения защиты основной области загрузочной Flash-памяти
		0   Запрет записи и стирания основного блока
		1   Защита выключена (по умолчанию)

Окончание таблицы 7.3

Поле	Биты	Описание
MNVRWE	7	Бит включения защиты NVR области основной Flash-памяти
		0   Запрет записи и стирания NVR блока
		1   Защита выключена (по умолчанию)
MFLASHWE	6	Бит включения защиты основной области основной Flash-памяти
		0   Запрет записи и стирания основного блока
		1   Защита выключена (по умолчанию)
AF	5	Выбор альтернативной функции для интерфейса внешней памяти, используемого при загрузке
		0   Первая функция
		1   Вторая функция
MODE	4	Выбор режима ширины шины данных, используемого при загрузке
		0   8-битный режим
		1   16-битный режим
TAC	3-0	Длительность цикла переключения шины. Поле задает паузу между циклами чтения и записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0h соответствует одному такту, значение Fh – шестнадцать
–	31-12	Зарезервировано

### 7.3 Сервисный сброс всей Flash-памяти

1 Во время сброса микроконтроллера анализируется состояние вывода SERVEN. Если вывод находится в состоянии логической единицы (подтянут к 3,3 В), то загрузочная и основная Flash-памяти переводятся в режим, в котором чтение запрещено (при чтении возвращаются нули).

2 Далее по отладочному интерфейсу (SWD или JTAG) должна быть подана команда записи значения 0000\_0001h в регистр SERVCTL блока SIU, после чего будет активировано полное стирание. По завершении процесса стирания в этом же регистре выставится флаг DONE

Примечание – Если полное стирание не требуется, во время сброса на выводе SERVEN должен удерживаться логический ноль.

## 8 Контроллер внешней памяти

Контроллер внешней памяти позволяет осуществлять коммутацию процессора микроконтроллера с внешней памятью типа SRAM. Контроллер может работать в 8-/16-разрядном режиме и согласовывает сигналы АНВ-шины с сигналами внешней памяти. На рисунке 8.1 показаны временные диаграммы циклов чтения и записи при работе с внешней памятью, описание обозначений приведено в таблице 8.1.

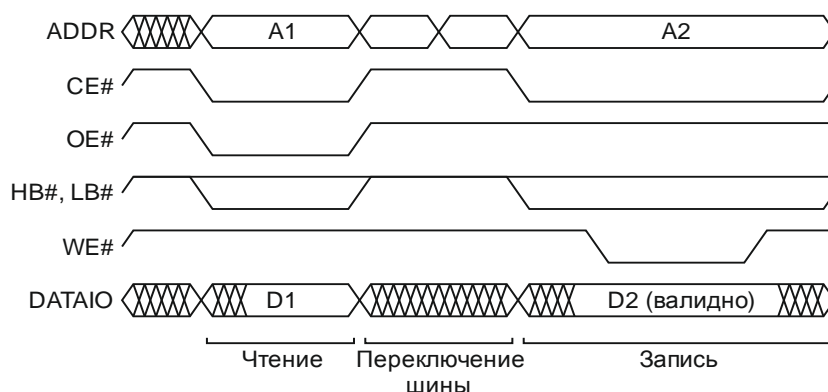


Рисунок 8.1 – Временная диаграмма работы контроллера внешней памяти

Таблица 8.1 – Сигналы, используемые при работе с внешней памятью

Обозначение сигнала	Вывод микроконтроллера	Назначение
ADDR	EXTMEM_ADDR	Шина адреса для чтения/записи. Сигнал, передаваемый во внешнюю память
CE#	EXTMEM_CEn	Сигнал выбора внешнего устройства
OE#	EXTMEM_OEn	Сигнал (строб) разрешения чтения для внешнего устройства
HB#, LB#	EXTMEM_UBn, EXTMEM_LBn	Сигналы разрешения для старшего и младшего байт
WE#	EXTMEM_WEn	Сигнал (строб) записи для внешнего устройства
DATAIO	EXTMEM_DATA	Данные внешней памяти

Примечание – Сигналы с символом «#» имеют низкий активный уровень.

Максимально доступный диапазон адресного пространства в 128 Мбайт (ширина адресной шины 24 бита) разбит на 8 окон по 16 Мбайт, см. таблицу 8.2.

Таблица 8.2 – Адресное пространство внешней памяти

Номер окна	Сигнал выбора	Адрес начала	Адрес конца
7	EXTMEM_CEn[7]	0x8700_0000	0x87FF_FFFF
6	EXTMEM_CEn[6]	0x8600_0000	0x86FF_FFFF
5	EXTMEM_CEn[5]	0x8500_0000	0x85FF_FFFF
4	EXTMEM_CEn[4]	0x8400_0000	0x84FF_FFFF
3	EXTMEM_CEn[3]	0x8300_0000	0x83FF_FFFF
2	EXTMEM_CEn[2]	0x8200_0000	0x82FF_FFFF
1	EXTMEM_CEn[1]	0x8100_0000	0x81FF_FFFF
0	EXTMEM_CEn[0]	0x8000_0000	0x80FF_FFFF

Каждое окно может использовать индивидуальные настройки разрядности и циклов доступа. Также каждому окну соответствует свой сигнал выбора EXTMEM\_CEn, который вырабатывается автоматически при обращении к адресу, попадающему в соответствующее окно.

Управление контроллером внешней памяти осуществляется посредством группы восьми регистров WINCFG для каждого окна в отдельности. Разрядность шины задается битом MODE. В контроллере внешней памяти предусмотрен цикл переключения шины между циклами чтения и записи. Это реализовано в целях предотвращения кратковременных скачков тока в то время, когда процессор и внешнее устройство пытаются одновременно управлять шиной данных. Настройка длительности циклов чтения, записи и переключения осуществляется посредством полей RDCYC, WRCYC и TACYC. На рисунке 8.1 показан пример для случая, когда RDCYC = WRCYC = TACYC = 001h, т. е. длительность каждого цикла составляет два такта сигнала системной частоты.

Для подключения внешней памяти можно использовать две группы выводов микроконтроллера. В одной группе используется первая альтернативная функция выводов, во второй – вторая. Два варианта подключения внешней памяти к микроконтроллеру отображены в таблице 8.3.

Таблица 8.3 – Подключение внешней памяти к выводам микроконтроллера

Альтернативная функция 1		Альтернативная функция 2		Интерфейс	
GPIO	Вывод	GPIO	Вывод		
D14	U17	K8	T3	EXTMEM_WEn	Сигнал записи во внешнюю память
D15	V17	K9	U3	EXTMEM_OEn	Сигнал чтения для внешнего устройства
D13	V16	K11	V3	EXTMEM_UBn	Сигналы разрешения для старшего байта
D12	U16	K10	V2	EXTMEM_LBn	Сигналы разрешения для младшего байта
D11	V15	G15	V11	EXTMEM_CEn[7]	Сигнал выбора внешнего устройства окна 7
D10	U15	G14	V10	EXTMEM_CEn[6]	Сигнал выбора внешнего устройства окна 6
D9	V14	G13	U10	EXTMEM_CEn[5]	Сигнал выбора внешнего устройства окна 5
D8	U14	G12	T10	EXTMEM_CEn[4]	Сигнал выбора внешнего устройства окна 4
D7	V13	G11	S10	EXTMEM_CEn[3]	Сигнал выбора внешнего устройства окна 3
D6	U13	G10	R10	EXTMEM_CEn[2]	Сигнал выбора внешнего устройства окна 2
D5	T13	G9	V9	EXTMEM_CEn[1]	Сигнал выбора внешнего устройства окна 1
D4	V12	G8	U9	EXTMEM_CEn[0]	Сигнал выбора внешнего устройства окна 0
G7	T9	K7	S4	EXTMEM_ADDR[23]	Шина адреса
G6	S9	K6	S1	EXTMEM_ADDR[22]	
G5	V8	K5	S2	EXTMEM_ADDR[21]	
G4	U8	K4	R1	EXTMEM_ADDR[20]	
G3	V7	K3	R2	EXTMEM_ADDR[19]	
G2	U7	K2	Q2	EXTMEM_ADDR[18]	
G1	V6	K1	Q3	EXTMEM_ADDR[17]	
G0	U6	K0	P1	EXTMEM_ADDR[16]	
F15	M20	J15	P2	EXTMEM_ADDR[15]	

Окончание таблицы 8.3

Альтернативная функция 1		Альтернативная функция 2		Интерфейс	
GPIO	Вывод	GPIO	Вывод		
F14	J17	J14	M3	EXTMEM_ADDR[14]	Шина адреса
F13	J16	J13	N2	EXTMEM_ADDR[13]	
F12	K20	J12	M1	EXTMEM_ADDR[12]	
F11	K18	J11	L5	EXTMEM_ADDR[11]	
F10	L20	J10	N5	EXTMEM_ADDR[10]	
F9	L19	J9	L3	EXTMEM_ADDR[9]	
F8	L18	J8	L2	EXTMEM_ADDR[8]	
F7	L17	J7	L1	EXTMEM_ADDR[7]	
F6	M19	J6	K5	EXTMEM_ADDR[6]	
F5	M17	J5	K4	EXTMEM_ADDR[5]	
F4	N20	J4	K3	EXTMEM_ADDR[4]	
F3	N19	J3	K2	EXTMEM_ADDR[3]	
F2	N17	J2	K1	EXTMEM_ADDR[2]	
F1	N16	J1	J1	EXTMEM_ADDR[1]	
F0	P20	J0	J3	EXTMEM_ADDR[0]	
E15	P19	H15	H2	EXTMEM_DATA[15]	Шина данных
E14	P18	H14	H1	EXTMEM_DATA[14]	
E13	Q19	H13	G5	EXTMEM_DATA[13]	
E12	Q20	H12	G2	EXTMEM_DATA[12]	
E11	R19	H11	G1	EXTMEM_DATA[11]	
E10	R20	H10	F4	EXTMEM_DATA[10]	
E9	S19	H9	F2	EXTMEM_DATA[9]	
E8	S20	H8	F1	EXTMEM_DATA[8]	
E7	T19	H7	E2	EXTMEM_DATA[7]	
E6	T20	H6	E1	EXTMEM_DATA[6]	
E5	V20	H5	D3	EXTMEM_DATA[5]	
E4	V19	H4	D2	EXTMEM_DATA[4]	
E3	V18	H3	D1	EXTMEM_DATA[3]	
E2	U18	H2	C2	EXTMEM_DATA[2]	
E1	S17	H1	C1	EXTMEM_DATA[1]	
E0	S16	H0	B1	EXTMEM_DATA[0]	

Для загрузки с внешней памяти необходимо в конфигурационном слове CFGWORD (см. подраздел 7.2):

- задать поля TAC, WRC, RDC, отвечающие за циклы переключения, записи и чтения соответственно;
- выбрать номер альтернативной функции портов, куда подключена память, установив поле AF;
- выбрать ширину шины данных, задав поле MODE;
- задать маску (поле MASK) шины адреса, определив, какие из выводов адресной шины будут переключены для работы с внешней памятью.

После данных настроек необходимо подтянуть входы BOOT1, BOOT0 к состоянию 11b. Тогда после сброса (по сигналу от POR), контроллер произведет старт с внешней памяти.

При использовании маски адреса и 8-разрядной шины данных, неиспользуемые разряды не будут проинициализированы как выходы с альтернативными функциями, оставаясь портами общего назначения. При загрузке с внешней памяти автоматически включается альтернативная функция только для сигнала EXTMEM\_CEn[0] – внешняя память подключается к нулевому окну. Другие сигналы EXTMEM\_CEn можно разрешить уже в процессе работы программы.

## 9 Контроллер прямого доступа к памяти DMA

Основные свойства и отличительные особенности:

- 32 канала контроллера прямого доступа к памяти DMA;
- каждый канал DMA<sub>i</sub> (где i от 0 до 31) имеет свои сигналы управления передачей данных и программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается исходя из уровня приоритета, определяемого номером канала DMA<sub>i</sub>;
- поддержка различного типа передачи данных в пределах внутреннего ОЗУ: память – память, память – периферия, периферия – память;
- поддержка различных типов циклов DMA;
- поддержка циклического режима передачи;
- поддержка передачи данных различной разрядности;
- каждому каналу DMA<sub>i</sub> доступна первичная и альтернативная структура управляющих данных канала;
- все данные канала хранятся в ОЗУ в структуре управляющих данных канала;
- разрядность данных приемника равна разрядности данных передатчика;
- количество передач в одном цикле может программироваться от 1 до 1024;
- инкремент адреса передачи может быть больше, чем разрядность данных;
- возможность начать передачи по сигналам от блоков: UART, SPI, ADC, TMR, ETMR, PWM, QEP, SDFM.

Аппаратные источники запросов каналов контроллера DMA указаны в таблице 9.1.

Таблица 9.1 – Аппаратные источники запросов каналов контроллера DMA

Номер канала	Аппаратный источник запросов	Описание
0	UART0_TX	Канал DMA от UART0 по передаче
1	UART1_TX	Канал DMA от UART1 по передаче
2	UART2_TX	Канал DMA от UART2 по передаче
3	UART3_TX	Канал DMA от UART3 по передаче
4	UART0_RX	Канал DMA от UART0 по приему
5	UART1_RX	Канал DMA от UART1 по приему
6	UART2_RX	Канал DMA от UART2 по приему
7	UART3_RX	Канал DMA от UART3 по приему
8	ADC_SEQ0	Канал DMA от секвенсора 0 блока АЦП
9	ADC_SEQ1	Канал DMA от секвенсора 1 блока АЦП
10	ADC_SEQ2	Канал DMA от секвенсора 2 блока АЦП
11	ADC_SEQ3	Канал DMA от секвенсора 3 блока АЦП
12	ADC_SEQ4	Канал DMA от секвенсора 4 блока АЦП
13	ADC_SEQ5	Канал DMA от секвенсора 5 блока АЦП
14	ADC_SEQ6	Канал DMA от секвенсора 6 блока АЦП
15	ADC_SEQ7	Канал DMA от секвенсора 7 блока АЦП
16	SPI0_TX	Канал DMA от SPI0 по передаче
17	SPI1_TX	Канал DMA от SPI1 по передаче
18	SPI2_TX	Канал DMA от SPI2 по передаче
19	SPI3_TX	Канал DMA от SPI3 по передаче
20	SPI0_RX	Канал DMA от SPI0 по приему
21	SPI1_RX	Канал DMA от SPI1 по приему
22	SPI2_RX	Канал DMA от SPI2 по приему
23	SPI3_RX	Канал DMA от SPI3 по приему
24-31	В соответствии с регистром DMAMUX блока SIU	Каналы DMA с конфигурируемыми источниками запросов



## 9.1 Программное управление контроллером DMA

Контроллер DMA выполняет передачи 8-, 16- и 32-разрядных данных. Разрядность данных источника и приемника должны быть одинаковыми.

Контроллер DMA позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса зависит от разрядности передаваемых данных: минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных; максимальная величина – одно слово. Контроллер DMA может быть настроен на работу с фиксированным адресом (например, для работы с FIFO).

Контроллер DMA имеет возможность обслуживать сигналы запроса на одиночный обмен SREQ и сигналы запроса на пакетный обмен BREQ блоков UART, SPI. Блок ADC генерирует только запросы на пакетный обмен BREQ.

Каждому каналу контроллера DMA соответствуют две структуры управляющих данных: первичная и альтернативная. В ОЗУ должна быть отведена область для хранения этих структур.

На рисунке 9.1 показана область памяти, необходимая контроллеру для структур управляющих данных каналов, при использовании всех 32 каналов и опциональной альтернативной структуры данных.

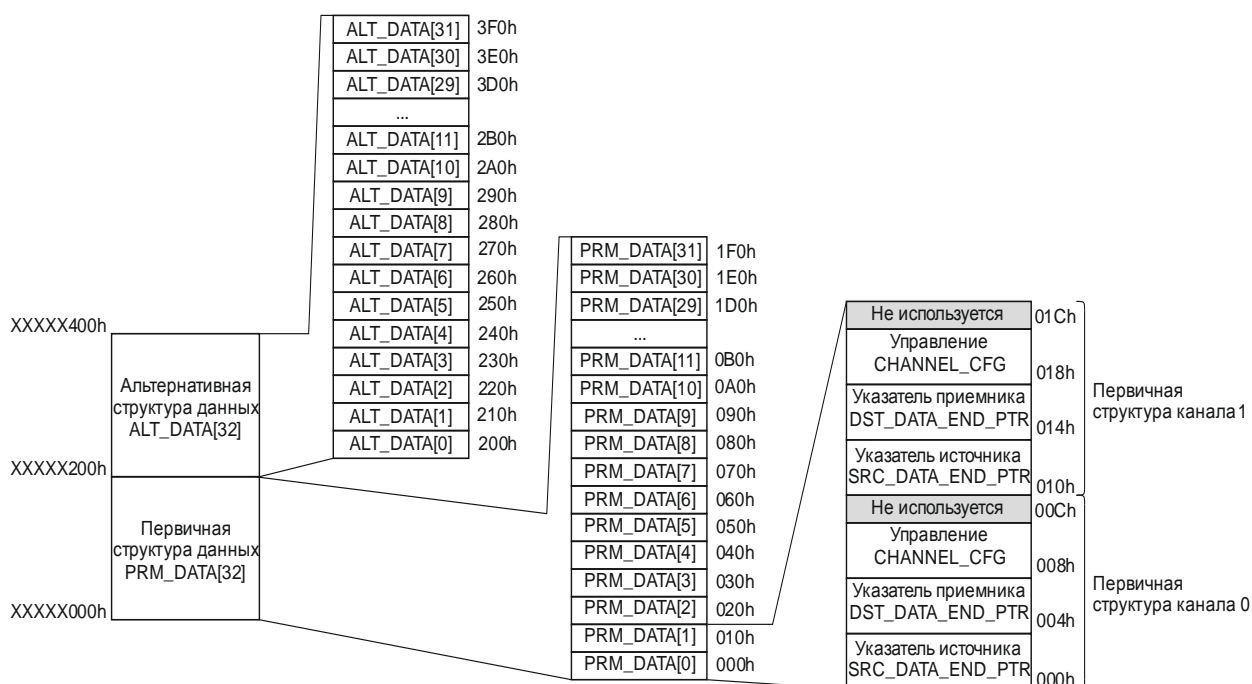


Рисунок 9.1 – Карта памяти для 32 каналов контроллера DMA, включая альтернативную структуру

Объем структуры, показанной на рисунке 9.1, составляет 1024 байт. Контроллер использует младшие разряды адреса для доступа ко всем элементам структуры управляющих данных, и поэтому разрешенные значения базового адреса для первичной структуры управляющих данных – XXXX\_X000h, XXXX\_X400h, XXXX\_X800h и т. д.

Базовый адрес для первичной структуры управляющих данных возможно установить путем записи соответствующего значения в регистр BASEPTR.

В таблице 9.2 перечислены разряды адреса, обеспечивающие контроллеру DMA доступ к различным элементам структуры управляющих данных.

Таблица 9.2 – Разряды адреса, используемые для доступа к управляющим данным 32 каналов контроллера DMA

Разряды адреса										
	9	8	7	6	5	4	3	2	1	0
	S	CHNL					EL			
Обозначение	Биты	Действие								
S	9	Выбор структуры управляющих данных:								
		0	Первичная							
		1	Альтернативная							
CHNL	8-4	Выбор канала. Допустимые значения 0h-1Fh								
EL	3-0	Выбор управляющего элемента:								
		0h	Указатель конца данных источника							
		4h	Указатель конца данных приемника							
		8h	Конфигурация структуры управляющих данных							
		Ch	Не используется. Контроллер не имеет доступа к этому адресу							

Не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, он вычисляется автоматически и помещается в регистр ALTBASEPTR.

Любая из структур управляющих данных каждого канала состоит из двух указателей адреса (приемника и источника данных) и ячейки управления канала.

#### Управление канала CHANNEL\_CFG

32-разрядная ячейка памяти, содержащая конфигурационную информацию для осуществления передач DMA (на рисунке 9.1 отмечена как «Управление ...»). В начале цикла DMA или в начале  $2^R$  передачи контроллер DMA считывает значение этой ячейки. После выполнения  $2^R$  или N передач он сохраняет обновленное ее значение обратно в память. Структура регистра CHANNEL\_CFG приведена в таблице 9.3.

Таблица 9.3 – Структура управляющих данных канала

CHANNEL_CFG															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DST_INC		DST_SIZE		SRC_INC		SRC_SIZE		DST_PROT_CTRL			SRC_PROT_CTRL			R_POWER	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_POWER		N_MINUS_1									NEXT_USEBU_RST	CYCLE_CTRL			

Продолжение таблицы 9.3

Поле	Биты	Описание			
DST_INC	31-30	Шаг инкремента адреса приемника. Код, записанный в поле DST_INC, задает шаг, который в свою очередь зависит от разрядности данных источника.			
		Код	Разрядность данных источника		
			Байт	Слово (16 бит)	Двойное слово (32 бита)
		00	Шаг – байт	Зарезервировано	Зарезервировано
		01	Шаг – слово (16 бит)		Зарезервировано
		10	Шаг – двойное слово (32 бита)		
11	Нет инкремента. Адрес остается равным значению ячейки DST_DATA_END_PTR				
DST_SIZE	29-28	Разрядность данных приемника. Значение этого поля должно быть равно значению поля SRC_SIZE. Примечание – Если контроллер обнаруживает неравные значения этих полей, он при ближайшем обновлении поля N_MINUS_1 устанавливает значение поля DST_SIZE, равное SRC_SIZE.			
SRC_INC	27-26	Шаг инкремента адреса источника. Код, записанный в поле SRC_INC, задает шаг, который в свою очередь зависит от разрядности данных источника.			
		Код	Разрядность данных источника		
			Байт	Слово (16 бит)	Двойное слово (32 бита)
		00	Шаг – байт	Зарезервировано	Зарезервировано
		01	Шаг – слово (16 бит)		Зарезервировано
		10	Шаг – двойное слово (32 бита)		
11	Нет инкремента. Адрес остается равным значению ячейки SRC_DATA_END_PTR				
SRC_SIZE	25-24	Разрядность данных источника			
		00	Байт		
		01	Слово (16 бит)		
		10	Двойное слово (32 бита)		
		11	Зарезервировано. Не использовать!		
DST_PROT_CTRL	23-21	Задает параметры защиты шины АНВ при записи данных в приемник			
		Код	Биты поля DST_PROT_CTRL		
			23	22	21
		0	Доступ не кэшируется	Доступ не буферизуется	Доступ непривилегированный
1	Доступ кэшируется	Доступ буферизуется	Доступ привилегированный		
SRC_PROT_CTRL	20-18	Задает параметры защиты шины АНВ при чтении данных из источника			
		Код	Биты поля SRC_PROT_CTRL		
			20	19	18
		0	Доступ не кэшируется	Доступ не буферизуется	Доступ непривилегированный
1	Доступ кэшируется	Доступ буферизуется	Доступ привилегированный		

Продолжение таблицы 9.3

Поле	Биты	Описание																								
R_POWER	17-14	<p>Параметр R.                      Задаёт количество передач канала DMA до выполнения контроллером DMA процедуры арбитража (перearбитрации).                      Количество передач равно <math>2^R</math></p>																								
		<table border="1"> <thead> <tr> <th>Код</th> <th>Количество передач</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>1 (арбитраж производится после каждой передачи DMA)</td> </tr> <tr> <td>1h</td> <td>2</td> </tr> <tr> <td>2h</td> <td>4</td> </tr> <tr> <td>3h</td> <td>8</td> </tr> <tr> <td>4h</td> <td>16</td> </tr> <tr> <td>5h</td> <td>32</td> </tr> <tr> <td>6h</td> <td>64</td> </tr> <tr> <td>7h</td> <td>128</td> </tr> <tr> <td>8h</td> <td>256</td> </tr> <tr> <td>9h</td> <td>512</td> </tr> <tr> <td>Ah – Fh</td> <td>1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024)</td> </tr> </tbody> </table>	Код	Количество передач	0h	1 (арбитраж производится после каждой передачи DMA)	1h	2	2h	4	3h	8	4h	16	5h	32	6h	64	7h	128	8h	256	9h	512	Ah – Fh	1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024)
		Код	Количество передач																							
		0h	1 (арбитраж производится после каждой передачи DMA)																							
		1h	2																							
		2h	4																							
		3h	8																							
		4h	16																							
		5h	32																							
		6h	64																							
		7h	128																							
		8h	256																							
		9h	512																							
		Ah – Fh	1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024)																							
<p>Примечание – Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.</p>																										
N_MINUS_1	13-4	<p>Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл</p>																								
		<table border="1"> <thead> <tr> <th>Код</th> <th>Количество передач</th> </tr> </thead> <tbody> <tr> <td>000h</td> <td>1</td> </tr> <tr> <td>001h</td> <td>2</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>3FFh</td> <td>1024</td> </tr> </tbody> </table>	Код	Количество передач	000h	1	001h	2	...	...	3FFh	1024														
		Код	Количество передач																							
		000h	1																							
		001h	2																							
		...	...																							
3FFh	1024																									
<p>Примечание – Контроллер DMA обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения всего цикла DMA.</p>																										
NEXT_USEBURST	3	<p>Контролирует установку соответствующего каналу бита в регистре USEBURSTSET, если контроллер работает в периферийном режиме «разборка-сборка» и завершает цикл DMA, используя альтернативные управляющие данные.</p>																								
		<p>Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер DMA сбрасывает соответствующий каналу бит в регистре USEBURSTSET, если количество оставшихся передач DMA меньше, чем <math>2^R</math>. Программирование бита NEXT_USEBURST определяет, будет ли контроллер DMA дополнительно переопределять состояние бита в регистре USEBURSTSET.</p>																								
		<p>Если контроллер выполняет цикл DMA в периферийном режиме «разборка-сборка», то после окончания цикла, использующего альтернативные управляющие данные, дальнейшие действия будут зависеть от состояния бита NEXT_USEBURST</p>																								

Окончание таблицы 9.3

Поле	Биты	Описание	
NEXT_USE BURST	3	0	Контроллер DMA не изменяет значение соответствующего каналу бита в регистре USEBURSTSET. Если бит CH <sub>i</sub> в USEBURSTSET сброшен, то при выполнении циклов DMA с использованием альтернативных управляющих данных контроллер DMA отвечает как на запросы BREQ, так и на запросы SREQ от периферии
		1	Контроллер DMA изменяет значение соответствующего каналу бита в регистре USEBURSTSET, а именно – устанавливает бит. Поэтому для оставшихся циклов DMA с использованием альтернативных управляющих данных контроллер DMA реагирует только на запросы BREQ от периферии
CYCLE_CTRL	2-0	Поле задания типа цикла DMA	
		000b	Недействительный. Структура управляющих данных канала в запрещенном состоянии
		001b	Основной
		010b	Авто-запрос
		011b	«Пинг-понг»
		100b	Работа с памятью в режиме «разборка-сборка» с использованием первичных управляющих данных канала
		101b	Работа с памятью в режиме «разборка-сборка» с использованием альтернативных управляющих данных канала
		110b	Работа с периферией в режиме «разборка-сборка» с использованием первичных управляющих данных канала
		111b	Работа с периферией в режиме «разборка-сборка» с использованием альтернативных управляющих данных канала
		Примечание – После завершения всего цикла передач контроллер DMA устанавливает значение поля CYCLE_CTRL в 000b, переводя тем самым тип цикла в «недействительный». Это позволяет избежать повторения выполненной передачи DMA.	

**Указатель конца данных источника SRC\_DATA\_END\_PTR и указатель конца данных приемника DST\_DATA\_END\_PTR**

Указатель конца данных источника SRC\_DATA\_END\_PTR и указатель конца данных приемника DST\_DATA\_END\_PTR – 32-разрядные ячейки памяти, которые содержат адрес месторасположения конца данных источника и приемника соответственно. Перед тем как контроллер DMA выполнит передачу, необходимо определить их значения. Контроллер DMA считывает значения этих областей перед началом  $2^R$  передач.

Для вычисления адреса источника передачи контроллер DMA выполняет сдвиг влево значения N\_MINUS\_1 на количество разрядов, соответствующее полю SRC\_INC, и затем вычитает получившееся значение от значения SRC\_DATA\_END\_PTR.

Подобным образом вычисляется начальный адрес приемника/источника, и контроллер DMA выполняет сдвиг влево значения N\_MINUS\_1 на количество разрядов, соответствующее полю DST\_INC, и затем вычитает получившееся значение от значения DST\_DATA\_END\_PTR.

## 9.2 Правила обмена данными

Следует избегать адресации 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.

Системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0000\_0000h, если не указано иное.

Контроллер DMA использует правила обмена данными, см. таблицу 9.4, при соблюдении следующих условий:

- канал контроллера DMA включен (установлен соответствующий каналу бит в регистре ENSET и бит MASTEREN в регистре CFG);
- запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре REQMASKSET).

Таблица 9.4 – Перечень правил, при которых передачи по каналам DMA<sub>i</sub> разрешены и запросы не маскируются (i – номер канала)

Номер правила	Описание
1	Если канал не активен (передача не идет в данный момент), то установка бита CH <sub>i</sub> в регистре SWREQ или запрос от соответствующей периферии инициирует передачу по каналу i
2	Одновременно активен может быть только один канал
3	Если запрос от периферии происходит в момент, когда канал активен, то контроллер обслужит этот запрос после завершения текущей передачи
4	Если приходит сразу несколько запросов от периферии для одного канала в момент, когда канал активен, то контроллер обслужит только первый запрос после завершения текущей передачи
5	Для циклов DMA, отличных по типу от периферийного режима «разборка-сборка», по окончании 2 <sup>R</sup> передач контроллер DMA сбрасывает бит CH <sub>i</sub> в регистре USEBURSTSET, если количество оставшихся передач меньше, чем 2 <sup>R</sup> , позволяя периферии завершить передачи, используя как SREQ запросы, так и BREQ. В периферийном режиме «разборка-сборка» контроллер сбрасывает бит CH <sub>i</sub> в регистре USEBURSTSET, только если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2 <sup>R</sup>
6	Контроллер DMA игнорирует запрос SREQ, если бит CH <sub>i</sub> регистра WAITONREQ сброшен или установлен бит CH <sub>i</sub> регистра USEBURSTSET
7	Необходимо с осторожностью устанавливать разряды регистра USEBURSTSET. Если значение, указанное в регистре N_MINUS_1 меньше, чем значение 2 <sup>R</sup> , то контроллер DMA не очистит разряды USEBURSTSET, и поэтому одиночные запросы SREQ будут запрещены. Если программные запросы через регистр SWREQ не генерируются, и периферия не осуществляет запросов на пакетную обработку BREQ, то контроллер DMA никогда не выполнит необходимых передач

Окончание таблицы 9.4

Номер правила	Описание
8	Для типов циклов DMA, отличных от периферийного режима «разборка-сборка», если придет запрос SREQ, то контроллер DMA выполнит одну передачу. В периферийном режиме «Исполнение с изменением конфигурации», если придет запрос SREQ, контроллер DMA выполняет $2^R$ передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
9	Для типов циклов DMA, отличных от периферийного режима «разборка-сборка», если одновременно пришли запросы SREQ и BREQ, то приоритет предоставляется BREQ, и контроллер DMA выполняет $2^R$ передач (или число передач, указанное в поле N_MINUS_1). В периферийном режиме «разборка-сборка», если одновременно пришли запросы SREQ и BREQ, то приоритет также предоставляется BREQ, и контроллер DMA выполняет $2^R$ передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет $2^R$ передач (или число передач, указанное в поле N_MINUS_1), используя альтернативную структуру управляющих данных
10	В периферийном режиме «разборка-сборка», если бит NEXT_USEBURST в CHANNEL_CFG установлен, то контроллер DMA устанавливает соответствующий каналу бит в регистре USEBURSTSET после окончания цикла DMA, использующего альтернативные управляющие данные
11	Когда установлен бит CHi регистра REQMASKSET, контроллер DMA игнорирует запросы SREQ и BREQ

При отключении канала (бит CHi регистра ENSET сброшен) контроллер DMA осуществляет передачи согласно правилам, представленным в таблице 9.5.

Таблица 9.5 – Перечень правил осуществления передач для запрещенных каналов

Номер правила	Описание
1	Если приходит запрос на пакетную обработку BREQ от периферии, то происходит вызов прерывания канала контроллера DMA (если было включено). Это позволяет сигнализировать о запросе, даже если канал выключен
2	Если приходит запрос на одиночную передачу SREQ от периферии, то происходит вызов прерывания канала контроллера DMA (если было включено) при условии, что бит CHi регистра WAITONREQ установлен, а бит CHi регистра USEBURSTSET сброшен. Это позволяет сигнализировать о запросе, даже если канал выключен

### 9.3 Правила арбитража

Контроллер DMA имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер DMA имеет настройки, которые определяют количество передач по шине АНВ до повторения арбитража (перearбитрации). Это значение задается параметром R (поле R\_POWER в регистре CHANNEL\_CFG структуры управляющих данных канала).

Количество транзакций одного канала до переарбитрации при этом равно  $2^R$ . Например, если  $R = 4$ , то арбитраж будет проводиться через каждые 16 передач DMA.

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 если результат деления  $2^R$  на  $N$  не целое число, контроллер всегда выполняет последовательность из  $2^R$  передач до тех пор, пока не выполнится условие  $N < 2^R$ . Контроллер выполняет оставшиеся  $N$  передач в конце цикла DMA.

### 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 или высокий. Изменение уровня приоритета осуществляется установкой соответствующего бита  $CH_i$  ( $i$  – номер канала) в регистрах  $PRIORITYSET$  и  $PRIORITYCLR$ . Канал 0 имеет наивысший уровень приоритета.

Порядок каналов по уменьшению уровня приоритета представлен в таблице 9.6.

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов  $DMA_i$ . Рисунок 9.2 иллюстрирует процесс выбора следующего канала для обслуживания.

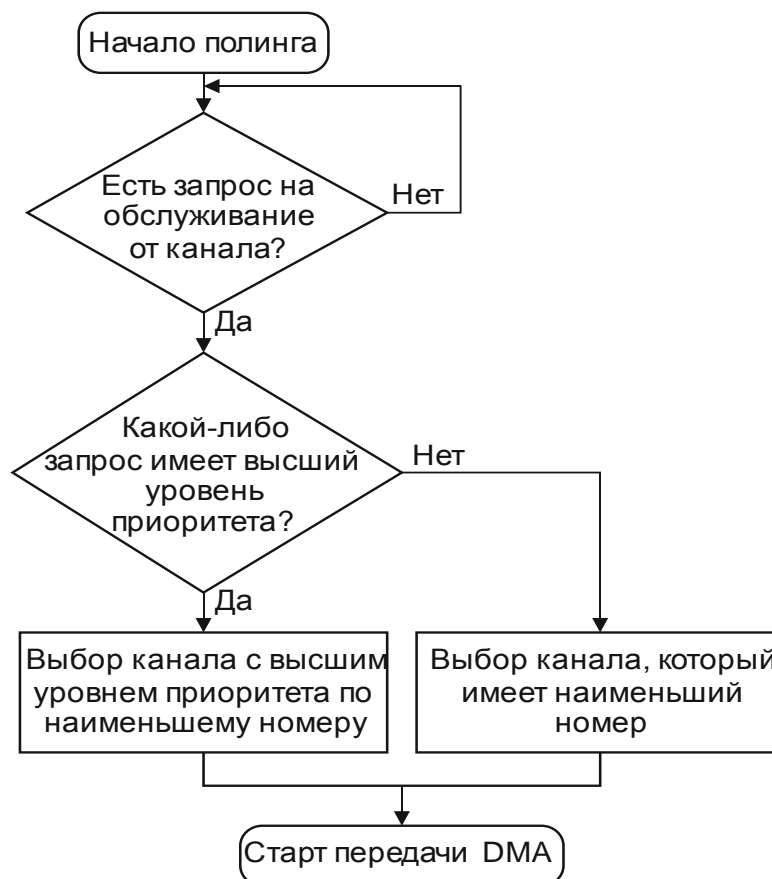



Рисунок 9.2 – Алгоритм выбора (полинга) следующего канала для обслуживания



Таблица 9.6 – Распределение приоритетов

Номер канала CHi	Состояние бита CHi в регистре PRIORITYSET	Уровень приоритета	Порядок изменения уровня приоритета
0	1	Высокий	Снижение уровня приоритета 
1	1	Высокий	
2	1	Высокий	
3	1	Высокий	
...	...	...	
30	1	Высокий	
31	1	Высокий	
0	0	По умолчанию	
1	0	По умолчанию	
2	0	По умолчанию	
3	0	По умолчанию	
...	...	...	
30	0	По умолчанию	
31	0	По умолчанию	

#### 9.4 Типы циклов

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.

Поддерживаются следующие типы циклов DMA:

- недействительный (структура управляющих данных канала в запрещенном состоянии);
- основной;
- авто-запрос;
- «пинг-понг»;
- работа с памятью в режиме «разборка-сборка» (scatter-gather);
- работа с периферией в режиме «разборка-сборка».

Задание типа цикла DMA осуществляется программированием поля CYCLE\_CTRL регистра CHANNEL\_CFG структуры управляющих данных канала.

##### Недействительный цикл

После окончания цикла контроллер DMA устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

##### Основной цикл

В данном режиме контроллер DMA работает либо с первичными, либо с альтернативными управляющими данными канала, совершая по  $2^R$  передач по каждому запросу.

Перед началом работы необходимо включить контроллер DMA и разрешить работу канала: установить соответствующий каналу бит в регистре ENSET и бит MASTEREN в регистре CFG, а также проверить, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре REQMASKSET).

После того как разрешена работа канала, цикл DMA выглядит следующим образом:

1 Контроллер DMA ожидает получения запроса (программного либо от периферии) на обработку. Если запрос получен, то контроллер DMA переходит к шагу 2.

2 Контроллер DMA выполняет  $2^R$  передач. Если число оставшихся передач 0, контроллер DMA переходит к шагу 4, иначе выполняется шаг 3.

3 Происходит осуществление арбитража: если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала, иначе происходит ожидание очередного запроса на обработку по каналу, и если периферийный блок или программа его выдает, то контроллер DMA переходит к шагу 2.

4 Контроллер DMA указывает центральному процессору на завершение цикла DMA. Вызывается соответствующее каналу прерывание (если было включено).

### **Авто-запрос**

Контроллеру DMA необходим лишь одиночный запрос 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 программы или периферийных блоков. Контроллер DMA позволяет выбрать для использования либо первичную, либо альтернативную структуру управляющих данных канала.

Перед началом работы необходимо включить контроллер DMA и разрешить работу канала: установить соответствующий каналу бит в регистре ENSET и бит MASTEREN в регистре CFG, а также проверить, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре REQMASKSET).

После того как разрешена работа канала, цикл DMA выглядит следующим образом:

1 Контроллер DMA ожидает получения запроса (программного либо от периферии) на обработку. Если запрос получен, то контроллер DMA переходит к шагу 2.

2 Контроллер выполняет  $2^R$  передач. Если число оставшихся передач 0, контроллер DMA переходит к шагу 4, иначе выполняется шаг 3.

3 Осуществление арбитража: если высокоприоритетный канал выдает запрос на обработку, то контроллер DMA начинает обслуживание этого канала, иначе контроллер переходит к шагу 2.

4 Контроллер DMA указывает центральному процессору на завершение цикла DMA. Вызывается соответствующее каналу прерывание (если было включено).

Отличие от режима «основной» состоит в том, что в режиме «авто-запрос» контроллер DMA позволит осуществить все N транзакций по одному запросу, в то время как в основном режиме по каждому запросу будет выполняться лишь  $2^R$  передач.

### **Режим «пинг-понг»**

Контроллер DMA выполняет цикл DMA, используя одну из первичных структур управляющих данных, а затем выполняет еще один цикл DMA, используя альтернативную структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает режим «недействительный» или «основной», или пока процессор не запретит работу канала.

На рисунке 9.3 показан пример функционирования контроллера DMA в режиме «пинг-понг». Пояснения к рисунку 9.3 представлены в виде таблицы 9.7.

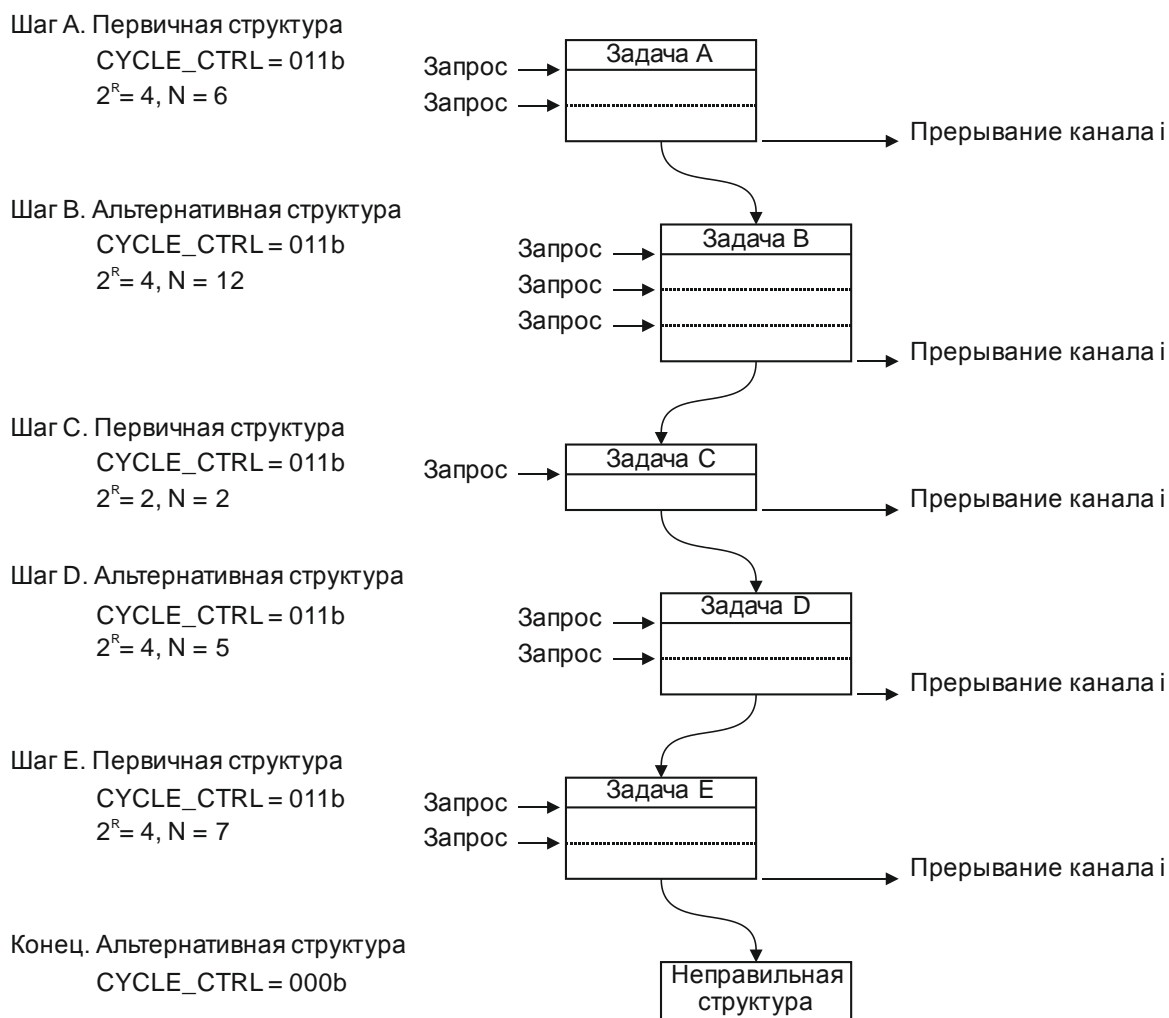


Рисунок 9.3 – Пример функционирования контроллера DMA в режиме «пинг-понг»

Таблица 9.7 – Пояснения к схеме на рисунке 9.3

Шаг	Действия процессора и контроллера DMA
А	<p>Процессор включает контроллер DMA и разрешает работу канала.</p> <p>В программе устанавливаются первичная структура управляющих данных для шага А и альтернативная структура управляющих данных для шага В. Это позволит контроллеру DMA переключиться к шагу В незамедлительно после выполнения шага А, при условии, что контроллер DMA не получит запрос на обработку от высокоприоритетного канала.</p> <p>Контроллер DMA получает запрос и выполняет четыре передачи DMA.</p> <p>Контроллер DMA выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер DMA продолжает цикл в ситуации отсутствия высокоприоритетных запросов.</p> <p>Контроллер DMA выполняет оставшиеся две передачи DMA.</p> <p>Контроллер DMA входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – 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 DMA не получит запрос на обработку от высокоприоритетного канала.</p> <p>После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг В.</p>

Окончание таблицы 9.7

Шаг	Действия процессора и контроллера DMA
В	<p>Контроллер DMA выполняет четыре передачи.                      Контроллер DMA выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер DMA продолжает цикл в ситуации отсутствия высокоприоритетных запросов.                      Контроллер DMA выполняет четыре передачи.                      Контроллер DMA выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер DMA продолжает цикл в ситуации отсутствия высокоприоритетных запросов.                      Контроллер DMA выполняет оставшиеся четыре передачи.                      Контроллер DMA входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – 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.</p>
С	<p>Контроллер DMA выполняет две передачи.                      Контроллер DMA входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После выполнения шага С процессор может установить первичные управляющие данные канала для шага E.                      После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D.</p>
D	<p>Контроллер DMA выполняет четыре передачи.                      Контроллер DMA выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер DMA продолжает цикл в ситуации отсутствия высокоприоритетных запросов.                      Контроллер DMA выполняет оставшуюся передачу.                      Контроллер DMA входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг E.</p>
E	<p>Контроллер DMA выполняет четыре передачи.                      Контроллер DMA выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер DMA продолжает цикл в ситуации отсутствия высокоприоритетных запросов.                      Контроллер DMA выполняет оставшиеся три передачи.                      Контроллер DMA входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов</p>

Если контроллер DMA получит новый запрос на обработку от данного канала, и этот запрос будет самым приоритетным, то контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные и по окончании шага D контроллер DMA установил поле CYCLE\_CTRL альтернативной управляющей структуры в состояние 000b, передачи DMA прекращаются.

**Работа с памятью в режиме «разборка-сборка»**

Алгоритм работы данного режима является оптимальным именно для работы с памятью, несмотря на это, его использование возможно для любого типа передачи данных: память – память, периферия – память, память – периферия, – с помощью как программных запросов, так и запросов от периферии.

В данном режиме контроллер DMA использует первичные управляющие данные для программирования альтернативных управляющих данных.

Контроллер DMA, получая начальный запрос на обработку, выполняет четыре передачи DMA, заполняя альтернативную структуру канала данными, доступными для первичной управляющей структуры. По окончании этих передач контроллер DMA входит в процедуру арбитража, и если более высокоприоритетных запросов не обнаружено, начинает цикл DMA, используя обновленные альтернативные управляющие данные, после – арбитраж, затем контроллер DMA выполняет еще четыре передачи DMA, вновь заполняя альтернативную структуру данными с помощью первичной структуры.

Контроллер DMA продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- передача с использованием альтернативной управляющей структуры будет выполнена в режиме цикла «основной»;

- контроллер считает «неправильную» структуру управляющих данных. После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем записи в поле CYCLE\_CTRL значения 000b.

Контроллер DMA устанавливает прерывание канала DMA<sub>i</sub> в этом режиме работы только тогда, когда последний цикл передач DMA выполняется с использованием режима «основной». Также необходимо помнить, что для режима «основной» авто-запросы не действуют.

В таблице 9.8 указаны константы, которые должны быть записаны пользователем в регистр CHANNEL\_C «разборка-сборка».

Таблица 9.8 – Конфигурация первичной структуры управляющих данных канала для работы с памятью в режиме «разборка-сборка»

CHANNEL_CFG																																																																			
<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="width: 20px; text-align: center;">31</td><td style="width: 20px; text-align: center;">30</td><td style="width: 20px; text-align: center;">29</td><td style="width: 20px; text-align: center;">28</td><td style="width: 20px; text-align: center;">27</td><td style="width: 20px; text-align: center;">26</td><td style="width: 20px; text-align: center;">25</td><td style="width: 20px; text-align: center;">24</td><td style="width: 20px; text-align: center;">23</td><td style="width: 20px; text-align: center;">22</td><td style="width: 20px; text-align: center;">21</td><td style="width: 20px; text-align: center;">20</td><td style="width: 20px; text-align: center;">19</td><td style="width: 20px; text-align: center;">18</td><td style="width: 20px; text-align: center;">17</td><td style="width: 20px; text-align: center;">16</td><td style="width: 20px; text-align: center;">15</td><td style="width: 20px; text-align: center;">14</td><td style="width: 20px; text-align: center;">13</td><td style="width: 20px; text-align: center;">12</td><td style="width: 20px; text-align: center;">11</td><td style="width: 20px; text-align: center;">10</td><td style="width: 20px; text-align: center;">9</td><td style="width: 20px; text-align: center;">8</td><td style="width: 20px; text-align: center;">7</td><td style="width: 20px; text-align: center;">6</td><td style="width: 20px; text-align: center;">5</td><td style="width: 20px; text-align: center;">4</td><td style="width: 20px; text-align: center;">3</td><td style="width: 20px; text-align: center;">2</td><td style="width: 20px; text-align: center;">1</td><td style="width: 20px; text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table> </div>				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	1	0	1	0	1	0	1	0	-	-	-	-	0	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	0	1	0	0
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																																				
1	0	1	0	1	0	1	0	-	-	-	-	0	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	0	1	0	0																																				
Поле	Биты	Конс- танта	Пояснение																																																																
DST_INC	31-30	10b	Контроллер DMA производит инкремент адреса с шагом в двойное слово																																																																
DST_SIZE	29-28	10b	Контроллер DMA осуществляет передачу двойным словом																																																																
SRC_INC	27-26	10b	Контроллер DMA производит инкремент адреса с шагом в двойное слово																																																																
SRC_SIZE	25-24	10b	Контроллер DMA осуществляет передачу двойными словами																																																																
DST_PROT_CTRL	23-21	–	Управление защитой шины при записи данных в приемник. Задается пользователем																																																																
SRC_PROT_CTRL	20-18	–	Управление защитой шины при чтении данных из источника. Задается пользователем																																																																
R_POWER	17-14	0010b	Контроллер DMA выполняет четыре передачи ( $2^R = 2^2 = 4$ )																																																																
N_MINUS_1	13-4	–	Настраивает контроллер на выполнение N передач DMA. Так как поле R_POWER задает значение 2, то необходимо задавать значение N, кратное 4. Число, равное N/4, это количество раз, которое нужно настраивать альтернативные управляющие данные. Задается пользователем																																																																
NEXT_USEBURST	3	0	Для данного режима бит должен быть сброшен																																																																
CYCLE_CTRL	2-0	100b	Контроллер DMA работает с памятью в режиме «разборка-сборка» с использованием первичных управляющих данных канала.																																																																

В указатель конца данных источника SRC\_DATA\_END\_PTR первичной структуры необходимо записать адрес конца области памяти, в которой последовательно расположено нужное количество наборов управляющих данных для программирования альтернативной структуры канала.

В указатель конца данных приемника DST\_DATA\_END\_PTR первичной структуры необходимо записать адрес конца альтернативной управляющей структуры используемого канала.

На рисунке 9.4 показан пример функционирования контроллера DMA в режиме «разборка-сборка». Пояснения к рисунку 9.4 приведены ниже («Инициализация» и «Функционирование»).



Рисунок 9.4 – Пример функционирования контроллера DMA в режиме «разборка-сборка»

### Инициализация

1 Первичная структура управляющих данных настраивается для работы с памятью в режиме «разборка-сборка» путем записи в CYCLE\_CTRL значения 100b. Так как управляющие данные канала состоят из четырех слов, R\_POWER = 0010b. Поскольку количество задач равно четырем, то N = 16, т. е. значение поля N\_MINUS\_1 = 00Fh.

2 Управляющие данные для шагов А, В, С, D располагаются в области ОЗУ. Адрес конца этой области заносится в регистр SRC\_DATA\_END\_PTR первичных управляющих данных. Пример размещения и заполнения управляющих данных для альтернативной структуры показан в таблице 9.9. Исходя из примера, в регистр SRC\_DATA\_END\_PTR первичной управляющей структуры необходимо занести значение 2000\_015Ch.

В регистр DST\_DATA\_END\_PTR первичной структуры необходимо занести адрес конца альтернативной структуры управляющих данных используемого канала. Например, при использовании канала 9 в регистр DST\_DATA\_END\_PTR необходимо занести значение XXXX\_X29Ch.

3 Включается контроллер DMA и разрешается работа канала путем установки соответствующего каналу бита в регистре ENSET и бита MASTEREN в регистре CFG. Также необходимо удостовериться, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре REQMASKSET).

Таблица 9.9 – Пример размещения управляющих данных для альтернативной структуры

Тип данных	Адрес ОЗУ	Регистр	Значение
Управляющие данные для задачи D	2000_015Ch	Не используется	XXXX_XXXXh
	2000_0158h	CHANNEL_CFG	CYCLE_CTRL = 001b, $2^R = 4$ , N = 4
	2000_0154h	DST_DATA_END_PTR	2000_DE00h
	2000_0150h	SRC_DATA_END_PTR	2000_D000h
Управляющие данные для задачи C	2000_016Ch	Не используется	XXXX_XXXXh
	2000_0168h	CHANNEL_CFG	CYCLE_CTRL = 101b, $2^R = 8$ , N = 5
	2000_0164h	DST_DATA_END_PTR	2000_CE00h
	2000_0160h	SRC_DATA_END_PTR	2000_C000h
Управляющие данные для задачи B	2000_017Ch	Не используется	XXXX_XXXXh
	2000_0178h	CHANNEL_CFG	CYCLE_CTRL = 101b, $2^R = 2$ , N = 8
	2000_0174h	DST_DATA_END_PTR	2000_BE00h
	2000_0170h	SRC_DATA_END_PTR	2000_B000h
Управляющие данные для задачи A	2000_018Ch	Не используется	XXXX_XXXXh
	2000_0188h	CHANNEL_CFG	CYCLE_CTRL = 101b, $2^R = 4$ , N = 3
	2000_0184h	DST_DATA_END_PTR	2000_AE00h
	2000_0180h	SRC_DATA_END_PTR	2000_A000h

### Функционирование

1 Первичная структура, копирование данных задачи A. По получении первого запроса на обслуживание контроллер DMA выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи A. Контроллер DMA генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Далее контроллер DMA выполняет задачу A с использованием альтернативных данных, по окончании генерирует авто-запрос для канала и проводит процедуру арбитража.

2 Первичная структура, копирование данных задачи B. По получении авто-запроса контроллер DMA выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи B. Контроллер DMA генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Далее контроллер DMA выполняет задачу B с использованием альтернативных данных, по окончании генерирует авто-запрос для канала и проводит процедуру арбитража.

3 Первичная структура, копирование данных задачи C. По получении авто-запроса на обслуживание контроллер DMA выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи C. Контроллер DMA генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Далее контроллер DMA выполняет задачу С с использованием альтернативных данных, по окончании генерирует авто-запрос для канала и проводит процедуру арбитража.

4 Первичная структура, копирование данных задачи D. По получении авто-запроса на обслуживание контроллер DMA выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи D. Контроллер DMA записывает в CYCLE\_CTRL первичных данных значение 000b для индикации о том, что эта структура управляющих данных является «неправильной». Далее контроллер DMA генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Далее контроллер DMA выполняет задачу D, используя тип цикла «основной». По завершении задачи генерирует прерывание канала DMA (если было включено) и входит в процедуру арбитража. Цикл работы с памятью в режиме «разборка-сборка» завершен.

#### **Работа с периферией в режиме «разборка-сборка»**

Алгоритм работы данного режима является оптимальным именно для работы с периферией, несмотря на это, его использование возможно для любого типа передачи данных: память – память, периферия – память, память – периферия, – с помощью как программных запросов, так и запросов от периферии.

В данном режиме контроллер DMA использует первичные управляющие данные для программирования альтернативных управляющих данных.

Контроллер DMA, получая начальный запрос на обработку, выполняет четыре передачи DMA, заполняя альтернативную структуру канала данными, доступными для первичной управляющей структуры. По окончании этих передач контроллер DMA без осуществления арбитража начинает цикл DMA, используя обновленные альтернативные управляющие данные, после – арбитраж, затем контроллер DMA выполняет еще четыре передачи DMA, вновь заполняя альтернативную структуру данными с помощью первичной структуры. Это единственный случай, при котором контроллер DMA не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

Контроллер DMA продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- передача с использованием альтернативной управляющей структуры будет выполнена в режиме цикла «основной»;

- контроллер DMA считает «неправильную» структуру управляющих данных. После исполнения DMA контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем записи в поле CYCLE\_CTRL значения 000b.

Контроллер DMA устанавливает прерывание канала DMA<sub>i</sub> в этом режиме работы только тогда, когда последний цикл передач DMA выполняется с использованием режима «основной» Также необходимо помнить, что для режима «основной» авто-запросы не действуют.

В таблице 9.10 указаны константы, которые должны быть записаны пользователем в регистр CHANNEL\_CFG первичной структуры управляющих данных канала для работы с периферией в режиме «разборка-сборка».



Таблица 9.10 – Конфигурация первичной структуры управляющих данных канала DMAi для работы с периферией в режиме «разборка-сборка»

CHANNEL_CFG																																			
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				
1	0	1	0	1	0	1	0	-	-	-	-	-	-	0	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	1	0
Поле	Биты	Конс-танта	Пояснение																																
DST_INC	31-30	10b	Контроллер DMA производит инкремент адреса с шагом в двойное слово																																
DST_SIZE	29-28	10b	Контроллер DMA осуществляет передачу двойным словом																																
SRC_INC	27-26	10b	Контроллер DMA производит инкремент адреса с шагом в двойное слово																																
SRC_SIZE	25-24	10b	Контроллер DMA осуществляет передачу двойными словами																																
DST_PROT_CTRL	23-21	–	Управление защитой шины при записи данных в приемник. Задается пользователем																																
SRC_PROT_CTRL	20-18	–	Управление защитой шины при чтении данных из источника. Задается пользователем																																
R_POWER	17-14	0010b	Контроллер DMA выполняет четыре передачи DMA ( $2^R = 2^2 = 4$ )																																
N_MINUS_1	13-4	–	Настраивает контроллер DMA на выполнение N передач DMA. Так как поле R_POWER задает значение, равное двум, то необходимо задавать значение N, кратное четырем. Число, равное N/4, это количество раз, которое нужно настраивать альтернативные управляющие данные. Задается пользователем																																
NEXT_USEBURST	3	0	Для данного режима бит должен быть сброшен																																
CYCLE_CTRL	2-0	110b	Контроллер работает с периферией в режиме «разборка-сборка» с использованием первичных управляющих данных канала DMAi																																

В указатель конца данных источника SRC\_DATA\_END\_PTR первичной структуры необходимо записать адрес конца области памяти, в которой последовательно расположено нужное количество наборов управляющих данных для программирования альтернативной структуры канала DMAi.

В указатель конца данных приемника DST\_DATA\_END\_PTR первичной структуры необходимо записать адрес конца альтернативной управляющей структуры используемого канала DMAi.

На рисунке 9.5 показан пример функционирования контроллера DMA в режиме «разборка-сборка». Пояснения к рисунку 9.5 приведены ниже («Инициализация» и «Функционирование»).

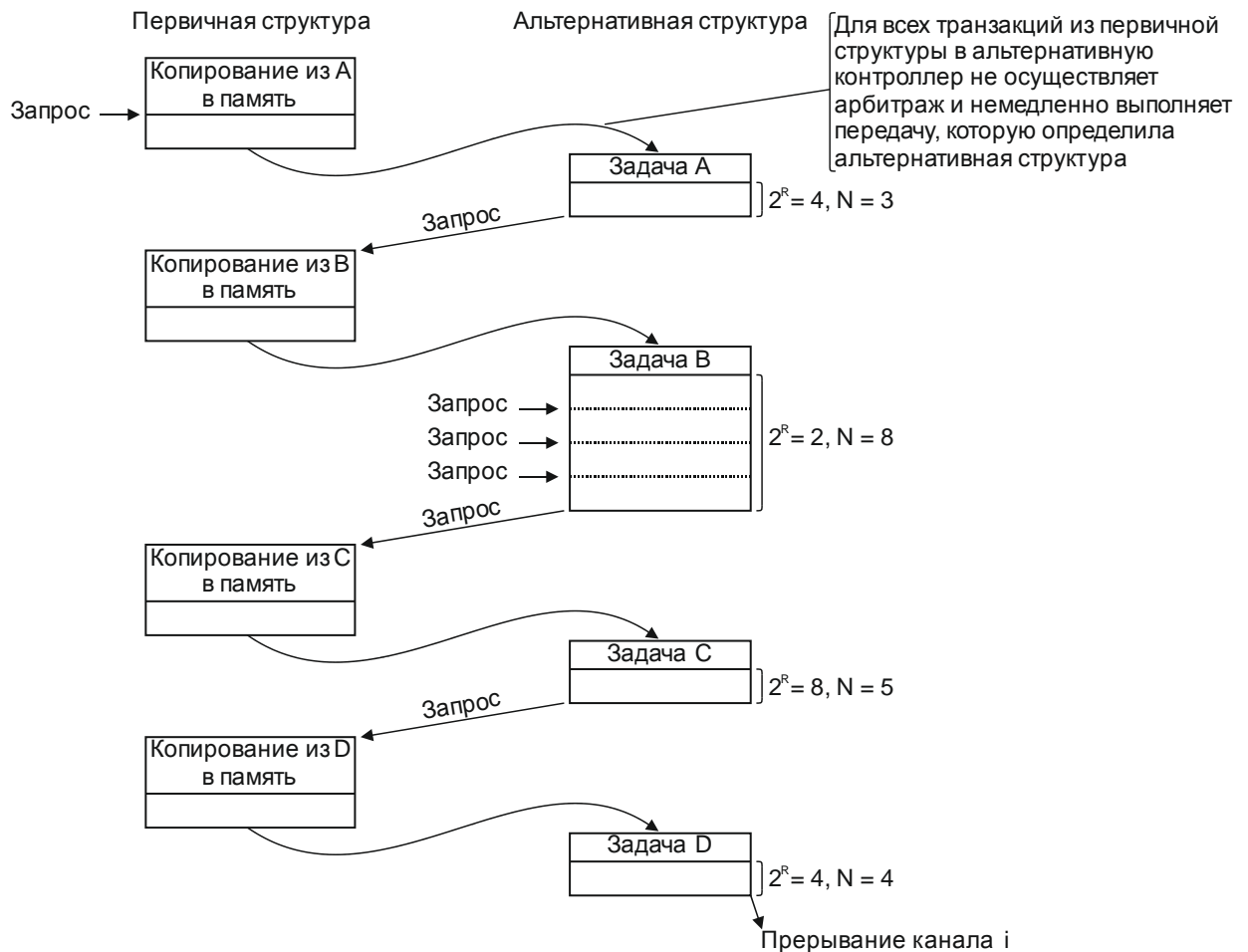


Рисунок 9.5 – Пример функционирования контроллера DMA в режиме работы с периферией «разборка-сборка»

### Инициализация

1 Первичная структура управляющих данных настраивается для работы с периферией в режиме «разборка-сборка» путем записи в CYCLE\_CTRL значения 110b. Поскольку управляющие данные канала состоят из четырех слов, R\_POWER = 0010b. Так как количество задач равно четырём, то N = 16, т. е. значение поля N\_MINUS\_1 = 00Fh.

2 Управляющие данные для шагов A, B, C, D располагаются в области ОЗУ. Адрес конца этой области заносится в регистр SRC\_DATA\_END\_PTR первичных управляющих данных. Пример размещения и заполнения управляющих данных для альтернативной структуры, показан в таблице 9.11. Исходя из примера, в регистр SRC\_DATA\_END\_PTR необходимо занести значение 2000\_015Ch.

В регистр DST\_DATA\_END\_PTR первичной структуры необходимо занести адрес конца альтернативной структуры управляющих данных используемого канала. Например, при использовании канала 9 в регистр DST\_DATA\_END\_PTR необходимо занести значение XXXX\_X29Ch.

3 Включается контроллер DMA и разрешается работа канала путем установки соответствующего каналу бита в регистре ENSET и бита MASTEREN в регистре CFG. Также необходимо удостовериться, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре REQMASKSET).

Таблица 9.11 – Пример размещения управляющих данных для альтернативной структуры

Тип данных	Адрес ОЗУ	Регистр	Значение
Управляющие данные для задачи D	2000_015Ch	Не используется	XXXX_XXXXh
	2000_0158h	CHANNEL_CFG	CYCLE_CTRL = 001b, $2^R = 4$ , N = 4
	2000_0154h	DST_DATA_END_PTR	2000_DE00h
	2000_0150h	SRC_DATA_END_PTR	2000_D000h
Управляющие данные для задачи C	2000_016Ch	Не используется	XXXX_XXXXh
	2000_0168h	CHANNEL_CFG	CYCLE_CTRL = 111b, $2^R = 8$ , N = 5
	2000_0164h	DST_DATA_END_PTR	2000_CE00h
Управляющие данные для задачи B	2000_0160h	SRC_DATA_END_PTR	2000_C000h
	2000_017Ch	Не используется	XXXX_XXXXh
	2000_0178h	CHANNEL_CFG	CYCLE_CTRL = 111b, $2^R = 2$ , N = 8
Управляющие данные для задачи A	2000_0174h	DST_DATA_END_PTR	2000_BE00h
	2000_0170h	SRC_DATA_END_PTR	2000_B000h
	2000_018Ch	Не используется	XXXX_XXXXh
Управляющие данные для задачи A	2000_0188h	CHANNEL_CFG	CYCLE_CTRL = 111b, $2^R = 4$ , N = 3
	2000_0184h	DST_DATA_END_PTR	2000_AE00h
	2000_0180h	SRC_DATA_END_PTR	2000_A000h

### Функционирование

1 Первичная структура, копирование данных для задачи A. По получении запроса на обслуживание контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи A.

Далее контроллер сразу же выполняет задачу A и по окончании проводит процедуру арбитража. После выставления нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

2 Первичная структура, копирование данных для задачи B. Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи B.

Далее контроллер выполняет задачу B. Для завершения задачи необходимо три запроса (программных или от периферии). По окончании контроллер проводит процедуру арбитража. После выставления нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

3 Первичная структура, копирование данных для задачи C. Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи C.

Далее контроллер выполняет задачу C и по окончании проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

4 Первичная структура, копирование данных для задачи D. Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D. Контроллер записывает в CYCLE\_CTRL первичных данных значение 000b для индикации о том, что эта структура управляющих данных является «неправильной». Далее контроллер выполняет задачу D, используя основной цикл DMA, входит в прерывание канала DMA<sub>i</sub> (если включено) и запускает процедуру арбитража. Цикл работы с периферией в режиме «разборка-сборка» завершен.

## 9.5 Циклический режим

По умолчанию контроллер DMA выполняет циклы однократно, т. е. чтобы выполнить исполненный цикл еще раз необходимо обновить поля N\_MINUS\_1 и CYCLE\_CTRL, а также включить соответствующий канал путём записи в регистр ENSET.

Однако существует возможность активировать циклический режим для основного цикла или авто-запроса. В этом режиме поле CYCLE\_CTRL и бит разрешения работы канала в регистре ENSET не будут сбрасываться по окончании цикла. Также значение N\_MINUS\_1 будет сохранено перед первой передачей и затем восстановлено, когда текущий цикл будет завершен. Таким образом, можно осуществлять циклические передачи, например, для генерации сигналов сложной формы и других задач, не задействуя при этом прерывания DMA.

Активация режима производится установкой соответствующего бита в регистре CIRCULARSET. Сброс режима – записью единицы в CIRCULARCLR.

## 9.6 Индикация ошибок

Контроллер DMA может отключить i-й канал в следующих случаях:

- при завершении цикла DMA;
- при чтении режима канала «недействительный»;
- при появлении ошибки на шине АНВ.

Как только контроллер DMA получает сообщение об ошибке по шине АНВ, он отключает канал, в котором обнаружена ошибка и устанавливает флаг VAL в регистре ERRCLR. Для того чтобы определить канал контроллера DMA, в котором произошла ошибка, программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно вызывали прерывания, т. е. завершали работу и отключались.

Алгоритм определения канала контроллера DMA с ошибкой:

- необходимо прочитать регистр ENSET с целью создания текущего списка отключенных каналов контроллера DMA;
- процессор должен сравнить список выключенных каналов контроллера DMA, полученный в результате чтения регистра ENSET, с данными о каналах, которые недавно вызывали прерывания. Канал контроллера DMA, который отключился и по которому отсутствуют данные о вызове прерывания, является каналом, связанным с ошибкой.

В контроллере DMA присутствует возможность использования режимов защиты шины АНВ: при записи в приемник, при чтении из источника и при обращении к структурам управляющих данных каналов. Защита шины в каждой из ситуаций настраивается индивидуально. Доступными режимами защиты являются: кэширование, буферизация, привилегированный доступ.

Защита шины при записи в приемник настраивается полем DST\_PROT\_CTRL в ячейке CHANNEL\_CFG структуры управляющих данных канала контроллера DMA.

Защита шины при чтении из источника настраивается полем SRC\_PROT\_CTRL в ячейке CHANNEL\_CFG структуры управляющих данных канала контроллера DMA.

Защита шины при обращении контроллера DMA к структурам управляющих данных каналов DMA<sub>i</sub> настраивается полем CHPROT в регистре CFG.

## 10 Система прерываний

Таблица прерываний представляет собой перечень адресов, соответствующих определенным обработчикам прерываний, см. таблицу 10.1.

Таблица 10.1 – Таблица прерываний

Номер вектора	Смещение	Обозначение	Описание
–	0000h	SP	Вершина стека
–	0004h	Reset	Сброс
–	0008h	NMI#	Немаскируемое прерывание NMI#
–	000Ch	HardFault	Любой отказ, если соответствующий обработчик не может быть запущен
–	0010h	MemManage	Прерывание по отказу системы управления памятью
–	0014h	BusFault	Прерывание по отказу шины АНВ
–	0018h	UsageFault	Прерывание по ошибке программы
–	001Ch-0028h	-	Зарезервировано
–	002Ch	SVCall	Обработка прерываний, вызванных инструкцией SVC
–	0030h	DebugMonitor	Прерывание монитора отладки
–	0034h	-	Зарезервировано
–	0038h	PendSV	Прерывание системного уровня. В приложении используется вызов «Супервизор», если этот запрос обслуживается базовой операционной системой
–	003Ch	SysTick	Прерывание системного уровня. Прерывание вызывается таймером SysTick
0	0040h	WDT	Прерывание блока сторожевого таймера
1	0044h	RCU	Прерывание блока RCU
2	0048h	RTC	Прерывание блока часов реального времени
3	004Ch	MFLASH	Прерывание контроллера основной Flash-памяти
4	0050h	BFLASH	Прерывание контроллера загрузочной Flash-памяти
5	0054h	GPIOA	Прерывание порта А
6	0058h	GPIOB	Прерывание порта В
7	005Ch	GPIOC	Прерывание порта С
8	0060h	GPIOD	Прерывание порта D
9	0064h	GPIOE	Прерывание порта E
10	0068h	GPIOF	Прерывание порта F
11	006Ch	GPIOG	Прерывание порта G
12	0070h	GPIOH	Прерывание порта H
13	0074h	GPIOJ	Прерывание порта J
140	0078h	GPIOK	Прерывание порта K
15	007Ch	GPIOI	Прерывание порта L
16	0080h	GIOM	Прерывание порта M

Продолжение таблицы 10.1

Номер вектора	Смещение	Обозначение	Описание
17	0084h	DMA_CH0	Прерывания контроллера DMA по каналам
18	0088h	DMA_CH1	
19	008Ch	DMA_CH2	
20	0090h	DMA_CH3	
21	0094h	DMA_CH4	
22	0098h	DMA_CH5	
23	009Ch	DMA_CH6	
24	00A0h	DMA_CH7	
25	00A4h	DMA_CH8	
26	00A8h	DMA_CH9	
27	00ACh	DMA_CH10	
28	00B0h	DMA_CH11	
29	00B4h	DMA_CH12	
30	00B8h	DMA_CH13	
31	00BCh	DMA_CH14	
32	00C0h	DMA_CH15	
33	00C4h	DMA_CH16	
34	00C8h	DMA_CH17	
35	00CCh	DMA_CH18	
36	00D0h	DMA_CH19	
37	00D4h	DMA_CH20	
38	00D8h	DMA_CH21	
39	00DCh	DMA_CH22	
40	00E0h	DMA_CH23	
41	00E4h	DMA_CH24	
42	00E8h	DMA_CH25	
43	00ECh	DMA_CH26	
44	00F0h	DMA_CH27	
45	00F4h	DMA_CH28	
46	00F8h	DMA_CH29	
47	00FCh	DMA_CH30	
48	0100h	DMA_CH31	
49	0104h	TMR0	Прерывание таймера 0
50	0108h	TMR1	Прерывание таймера 1
51	010Ch	TMR2	Прерывание таймера 2
52	0110h	TMR3	Прерывание таймера 3
53	0114h	ETMR0	Прерывание расширенного таймера 0
54	0118h	ETMR1	Прерывание расширенного таймера 1
55	011Ch	ETMR2	Прерывание расширенного таймера 2
56	0120h	ETMR3	Прерывание расширенного таймера 3
57	0124h	UART0_TD	Прерывание UART0 по окончанию передачи
58	0128h	UART0_MS	Прерывание UART0 по состоянию модема
59	012Ch	UART0_RX	Прерывание UART0 по заполнению FIFO
60	0130h	UART0_TX	Прерывание UART0 по опустошению FIFO
61	0134h	UART0_RT	Прерывание UART0 по таймауту приемника
62	0138h	UART0_E	Прерывание UART0 по ошибке
63	013Ch	UART0	Общее прерывание блока UART0

Продолжение таблицы 10.1

Номер вектора	Смещение	Обозначение	Описание
64	0140h	UART1_TD	Прерывание UART1 по окончанию передачи
65	0144h	UART1_MS	Прерывание UART1 по состоянию модема
66	0148h	UART1_RX	Прерывание UART1 по заполнению FIFO
67	014Ch	UART1_TX	Прерывание UART1 по опустошению FIFO
68	0150h	UART1_RT	Прерывание UART1 по таймауту приемника
69	0154h	UART1_E	Прерывание UART1 по ошибке
70	0158h	UART1	Общее прерывание блока UART1
71	015Ch	UART2_TD	Прерывание UART2 по окончанию передачи
72	0160h	UART2_MS	Прерывание UART2 по состоянию модема
73	0164h	UART2_RX	Прерывание UART2 по заполнению FIFO
74	0168h	UART2_TX	Прерывание UART2 по опустошению FIFO
75	016Ch	UART2_RT	Прерывание UART2 по таймауту приемника
76	0170h	UART2_E	Прерывание UART2 по ошибке
77	0174h	UART2	Общее прерывание блока UART2
78	0178h	UART3_TD	Прерывание UART3 по окончанию передачи
79	017Ch	UART3_MS	Прерывание UART3 по состоянию модема
80	0180h	UART3_RX	Прерывание UART3 по заполнению FIFO
81	0184h	UART3_TX	Прерывание UART3 по опустошению FIFO
82	0188h	UART3_RT	Прерывание UART3 по таймауту приемника
83	018Ch	UART3_E	Прерывание UART3 по ошибке
84	0190h	UART3	Общее прерывание блока UART 3
85	0194h	TUART0_RX	Прерывание TUART0 по приему
86	0198h	TUART0_TX	Прерывание TUART0 по передаче
87	019Ch	TUART0_RO	Прерывание TUART0 по переполнению буфера приемника
88	01A0h	TUART0_TO	Прерывание TUART0 по переполнению буфера передатчика
89	01A4h	TUART0	Общее прерывание блока TUART0
90	01A8h	TUART1_RX	Прерывание TUART0 по приему
91	01ACh	TUART1_TX	Прерывание TUART0 по передаче
92	01B0h	TUART1_RO	Прерывание TUART0 по переполнению буфера приемника
93	01B4h	TUART1_TO	Прерывание TUART0 по переполнению буфера передатчика
94	01B8h	TUART1	Общее прерывание блока TUART0
95	01BCh	SPI0	Общее прерывание контроллера SPI0
96	01C0h	SPI0_RX	Прерывание SPI0 по заполнению FIFO
97	01C4h	SPI0_TX	Прерывание SPI0 по опустошению FIFO
98	01C8h	SPI0_RO	Прерывание SPI0 по переполнению буфера приемника
99	01CCh	SPI0_RT	Прерывание SPI0 по таймауту приемника
100	01D0h	SPI1	Общее прерывание контроллера SPI1
101	01D4h	SPI1_RX	Прерывание SPI1 по заполнению FIFO
102	01D8h	SPI1_TX	Прерывание SPI1 по опустошению FIFO
103	01DCh	SPI1_RO	Прерывание SPI1 по переполнению буфера приемника
104	01E0h	SPI1_RT	Прерывание SPI1 по таймауту приемника
105	01E4h	SPI2	Общее прерывание контроллера SPI2

Продолжение таблицы 10.1

Номер вектора	Смещение	Обозначение	Описание
106	01E8h	SPI2_RX	Прерывание SPI2 по заполнению FIFO
107	01ECh	SPI2_TX	Прерывание SPI2 по опустошению FIFO
108	01F0h	SPI2_RO	Прерывание SPI2 по переполнению буфера приемника
109	01F4h	SPI2_RT	Прерывание SPI2 по таймауту приемника
110	01F8h	SPI3	Общее прерывание контроллера SPI 3
111	01FCh	SPI3_RX	Прерывание SPI3 по заполнению FIFO
112	0200h	SPI3_TX	Прерывание SPI3 по опустошению FIFO
113	0204h	SPI3_RO	Прерывание SPI3 по переполнению буфера приемника
114	0208h	SPI3_RT	Прерывание SPI3 по таймауту приемника
115	020Ch	I2C0	Прерывание контроллера I2C0
116	0210h	I2C1	Прерывание контроллера I2C1
117	0214h	SPWR0	Прерывание контроллера SpaceWire0
118	0218h	SPWR1	Прерывание контроллера SpaceWire1
119	021Ch	MILSTD0	Прерывание контроллера МПИ0
120	0220h	MILSTD1	Прерывание контроллера МПИ1
121	0224h	ECAP0	Прерывание блока захвата 0
122	0228h	ECAP1	Прерывание блока захвата 1
123	022Ch	ECAP2	Прерывание блока захвата 2
124	0230h	ECAP3	Прерывание блока захвата 3
125	0234h	ECAP4	Прерывание блока захвата 4
126	0238h	ECAP5	Прерывание блока захвата 5
127	023Ch	PWM0	Общее прерывание блока ШИМ0
128	0240h	PWM0_HD	Прерывание схемы удержания блока ШИМ0
129	0244h	PWM0_TZ	Прерывание детектора аварий блока ШИМ0
130	0248h	PWM1	Общее прерывание блока ШИМ1
131	024Ch	PWM1_HD	Прерывание схемы удержания блока ШИМ1
132	0250h	PWM1_TZ	Прерывание детектора аварий блока ШИМ1
133	0254h	PWM2	Общее прерывание блока ШИМ2
134	0258h	PWM2_HD	Прерывание схемы удержания блока ШИМ2
135	025Ch	PWM2_TZ	Прерывание детектора аварий блока ШИМ2
136	0260h	PWM3	Общее прерывание блока ШИМ3
137	0264h	PWM3_HD	Прерывание схемы удержания блока ШИМ3
138	0268h	PWM3_TZ	Прерывание детектора аварий блока ШИМ3
139	026Ch	PWM4	Общее прерывание блока ШИМ4
140	0270h	PWM4_HD	Прерывание схемы удержания блока ШИМ4
141	0274h	PWM4_TZ	Прерывание детектора аварий блока ШИМ4
142	0278h	PWM5	Общее прерывание блока ШИМ5
143	027Ch	PWM5_HD	Прерывание схемы удержания блока ШИМ5
144	0280h	PWM5_TZ	Прерывание детектора аварий блока ШИМ5
145	0284h	PWM6	Общее прерывание блока ШИМ6
146	0288h	PWM6_HD	Прерывание схемы удержания блока ШИМ6
147	028Ch	PWM6_TZ	Прерывание детектора аварий блока ШИМ6
148	0290h	PWM7	Общее прерывание блока ШИМ7
149	0294h	PWM7_HD	Прерывание схемы удержания блока ШИМ7
150	0298h	PWM7_TZ	Прерывание детектора аварий блока ШИМ7



Окончание таблицы 10.1

Номер вектора	Смещение	Обозначение	Описание
151	029Ch	PWM8	Общее прерывание блока ШИМ8
152	02A0h	PWM8_HD	Прерывание схемы удержания блока ШИМ8
153	02A4h	PWM8_TZ	Прерывание детектора аварий блока ШИМ8
154	02A8h	PWM9	Общее прерывание блока ШИМ9
155	02ACh	PWM9_HD	Прерывание схемы удержания блока ШИМ9
156	02B0h	PWM9_TZ	Прерывание детектора аварий блока ШИМ9
157	02B4h	QEP0	Прерывание квадратурного декодера 0
158	02B8h	QEP1	Прерывание квадратурного декодера 1
159	02BCh	QEP2	Прерывание квадратурного декодера 2
160	02C0h	QEP3	Прерывание квадратурного декодера 3
161	02C4h	ADC_SEQ0	Прерывание секвенсора 0 блока АЦП
162	02C8h	ADC_SEQ1	Прерывание секвенсора 1 блока АЦП
163	02CCh	ADC_SEQ2	Прерывание секвенсора 2 блока АЦП
164	02D0h	ADC_SEQ3	Прерывание секвенсора 3 блока АЦП
165	02D4h	ADC_SEQ4	Прерывание секвенсора 4 блока АЦП
166	02D8h	ADC_SEQ5	Прерывание секвенсора 5 блока АЦП
167	02DCh	ADC_SEQ6	Прерывание секвенсора 6 блока АЦП
168	02E0h	ADC_SEQ7	Прерывание секвенсора 7 блока АЦП
169	02E4h	ADC_DC	Прерывание компараторов блока АЦП
170	02E8h	ETH	Прерывание контроллера Ethernet
171	02ECh	CAN0	Прерывания контроллера CAN
172	02F0h	CAN1	
173	02F4h	CAN2	
174	02F8h	CAN3	
175	02FCh	CAN4	
176	0300h	CAN5	
177	0304h	CAN6	
178	0308h	CAN7	
179	030Ch	CAN8	
180	0310h	CAN9	
181	0314h	CAN10	
182	0318h	CAN11	
183	031Ch	CAN12	
184	0320h	CAN13	
185	0324h	CAN14	
186	0328h	CAN15	
187	032Ch	FPU	Прерывание исключений блока FPU
188	0330h	TMU	Прерывание блока тригонометрических вычислений
189	0334h	LAU	Прерывание блока программируемых логических ячеек
190	0338h	SDFM	Прерывание блока программируемых логических ячеек
191	033Ch	OWI0	Прерывание контроллера однопроводного интерфейса 0
192	0340h	OWI1	Прерывание контроллера однопроводного интерфейса 1

Первоначальным адресом начала таблицы прерываний является 0000\_0000h.

Таблица векторов может быть размещена по другому адресу в памяти программ или в ОЗУ. В случае размещения таблицы прерываний в области ОЗУ появляется возможность изменять обработчики прерываний в процессе выполнения программы. Положение таблицы векторов в памяти определяется регистром VTOR, см. таблицу 10.2.

Таблица 10.2 – Регистр смещения таблицы векторов

VTOR		E000_ED08h	Сброс: 0000_0000h
Поле	Биты	Описание	
TBLOFF	31-8	Биты 31-8 адреса таблицы векторов	
0	7-0	Младший байт адреса таблицы векторов. Всегда должен быть равен 00h.	

Обработчики прерываний можно динамически менять, но при этом обязательно следует располагать следующие элементы:

- начальное значение основного указателя стека;
- вектор сброса Reset;
- вектор NMI;
- вектор исключения HardFault.

Остальные прерывания не могут генерироваться, пока не будут разрешены.

### 10.1 Контроллер прерываний NVIC

Контроллер прерываний NVIC обеспечивает:

- программное задание уровня приоритета независимо для каждого прерывания в диапазоне от 0 до 3 (прерывание с уровнем 0 имеет наивысший приоритет);
- генерирование сигнала прерывания по фронту и по уровню сигнала;
- динамическое изменение приоритета прерываний;
- разделение по группам с одинаковым приоритетом и по подгруппам внутри одной группы;
- передача управления из одного обработчика в другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние при входе в обработчик прерывания и восстанавливает свое состояние после завершения обработки прерывания, т. е. без необходимости программирования этих операций.

### 10.2 Обработка прерываний по уровню и по фронту

Контроллер прерываний NVIC поддерживает прерывания, как по фронту так и по уровню. Прерывание по фронту – импульсное прерывание, которое может иметь длительность большую или равную длительности такта системной частоты.

Прерывание по уровню возникает до тех пор, пока устройством удерживается заданный уровень сигнала. Если прерывание по уровню не было снято до завершения работы обработчика прерываний, то контроллер NVIC вновь начинает его обработку.

В случае прихода импульсного прерывания от любого источника в момент обработки предыдущего, в контроллере NVIC устанавливается флаг, сигнализирующий о приходе нового прерывания, которое будет обработано после завершения обработки текущего прерывания. В случае, если контроллер NVIC находится в состоянии ожидания и приходит импульсное прерывание от того же источника, обработка выполнится только один раз.

Для управления прерываниями используются пять групп регистров  $ISERi$ ,  $ICERi$ ,  $ISPRi$ ,  $ICPRi$  и  $IABRi$ , где индекс  $i = 0, 1, 2$ . Подробное описание – в приложении В.

## 11 Порты ввода-вывода

В состав микроконтроллера входят двенадцать 16-разрядных портов ввода-вывода: А, В, С, D, E, F, G, H, J, K, L, М. Структуры портов и функционирования – идентичны.

Каждый цифровой вывод порта микроконтроллера может использоваться как двунаправленный вывод общего назначения (режим GPIO). Помимо этого, все выходы имеют альтернативную функцию (или функции).

По умолчанию порты находятся в сбросе и не тактируются. Активировать порты можно с помощью соответствующих бит регистров HCLKCFG, HRSTCFG блока RCU.

### 11.1 Функционирование порта

Полученные данные сохраняются в регистре DATA порта. Данные для передачи записываются в регистр DATAOUT порта. Существует возможность модификации состояния регистра DATAOUT путем записи единиц в регистр DATAOUTSET для установки соответствующих бит, в регистр DATAOUTCLR – для сброса бит, в регистр DATAOUTTGL – для инверсии бит.

На рисунке 11.1 приведена структурная схема вывода цифрового порта микроконтроллера. Схемы всех выводов идентичны.

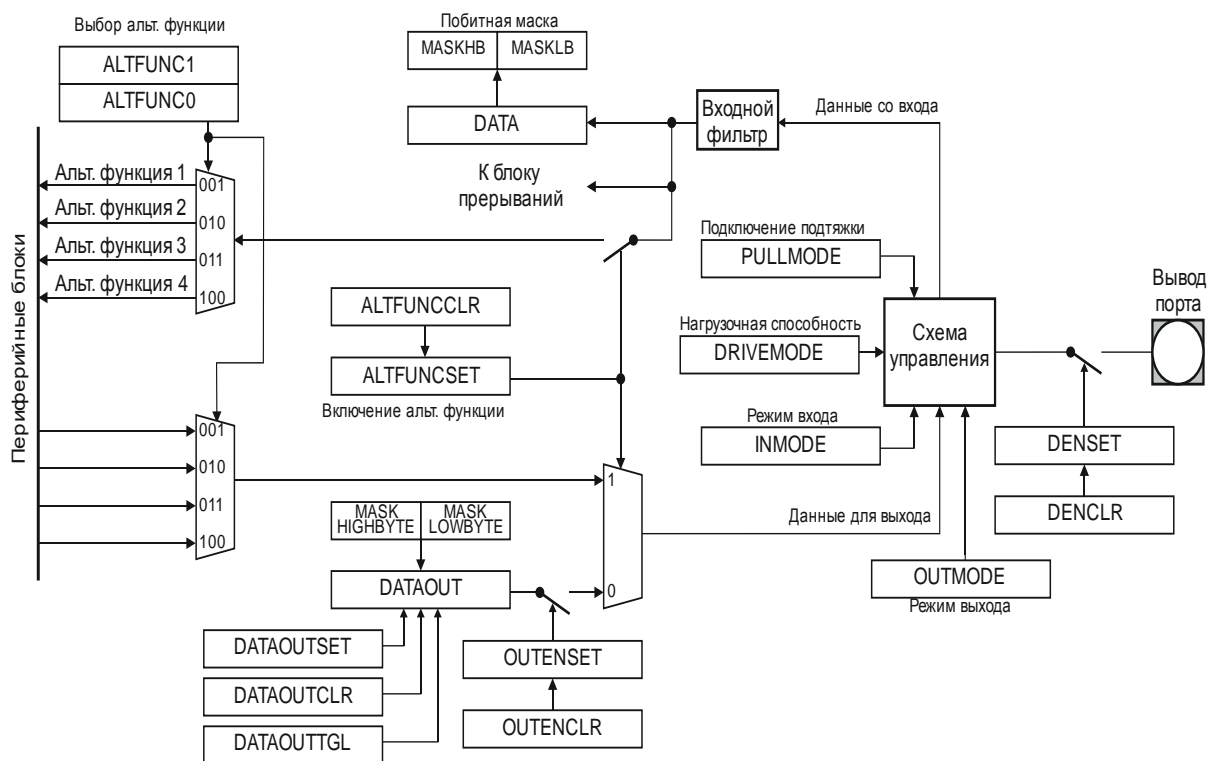


Рисунок 11.1 – Структурная схема вывода цифрового порта микроконтроллера

Схема состоит из двунаправленной площадки вывода, фильтра входных сигналов, мультиплексора выбора номера альтернативной функции, мультиплексора выбора режима работы (режим GPIO либо режим альтернативной функции).

Для каждого вывода задается режим работы, номер альтернативной функции, нагрузочная способность и быстродействие вывода, режим подтяжки, а также производится настройка порта на работу в режиме с открытым стоком/исток. Для работы с периферийными блоками включается режим альтернативной функции. Входной сигнал может подаваться для дальнейшей обработки как напрямую (асинхронный вход), так и проходить обработку через фильтр.

После сброса все выводы конфигурируются как выводы общего назначения (режим GPIO) и находятся в третьем состоянии. Перед тем как взаимодействовать с выводом, как цифровым входом, выходом или альтернативной функцией, необходимо разрешить цифровую работу вывода, записав единицы в соответствующие разряды регистра DENSET. Для сброса установленных бит следует записать единицы в регистр DENCLR.

Выбор нагрузочной способности и быстродействия вывода определяется полями регистра DRIVEMODE, а режим подтяжки конфигурируется регистром PULLMODE.

Разрешение работы выходных каскадов определяется состоянием бит регистра OUTENSET (для сброса установленных бит следует записать единицы в регистр OUTENCLR), а их режимы («push-pull», открытый сток/исток) состоянием полей в регистре OUTMODE.

Режим работы входной цепи настраивается с помощью регистра INMODE.

## 11.2 Режим альтернативных функций

Для перевода желаемого вывода порта в режим альтернативной функции необходимо установить соответствующий бит в регистре ALTFUNCSET порта. Для отключения альтернативной функции нужно записать единицу в соответствующий бит регистра ALTFUNCCLR. Выбор номера альтернативной функции для младшей части порта (выводы 0-7) осуществляется с помощью регистра ALTFUNCNUM0, а для старшей (выводы 8-15) – ALTFUNCNUM1. Каждому выводу соответствуют четыре бита регистра.

Входы и выходы периферийных блоков в процессе работы коммутируются с выводами микроконтроллера при условии, что для этих выводов включен режим альтернативной функции. В связи с этим периферийный блок может передавать информацию на несколько выводов одновременно. В то же время прием информации может осуществляться только с одного вывода, во избежание конфликтов уровней сигналов (для этого дополнительно предусмотрена система приоритета альтернативных функций). Количество выводов, сигналы с которых могут быть переданы на периферийный блок, для каждого блока различно.

Распределение приоритетов для входных функций происходит следующим образом: наивысший приоритет имеют функции с наименьшим номером, наименьший приоритет – функции с наибольшим номером. Но при этом, если на разных выводах встречаются идентичные функции с одинаковыми номерами, то их приоритеты распределяются по букве порта и номеру пина: чем номер пина меньше, а буква порта ближе к началу алфавита, тем приоритет больше.

Указанные правила программирования выводов для приема данных распространяются на все периферийные блоки.

Для примера, рассмотрим схему взаимодействия входа RX блока CAN0 и выводов микроконтроллера, запрограммированных на прием внешних сигналов.

Вход RX блока CAN0 может принимать информацию с одного из выводов C13, D8, H13 или M13.

Приоритеты для CAN0\_RX распределены следующим образом (от наивысшего к наименьшему): первая функция на C13, вторая функция на D8, третья функция на H13, четвертая функция на M13.

Если для выводов C13, D8, H13 и M13 одновременно выбрана функция CAN0\_RX, то на вход блока CAN будет скоммутирован вывод, альтернативная функция которого имеет более высокий приоритет.

## 11.3 Входные фильтры

Ко всем площадкам выводов подключены входные фильтры. На рисунке 11.2 показана структурная схема фильтра вывода порта.

Входной сигнал с вывода порта может приниматься как напрямую (асинхронный режим), так и пересинхронизироваться (синхронизироваться с тактовой частотой работы микроконтроллера). Управление осуществляется регистрами SYNCSET/SYNCLCR.

Дополнительно есть возможность включения накопления трех или шести отсчетов входного сигнала для помехоустойчивости вывода. Если результаты всех отсчетов совпадают, сигнал передается дальше по схеме, в противном случае состояние сигнала не меняется. Временные интервалы между отсчетами задаются в количестве тактов системной частоты посредством регистра QUALSAMPLE. Временной интервал задается один для всех выводов порта.

Включение фильтра и задание режима его работы осуществляется посредством регистров QUALSET/QUALCLR и QUALMODESET/QUALMODECLR соответственно.

Одновременно оба режима активными быть не могут – при установленных единицах в одних и тех же разрядах SYNCSET и QUALSET сигнал будет проходить напрямую с входа фильтра на его выход.

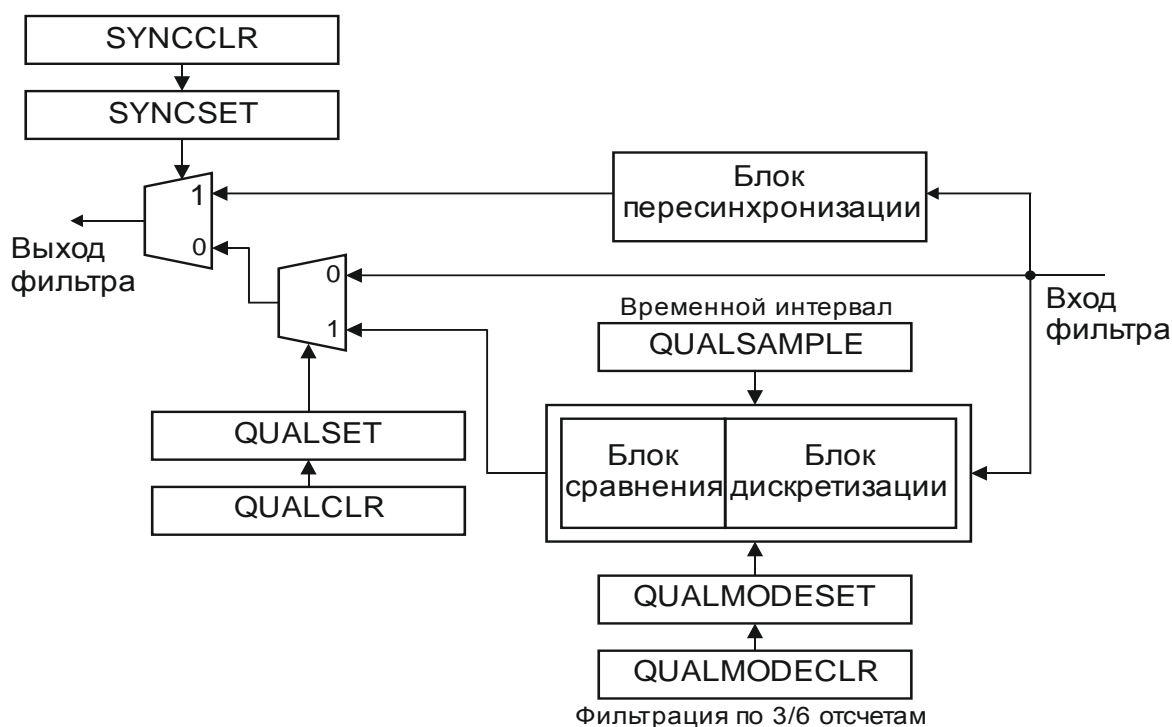


Рисунок 11.2 – Структурная схема фильтра вывода порта

#### 11.4 Прерывания

Каждый из выводов способен генерировать прерывание. На рисунке 11.3 показана структурная схема блока генерации прерываний.

Схема вывода позволяет также осуществлять гибкое управление прерываниями и задавать, по какому аппаратному событию генерировать прерывание (по какому фронту или уровню). При возникновении прерывания в регистре INTSTATUS устанавливается соответствующий флаг, и выставляется прерывание в контроллере прерываний NVIC. Прерывание может быть сгенерировано программно записью единицы в соответствующий бит регистра INTSTATUS.

Прерывание может быть сброшено программно записью единицы в соответствующий бит регистра INTCLEAR. Для разрешения прерывания вывода порта следует записать единицу в соответствующий выводу бит регистра INTENSET, а для запрета прерывания – единицу в бит регистра INTENCLR.

Для задания типа события (уровень или фронт), по которому генерируется прерывание, используется регистр INTTYPESET, для задания полярности

(низкий/высокий уровень или положительный/отрицательный фронт) используется INTPOLSET, а для сброса настроек – INTTYPECLR и INTPOLCLR соответственно.

Существует возможность организации прерывания по обоим фронтам – сначала необходимо задать тип прерывания – фронт (запись в INTTYPESET), а затем записать соответствующие единицы в INTEDGESET. В этом режиме состояние регистра полярности INTPOLSET игнорируется. Отключить режим генерации прерывания по обоим фронтам можно записью в INTEDGECLR, в таком случае для генерации в дальнейшем будет использована текущая настройка полярности (регистр INTPOLSET).

В режиме прерывания по уровню состояние регистра INTEDGESET не влияет на их генерацию.

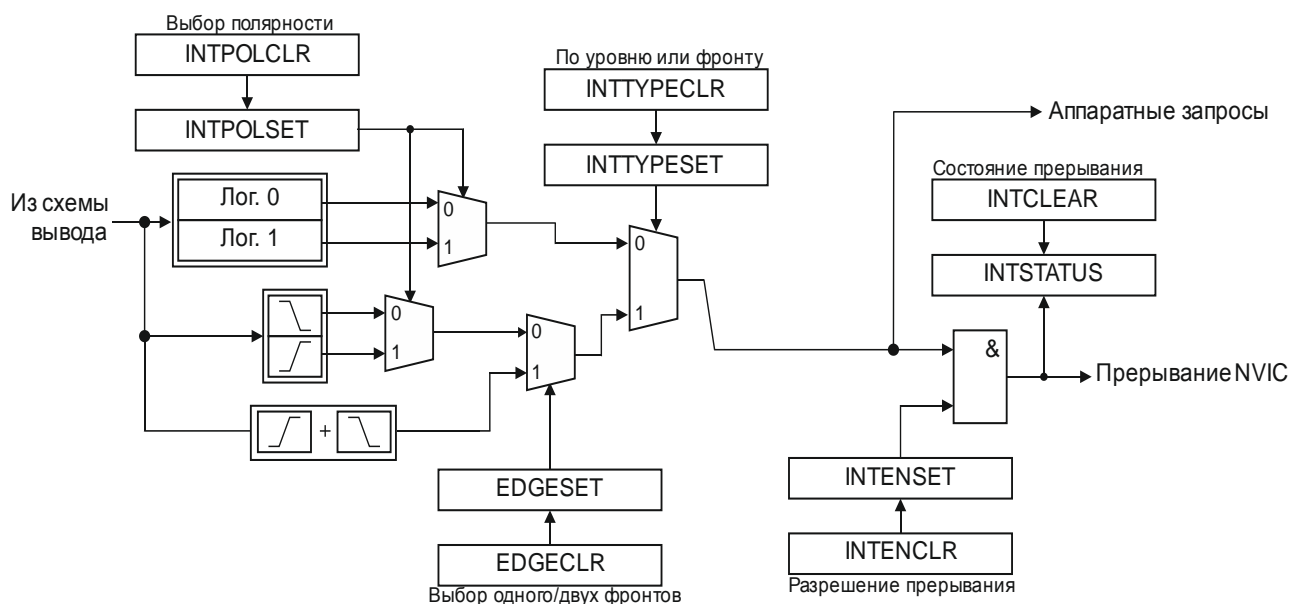


Рисунок 11.3 – Структурная схема блока генерации прерываний

## 11.5 Генерация аппаратных запросов

Кроме генерации прерываний, порты также имеют возможность генерации запросов к другой периферии. Генерация запроса происходит по условиям возникновения прерывания на выбранном выводе, при этом не важно, маскировано само прерывание или нет (состояние INTENSET), влияние оказывают только настройки генерации прерывания (INTTYPESET, INTPOLSET, INTEDGESET).

Для того чтобы в соответствии с настройками прерывания генерировался BREQ запрос к контроллеру DMA, необходимо осуществить запись единиц в DMAREQSET. Отключение генерации запросов к контроллеру DMA осуществляется через DMAREQCLR.

Для того чтобы в соответствии с настройками прерывания сгенерировался запрос начала преобразования АЦП, необходимо осуществить запись единиц в ADCSOCSET. Отключение генерации запросов начала преобразования АЦП осуществляется через регистр ADCSOCLR.

Для того чтобы в соответствии с настройками прерывания сгенерировался запрос RXEV к ядру, необходимо осуществить запись единиц в RXEVSET. Отключение генерации запросов RXEV к ядру осуществляется через RXEVCLR.

## 11.6 Механизм блокировки конфигурации

Для защиты конфигурации выводов от несанкционированного изменения реализован механизм блокировки.

Блокировка конфигурации осуществляется записью соответствующих единиц в регистр LOCKSET, сброс блокировки – в регистр LOCKCLR. При включенной блокировке игнорируется запись во все биты и поля настройки маскированных выводов – доступны для записи лишь регистры INTSTATUS и QUALSAMPLE.

По умолчанию все выводы являются разблокированными, а запись в регистры LOCKSET/LOCKCLR игнорируется. Для того чтобы сделать доступными для изменения регистры LOCKSET/LOCKCLR, необходимо в регистр LOCKKEY занести значение ключа ADEADBEEh. Если необходимо снова защитить от изменений регистры LOCKSET/LOCKCLR, то достаточно записать в LOCKKEY любое значение, отличное от ключа.

Текущий статус защиты регистров LOCKSET/LOCKCLR можно узнать, прочитав регистр LOCKKEY – значение 0000\_0001h соответствует снятой защите, 0000\_0000h – установленной.

### 11.7 Механизм маскирования

Для управления состоянием выводов порта дополнительно используется механизм маскирования. Он позволяет устанавливать желаемый уровень сигнала на нужном выводе, не затрагивая состояние других выводов. 16-разрядный порт условно разбивается на старший байт и младший байт. Для доступа по маске к младшему байту используется массив регистров MASKLB, а к старшему – MASKHB.

Каждый массив состоит из 256 регистров, каждый регистр имеет порядковый номер (от 00h до FFh), который является маской. Так, например, для порта A выделены две области памяти с адресами: 4001\_0400h – 4001\_07FCh для младшего байта и 4001\_0800h – 4001\_0BFCCh для старшего байта. Биты с 9 по 2 адреса являются маской. Таким образом, адресу 4001\_0400h соответствует маска 00h (MASKLB[0x00]), адресу 4001\_0404h – 01h (MASKLB[0x01]) и т. д.

Для того чтобы изменить состояние выводов порта с использованием маски, нужно записать новое значение в соответствующий элемент массива (MASKLB или MASKHB) с порядковым номером, совпадающим со значением маски.

Разряды порта, закрытые «нулями» маски, останутся неизменными, а остальные примут новые значения. На рисунке 11.4 показан механизм маскирования младшего байта порта A.

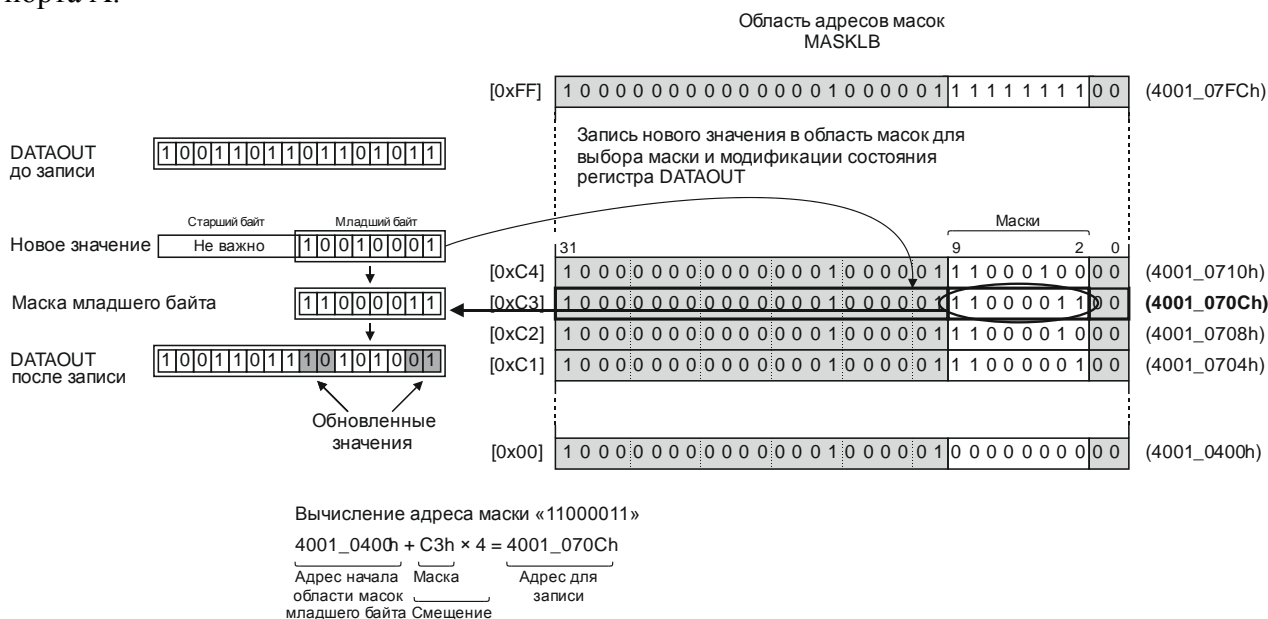


Рисунок 11.4 – Механизм изменения состояния младшего байта порта A с маскированием



Для изменения 0, 1, 6 и 7 битов регистра порта нужно использовать маску 1100\_0011b. Эта маска является частью (биты с 9 по 2) адреса 4001\_070Ch. Новое значение XX90h данных, которое требуется передать в порт (при этом старший байт числа не важен), нужно записать в ячейку с адресом 4001\_070Ch. Далее это значение будет аппаратно маскировано и размещено в регистре порта DATAOUT.

Аналогично выполняется маскирование старшего байта, см. рисунок 11.5. Разница лишь в том, что в данном случае берется старший байт нового значения, а младший не важен.

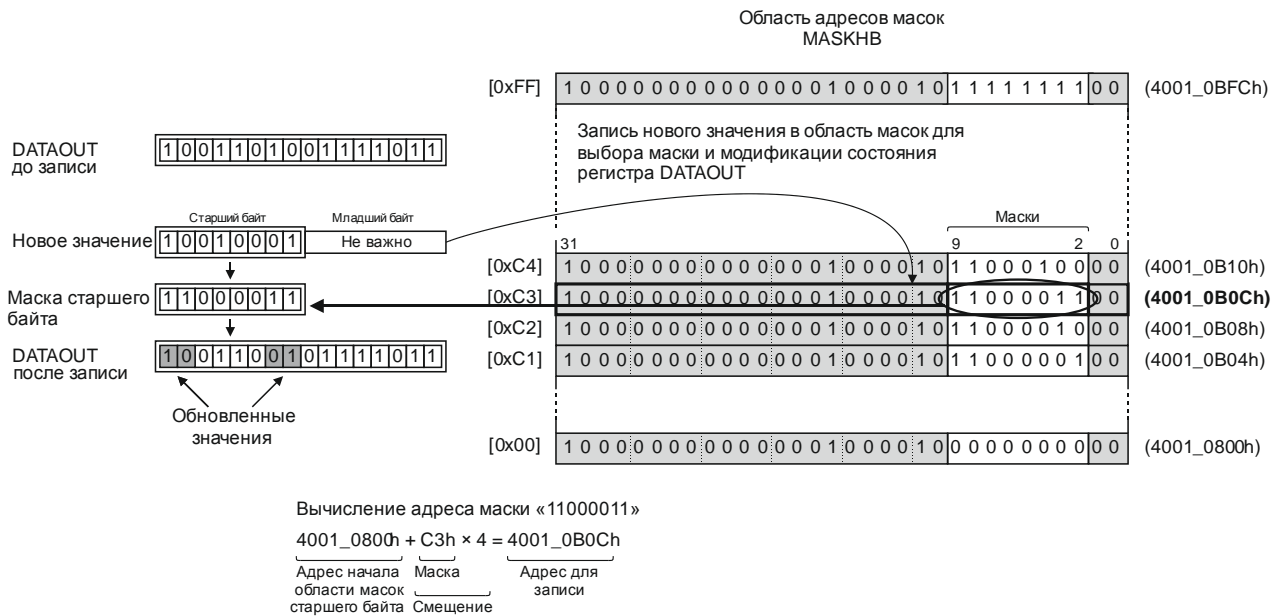


Рисунок 11.5 – Механизм изменения состояния старшего байта порта А с маскированием

## 12 Таймер TMR

Микроконтроллер содержит четыре блока 32-разрядных таймеров TMRx (где x от 0 до 3). Все блоки идентичны.

Счетчик таймера работает по системному тактовому сигналу. Кроме этого, таймер может управляться внешним сигналом, а также синхронизироваться по внешнему сигналу. На рисунке 12.1 представлена функциональная схема таймера.

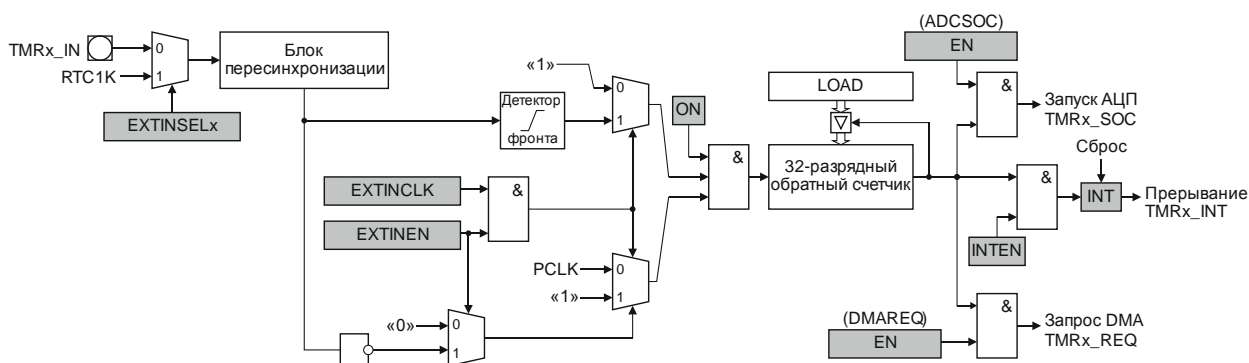


Рисунок 12.1 – Функциональная схема таймера

Управление таймером осуществляется посредством регистра CTRL. Начальное значение задается регистром VALUE. Для включения таймера нужно установить бит ON. Счетчик таймера декрементируется от значения, заданного регистром VALUE, до нуля на частоте тактового сигнала микроконтроллера. По достижении нуля счетчик таймера загружается значением, заданным регистром перезагрузки LOAD, и если разрешено битом INTEN, генерируется прерывание. При возникновении прерывания устанавливается флаг INT в регистре INTSTATUS.

Для каждого таймера имеется свой внешний вход синхронизации TMRx\_IN, выведенный на альтернативную функцию порта микроконтроллера. Дополнительно таймеры могут тактироваться сигналом RTC1K. Выбор между TMRx\_IN и RTC1K управляется состоянием бита EXTINSELx регистра TMRMUX блока SIU.

Если установлен бит EXTINEN, то счетчик таймера работает на частоте тактового сигнала микроконтроллера, только если сигнал на входе имеет уровень логической единицы.

Если одновременно установлены биты EXTINEN и EXTINCLK, то тактирование счетчика таймера происходит по положительному фронту внешнего сигнала. При этом частота внешнего сигнала должна быть, как минимум, в два раза меньше частоты системного тактового сигнала.

С помощью таймера возможно генерирование запроса DMA. Если записать в поле EN регистра DMAREQ единицу, то по достижении счетчиком нуля будет генерироваться запрос по соответствующему каналу DMA. Выбор необходимого канала DMA выполняется с помощью поля SRCSELx (где x – номер канала от 8 до 15) регистра DMAMUX блока SIU.

Также таймер может выдавать строб для запуска измерения АЦП. Если записать в поле EN регистра ADCSOC единицу, то по достижении счетчиком нуля будет генерироваться сигнал запуска АЦП.



## **13.2 Режимы счёта**

### **Режим свободного счёта**

По умолчанию счётчик работает в этом режиме – бит PERMODE сброшен.

Когда счётчик досчитает до нуля, то перезагружается максимальным значением, которое определяется разрядностью счётного регистра, и начинает снова декрементироваться.

### **Режим периодического счёта**

Переход в этот режим осуществляется установкой бита PERMODE.

Когда счётчик досчитает до нуля, то перезагружается значением из регистра LOAD и начинает декрементироваться.

### **Режим однократного счёта**

Переход в этот режим осуществляется установкой бита ONESH.

Счётчик считает от значения LOAD до нуля однократно. Чтобы продолжить счёт, бит ONESH должен быть сброшен (счёт начнется согласно состоянию PERMODE), или можно начать новый цикл однократного счёта, записав новое значение в LOAD.

## **13.3 Сопряжение с блоком DMA**

С помощью таймера возможно генерирование запроса DMA. Если записать в поле EN регистра DMAREQ единицу, то по достижении счетчиком нуля будет генерироваться запрос по соответствующему каналу DMA. Выбор необходимого канала DMA выполняется с помощью поля SRCSELx (где x – номер канала от 24 до 31) регистра DMAMUX блока SIU.

## 14 Блок программируемых логических ячеек LAU

Блок программируемых логических ячеек представляет собой набор настраиваемых логических макроячеек, соединенных как с внутренними сигналами контроллера, так и с внешними выводами (64 линии данных и 4 тактовых входа), и позволяет реализовывать широкий набор логических функций. Структурная схема блока показана на рисунке 14.1.

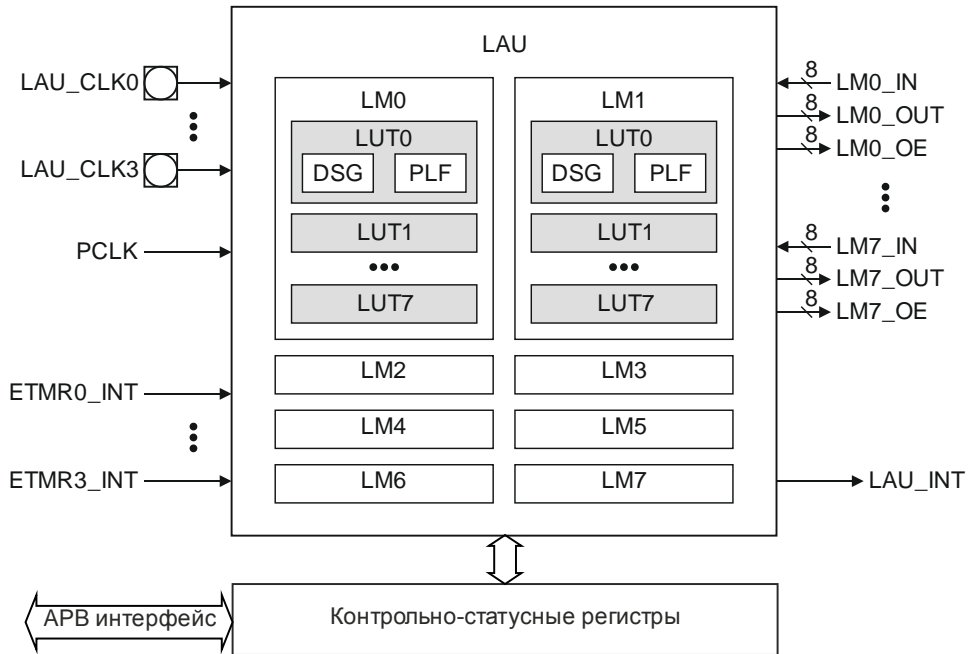


Рисунок 14.1 – Общая структурная схема блока программируемых логических ячеек

Каждая тройка выводов  $LMx\_IN[i]$ ,  $LMx\_OUT[i]$ ,  $LMx\_OE[i]$  (где  $x$  – номер LM блока от 0 до 7,  $i$  – номер LUT от 0 до 7) подключается к интерфейсу альтернативной функции выводов  $LAU\_LMx\_i$ .

### 14.1 Логический блок LUT

Минимальной структурной единицей является логический блок (Logic Unit – LUT). Он состоит из мультиплексора входных сигналов (Data Selection Gate – DSG) и непосредственно самой программируемой логической функции (Programmable Logic Function – PLF), см. рисунок 14.2.

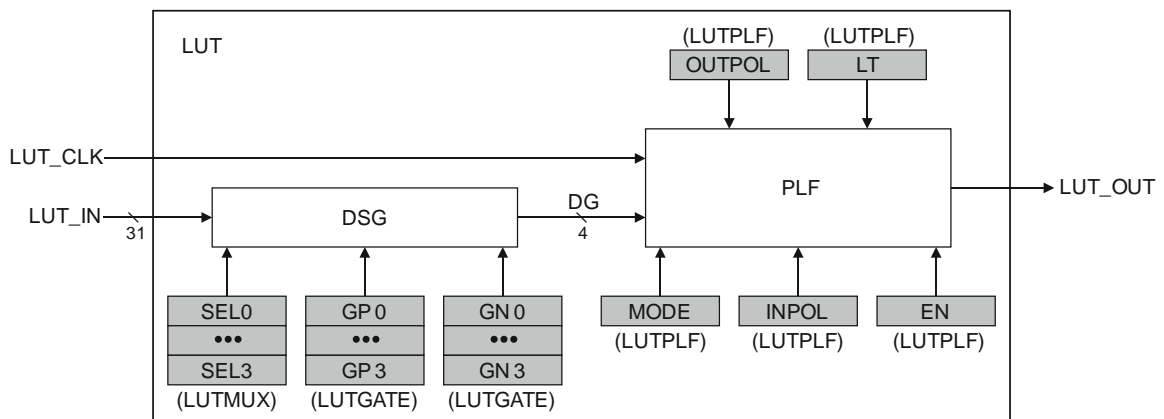


Рисунок 14.2 – Структурная схема одиночного логического блока LUT

На входную шину блока LUT\_IN поступает 31 сигнал, из которых с помощью мультиплексора DSG формируется 4-разрядная шина DG для построения требуемой логической функции. Также на вход блока поступает тактовый сигнал LUT\_CLK (подробнее о его выборе в подразделе 14.2).

### Мультиплексор входного сигнала DSG

Подробная структура мультиплексора DSG показана на рисунке 14.3.

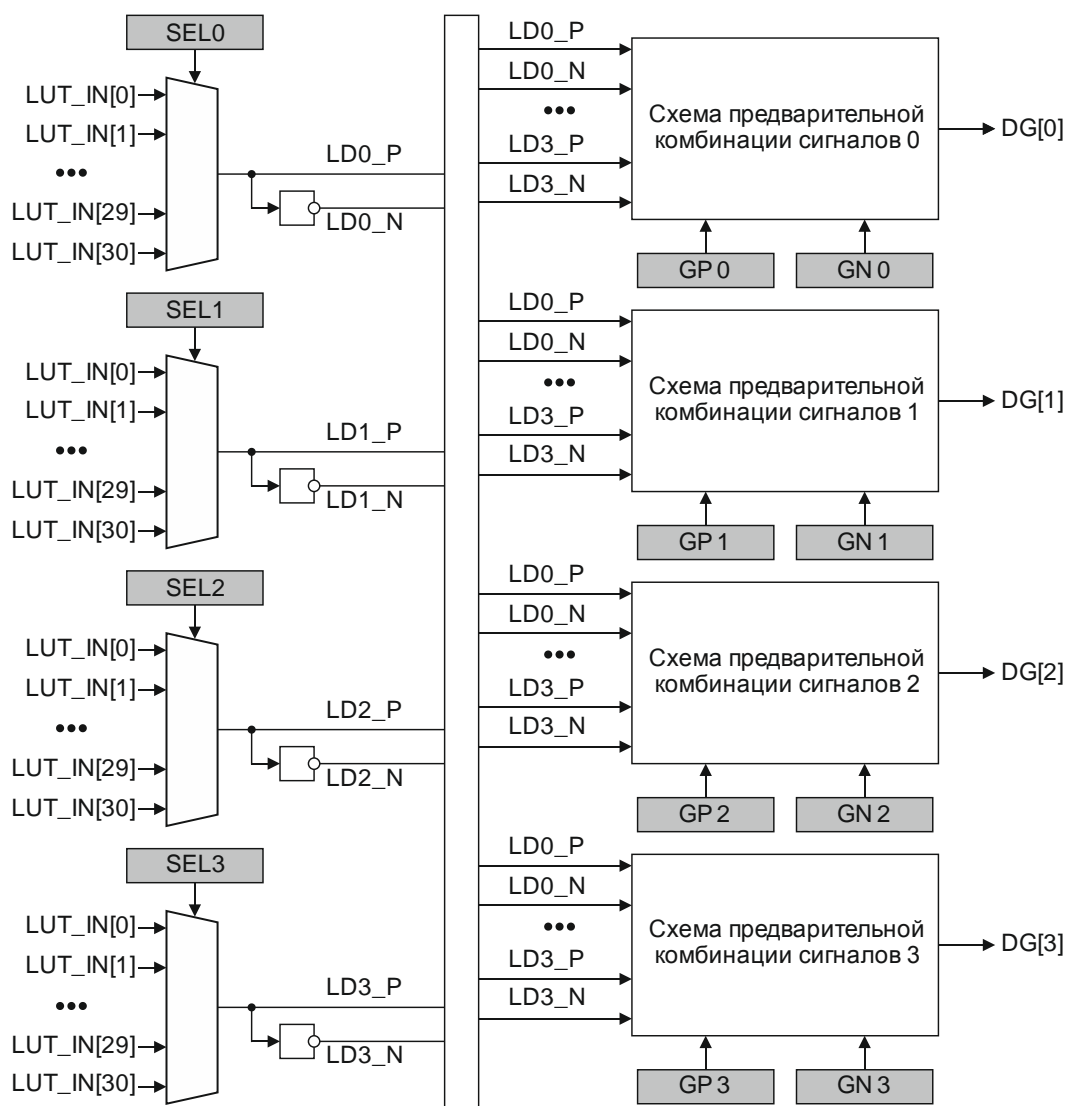


Рисунок 14.3 – Структура мультиплексора выбора входных сигналов DSG

Мультиплексор входного сигнала DSG управляется с помощью регистра LUTMUX. Поля SELn (где n от 0 до 3) используются для формирования четырёх промежуточных сигналов LD0\_P – LD3\_P и их инверсных копий LD0\_N – LD3\_N. Данные 8 сигналов поступают на n схем предварительной комбинации сигналов. Структура такой схемы показана на рисунке 14.4.

Бит поля GPn[i] (i от 0 до 3) регистра LUTGATE разрешает влияние i-го прямого сигнала на формирование n-го выхода DG. Соответственно, поле GNn[i] разрешает влияние соответствующего инверсного сигнала на DG[n].

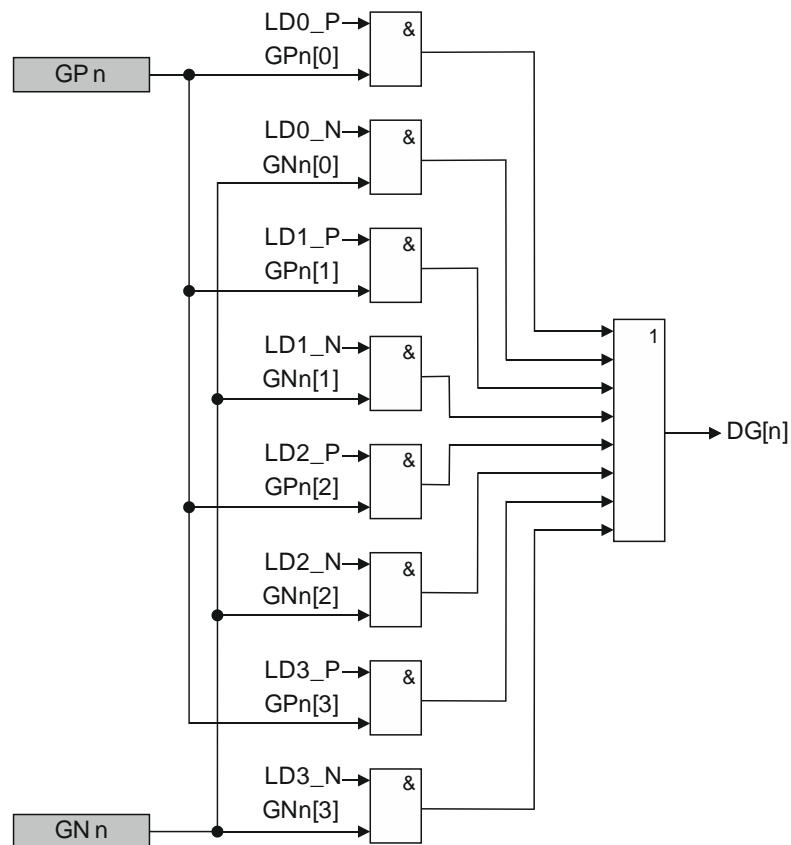


Рисунок 14.4 – Схема предварительной комбинации сигналов  $n$  ( $n$  от 0 до 3)

### Блок программируемой логической функции PLF

Блок программируемой логической функции PLF может преобразовать входную шину  $DG$ , используя одну из 7 функций: 2 варианта защелки Latch, 4 варианта триггера Flip-Flop и произвольную логическую функцию от четырех аргументов, заданную таблично. Выбор осуществляется с помощью поля  $MODE$  регистра  $LUTPLF$ . Дополнительно входные сигналы могут быть проинвертированы (поле  $INPOL$ ).

Разрешение работы выходного сигнала  $LUT\_OUT$  задается с помощью бита  $EN$ , а полярность контролируется полем  $OUTPOL$ . Структура PLF показана на рисунке 14.5.

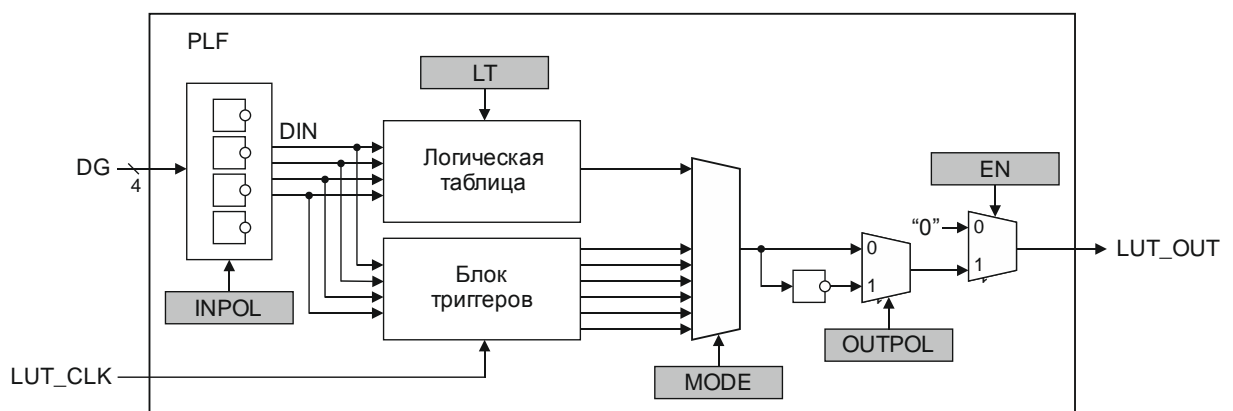


Рисунок 14.5 – Структурная схема блока программируемой логической функции PLF

Чтобы задать таблицу истинности, следует записать поле LT. Номер бита внутри поля соответствует определенной логической комбинации DIN, см. таблицу 14.1, а значение бита – состояние выхода таблицы при соответствующем состоянии входов. Таблица используется в качестве логической функции, если поле MODE = 1h.

Таблица 14.1 – Соответствие состояния входов логической таблицы и управляющего бита

DIN[3]	DIN[2]	DIN[1]	DIN[0]	Бит поля LT регистра LUTPLF
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Блок триггеров состоит из шести запоминающих устройств, два из которых являются асинхронными, а четыре тактируются по LUT\_CLK сигналу, см. рисунок 14.6.



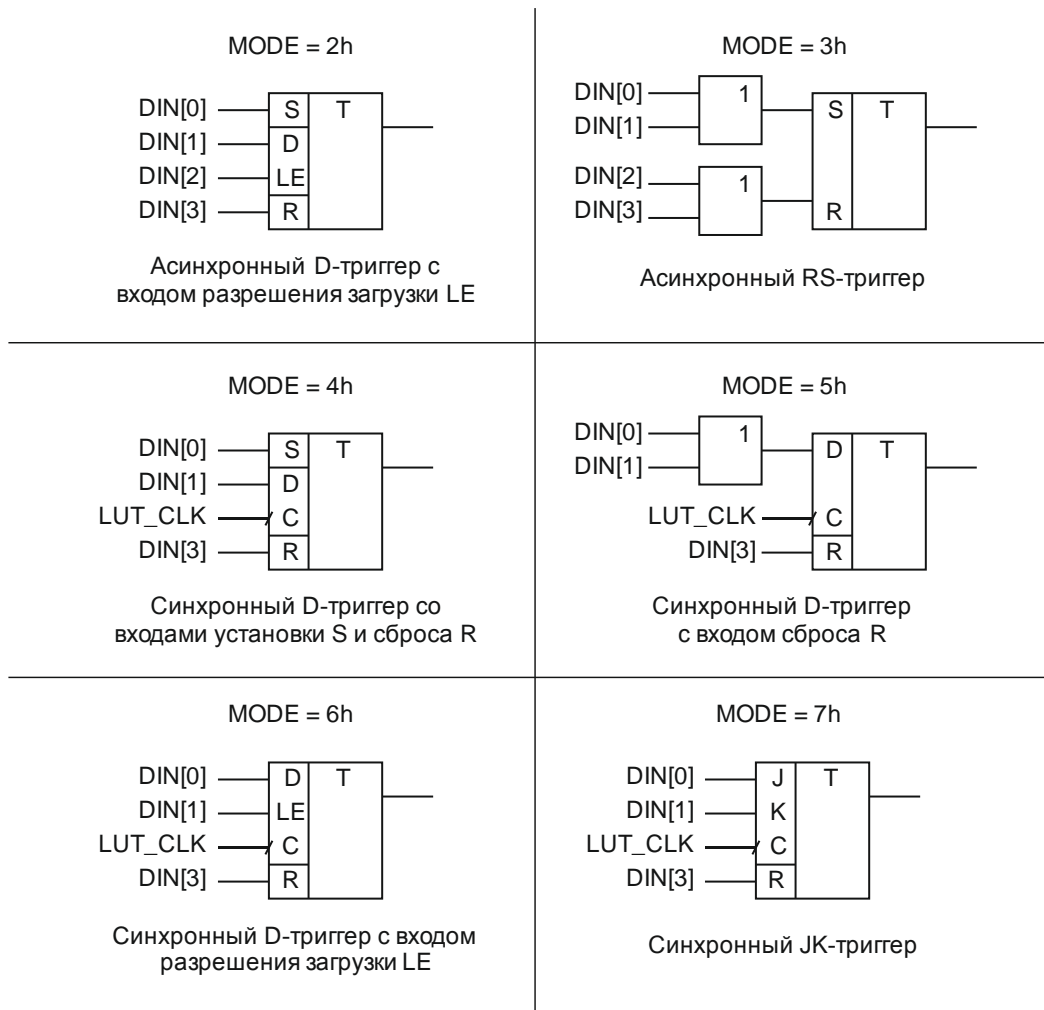


Рисунок 14.6 – Используемые в PLF типы триггеров

## 14.2 Логическая макроячейка LM

Каждые восемь блоков LUT объединяются в логическую макроячейку (Logic Macrocell – LM).

Выходы LUT соединены в шину, являющуюся частью входной шины LUT\_IN, которая служит источником формирования сигналов для PLF. Помимо этого в шину входят сигналы логического нуля и единицы, четыре сигнала прерываний от таймеров ETMR, восемь внешних входов LM\_IN, восемь линий LM\_BUS от других модулей LM. С помощью этих сигналов можно строить как логические функции, зависящие от внутренних сигналов системы, так и функции, использующие внешние сигналы, сигналы от других LUT и LM блоков.

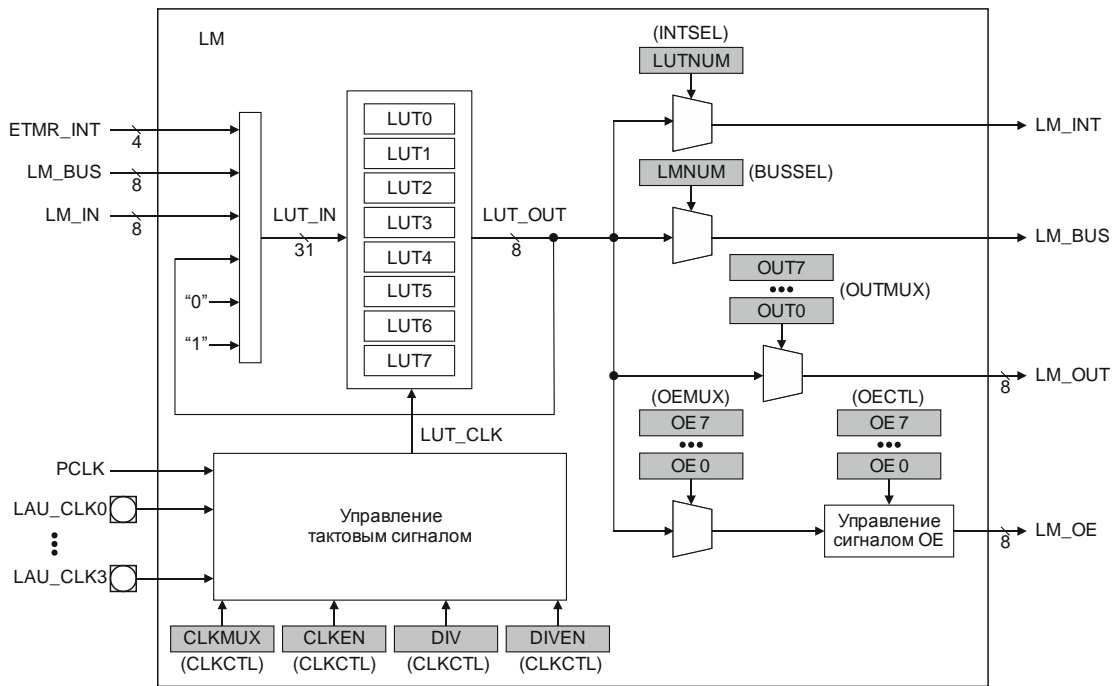


Рисунок 14.7 – Общая структурная схема блока LM

Один из выходных сигналов LUT может быть выбран в качестве сигнала прерывания. Также один из этих сигналов может быть выведен в общую шину коммутации блоков LM\_BUS. За выбор отвечают регистры INTSEL и BUSSEL соответственно.

Восемь независимых мультиплексов отвечают за выбор сигналов LUT в качестве выходных сигналов LM\_OUT, ведущих к портам. Еще восемь мультиплексов выбирают сигналы, используемые в качестве сигналов направления входа/выхода LM\_OE.

Каждый блок LM формирует восемь альтернативных функций порта LAU\_LM<sub>x</sub><sub>i</sub> (где x – номер LM блока от 0 до 7, i – номер LUT от 0 до 7). Каждый вывод может быть настроен в качестве входа или выхода. Причем направление может переключаться динамически в зависимости от сигнала, выбранного в качестве OE. В регистре OEMUX можно выбрать эти сигналы независимо для каждого бита. А с помощью регистра OECTL можно задать поведение – режим входа, выхода или динамическое переключение.

Отдельно стоит рассмотреть подсистему тактовых сигналов, см. рисунок 14.8.

С помощью регистра CLKCTL можно выбрать тактовый сигнал, который может быть использован при тактировании триггеров внутри LUT. В качестве источника может выступать тактовый сигнал PCLK или один из четырех внешних сигналов с выводов LM\_CLK0 – LM\_CLK3. При выборе сначала рекомендуется настраивать делитель, а затем разрешать сам сигнал битом CLKEN. Также следует иметь в виду, что если используются деленные тактовые сигналы, не гарантируется синхронность фаз для разных LM блоков.

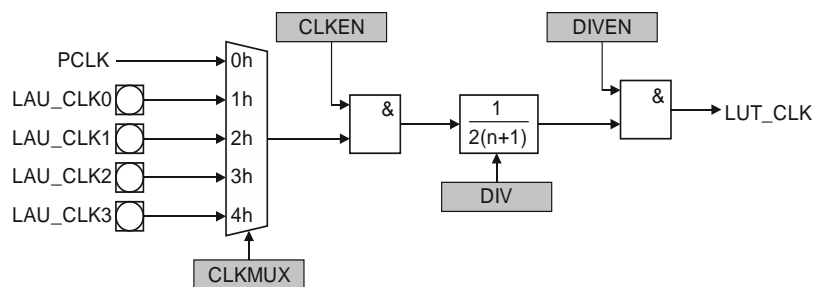


Рисунок 14.8 – Управление тактовым сигналом внутри LM

### 14.3 Логический массив LAU

Восемь блоков LM объединены в логический массив (Logic Array Unit – LAU). Структура LAU показана на рисунке 14.9.

Одиночные выходы LM\_BUS каждого блока LM собираются в одноименную шину и подключаются к входам всех макроячеек.

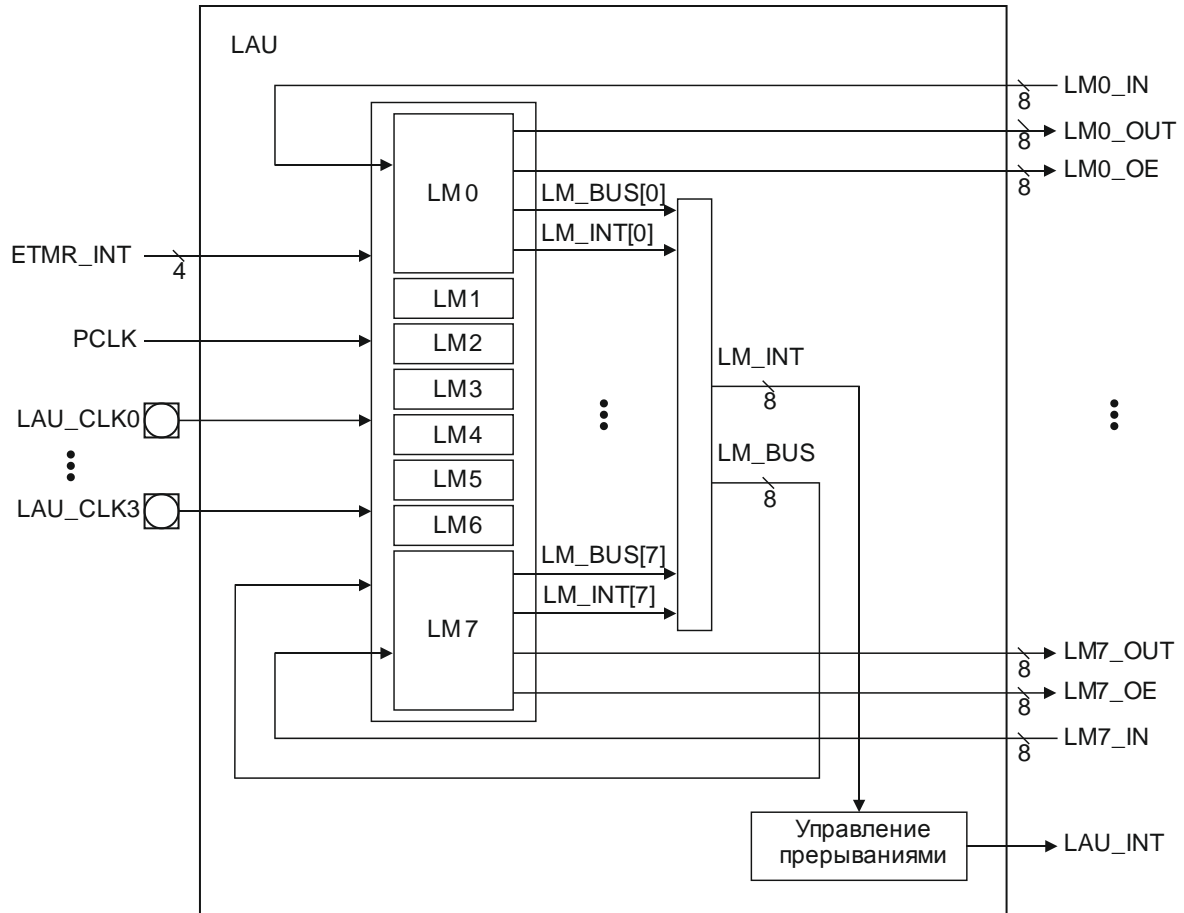


Рисунок 14.9 – Общая структурная схема блока LAU

Также на уровне логического массива осуществляется контроль прерываний. Каждая из линий прерываний блоков LM может быть маскирована (регистр IMSC). Состояние линии прерываний захватывается в регистры флагов RIS и MIS, сброс которых можно осуществить с помощью регистра ICR. В конечном итоге, линии прерываний от LM блоков объединяются по «ИЛИ» и формируют одно прерывание LAU\_INT, см. рисунок 14.10, где x – номер LM блока от 0 до 7.

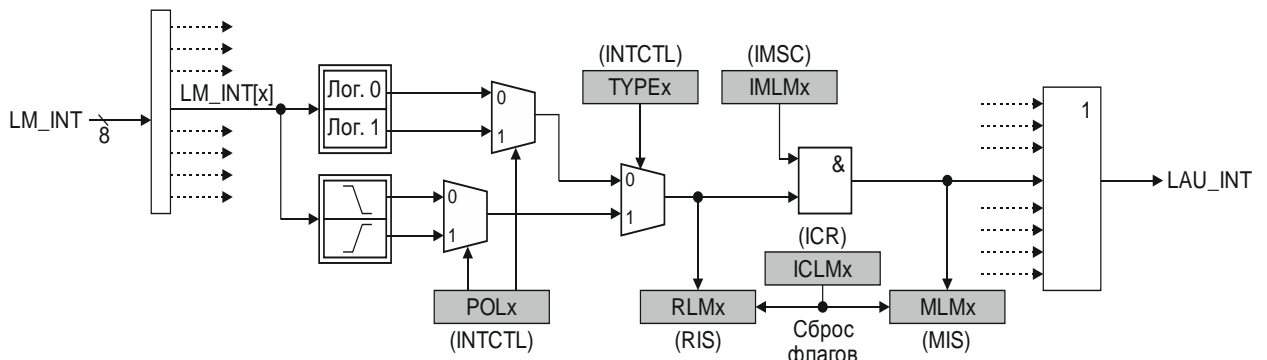


Рисунок 14.10 – Схема управления прерываниями

## 15 Блок тригонометрических вычислений TMU

Блок тригонометрических операций TMU позволяет проводить различные типы математических операций, часто применяемых в области векторного управления электроприводом:

- вычисление синуса и косинуса;
- перевод из прямоугольной системы координат в полярную;
- перевод из полярной системы координат в прямоугольную;
- поворот вектора на заданный угол.

Выбор функции осуществляется с помощью поля FUNC регистра CMD.

Блок TMU не связан с ядром, т. е. является периферийным АНВ блоком и имеет регистровую модель управления. Время вычисления любой из функций равняется 12 тактам SYCLK, дополнительно по одному такту тратится на загрузку аргументов и выгрузку результата.

Поддерживаются различные типы аргументов – форматы чисел с фиксированной точкой от Q28 до Q8, а также IEEE 754 формат представления чисел с плавающей точкой одинарной точности. Формат задается полем ARGТ регистра CMD. Результаты выдаются в том же формате, в котором были представлены входные аргументы. Разрешение и диапазон значений Q-форматов представлен в сводной таблице 15.1.

Таблица 15.1 – Диапазон и разрешение форматов представления чисел с фиксированной точкой

Формат	Диапазон		Разрешение
	Минимум	Максимум	
Q28	-8	7,999999996	0,000000004
Q27	-16	15,999999993	0,000000007
Q26	-32	31,999999985	0,000000015
Q25	-64	63,999999970	0,000000030
Q24	-128	127,999999940	0,000000060
Q23	-256	255,999999981	0,000000119
Q22	-512	511,999999762	0,000000238
Q21	-1024	1023,999999523	0,000000477
Q20	-2048	2047,999999046	0,000000954
Q19	-4096	4095,999998093	0,000001907
Q18	-8192	8191,999996185	0,000003815
Q17	-16384	16383,999992371	0,000007629
Q16	-32768	32767,999984741	0,000015259
Q15	-65536	65535,999969482	0,000030518
Q14	-131072	131071,999938965	0,000061035
Q13	-262144	262143,999877930	0,000122070
Q12	-524288	524287,999755859	0,000244141
Q11	-1048576	1048575,999511719	0,000488281
Q10	-2097152	2097151,999023437	0,000976563
Q9	-4194304	4194303,998046875	0,001953125
Q8	-8388608	8388607,996093750	0,003906250

Входные аргументы для вычисления задаются регистрами XIN, YIN, PHIN, результаты сохраняются в XOUT, YOUT, PHOUT. Назначение регистров варьируется в зависимости от вычисляемой функции.

Вычисления начинаются после установки бита START в регистре CMD. Таким образом, все управляющие поля находятся в одном регистре и могут быть сконфигурированы одной записью.

Окончание вычислений может быть определено тремя способами:

- программный контроль;
- чтение флага;
- блокирующее чтение регистра результата.

Программный контроль подразумевает расположение пустых операций следом за стартом вычислений, в том количестве, чтобы счетчик команд дошел до операции чтения результата после появления его в регистре. Например, 12 NOP команд или другой способ организации программной задержки.

Регистр STAT содержит флаг DONE, который установится, когда вычисление будет завершено. Сбрасывается флаг записью единицы.

Установка бита WAITRD в регистре CMD позволит осуществить блокирующее чтение любого из регистров результатов средствами АНВ-шины. Т. е. сразу после начала вычисления можно осуществить команду чтения из регистра результата, но завершена она будет только после завершения вычисления. Попытка таким образом прочитать результат без установленного бита WAITRD приведет к тому, что будет получено некорректное значение.

### 15.1 Вычисление синуса и косинуса

Параметры регистров входных аргументов и результатов при вычислении синуса и косинуса показаны в таблице 15.2.

Таблица 15.2 – Диапазон и значение регистров при вычислении синуса и косинуса

Регистр	Диапазон		Значение
	Минимум	Максимум	
XIN	–	–	Не используется
YIN	–	–	Не используется
PHIN	0	$2\pi$	Входной угол $\varphi$
XOUT	–1	1	$\cos(\varphi)$
YOUT	–1	1	$\sin(\varphi)$
PHOUT	–	–	Не используется

### 15.2 Перевод из прямоугольной системы координат в полярную

Параметры регистров входных аргументов и результатов при переводе из прямоугольной системы координат в полярную показаны в таблице 15.3.

Таблица 15.3 – Диапазон и значение регистров при переводе из прямоугольной системы координат в полярную

Регистр	Диапазон		Значение
	Минимум	Максимум	
XIN	–1	1	Координата $x$
YIN	–1	1	Координата $y$
PHIN	–	–	Не используется
XOUT	0	$\sqrt{2}$	Радиальная координата $r = \sqrt{x^2 + y^2}$
YOUT	–	-	Не используется
PHOUT	0	$2\pi$	Угловая координата $\varphi = \text{atan2}(y/x)$

### 15.3 Перевод из полярной системы координат в прямоугольную

Параметры регистров входных аргументов и результатов при переводе из полярной системы координат в прямоугольную показаны в таблице 15.4.

Таблица 15.4 – Диапазон и значение регистров при переводе из полярной системы координат в прямоугольную

Регистр	Диапазон		Значение
	Минимум	Максимум	
XIN	0	$\sqrt{2}$	Радиальная координата $r$
YIN	–	–	Не используется
PHIN	0	$2\pi$	Угловая координата $\varphi$
XOUT	–1	1	Координата $x = r \times \cos(\varphi)$
YOUT	–1	1	Координата $y = r \times \sin(\varphi)$
PHOUT	–	–	Не используется

### 15.4 Поворот вектора на заданный угол

Параметры регистров входных аргументов функции и результатов при повороте вектора на заданный угол показаны в таблице 15.5.

Таблица 15.5 – Диапазон и значение регистров при повороте вектора на заданный угол

Регистр	Диапазон		Значение
	Минимум	Максимум	
XIN	–1	1	Координата $x_0$
YIN	–1	1	Координата $y_0$
PHIN	0	$2\pi$	Угол поворота $\varphi$
XOUT	–1	1	Новая координата $x_1 = x_0 \times \cos(\varphi) - y_0 \times \sin(\varphi)$
YOUT	–1	1	Новая координата $y_1 = y_0 \times \cos(\varphi) + x_0 \times \sin(\varphi)$
PHOUT	–	–	Не используется

### 15.5 Генерация запросов к DMA

Установка бита DMAREQ в регистре CMD позволит генерировать запросы DMA по окончании вычислений.

### 15.6 Генерация исключений

Вычислительное ядро TMU работает с Q28 форматом внутри, все остальные форматы аппаратно приводятся к Q28. Диапазон значений Q28 представлен в таблице 15.1. Он накладывает некоторые ограничения на IEEE 754 формат представления чисел с плавающей точкой. Возможна генерация четырех типов исключений для каждого из регистров XIN, YIN, PHIN:

- входное значение является нечислом (NaN);
- входное значение является бесконечностью (Inf);
- слишком малое значение (недозаполнение формата Q28);
- слишком большое значение (переполнение формата Q28).

При генерации любого из исключений устанавливается соответствующий флаг в регистре STAT. Сброс флагов происходит автоматически при старте нового вычисления.

## **15.7 Генерация прерываний**

Блок TMU позволяет генерировать два вида прерываний.

Установка бита DONEIRQ в регистре CMD позволит генерировать прерывание по окончании вычислений.

Установка бита EXCREQ в регистре CMD позволит генерировать прерывание по появлению любого из исключений.

## 16 Часы реального времени RTC

Часы реального времени RTC предназначены для отсчета времени в микроконтроллере. Данный блок продолжает работу после отключения основного питания VDD33 и VDD12. При этом происходит переключение на батарейное питание RTC\_BAT.

Блок RTC работает от батарейного питания напряжением (1,6 – 3,6) В.

В качестве источника тактового сигнала RTCCLK используется кварцевый генератор с подключенным кварцевым резонатором (выводы XI\_RTC, XO\_RTC) с частотой 32,768 кГц.

На рисунке 16.1 показана схема подключения внешнего генератора синхросигнала.

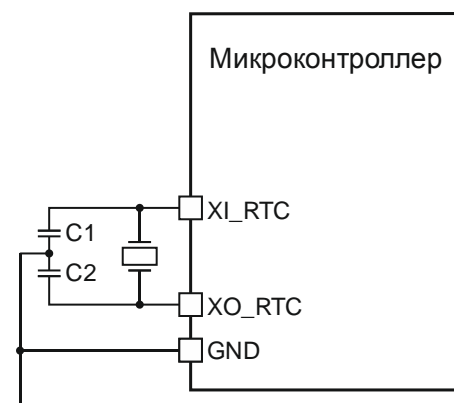


Рисунок 16.1 – Схема подключения внешнего генератора синхросигнала

### 16.1 Рекомендации по подключению и трассировке сигналов на печатной плате

Основные требования к подключению:

- минимизировать емкостные связи между сигналами RTC и другими сигналами;
- избегать параллельной трассировки сигналов RTC с другими высокоскоростными сигналами на печатной плате;
- поместить кварцевый осциллятор как можно ближе к выводам микросхемы;
- использовать симметричную трассировку для сигналов XI\_RTC и XO\_RTC;
- номиналы емкостей  $C1 = C2 = (10 - 50)$  пФ,  $C3 > 4,7$  мкФ

### 16.2 Функционирование

Функционирование RTC основано на подсчете импульсов тактового сигнала RTC1K частотой 1,024 кГц, который в свою очередь получается делением внешнего входного сигнала RTCCLK частотой 32,768 кГц на 32 с помощью 5-разрядного счетчика.

Счетчик долей секунд инкрементируется по сигналу RTC1K. Когда значение счетчика достигает 1024, инкрементируется счетчик секунд. По достижении счетчиком значения 59 инкрементируется счетчик минут и т. д. до счетчика лет.

Все счетчики имеют двоично-десятичный формат, за исключением счетчика долей секунд, который имеет двоичный формат. Каждый счетчик имеет теневой регистр, из которого загружается по команде в программе пользователя.

Блок RTC может корректно обрабатывать високосные года. Максимально значение лет, которые может отсчитать блок RTC, составляет 99.

Инициализация счетчиков осуществляется записью корректных значений.

Примечание – Сигнал сброса микроконтроллера не сбрасывает счетчики блока RTC, а только их теневые регистры. После снятия сигнала сброса рекомендуется подождать не менее 1 мс, прежде чем осуществлять операции записи/чтения блока RTC.



### **Корректные значения для записи в регистры блока RTC**

В течение работы значения счетчиков постоянно переписываются в теньевые регистры по заднему фронту тактового сигнала. В это время запись и чтение регистров блока RTC запрещена. Для считывания информации о текущем времени и дате следует запретить перезаписывание теньевых регистров путем сброса бита UPDTEN регистра SHDW (при этом счетчики продолжают инкрементироваться).

Только после этого можно выполнить операцию чтения над одним или несколькими регистрами. Также можно считать состояние общего регистра TIME. После того как все желаемые регистры прочитаны нужно снова включить обновление (перезаписывание) теньевых регистров, установив бит UPDTEN.

Для того чтобы изменить значение одного или нескольких счетчиков, следует аналогично выключить обновление теньевых регистров (сброс UPDTEN), последовательно записать новые значения, после чего снова включить обновление. Новые значения будут загружены в соответствующие счетчики по ближайшему положительному фронту тактового сигнала RTC1K.

Примечание – Задержка времени от включения обновления до перезаписи счетчиков составляет два такта синхросигнала RTC1K и равняется 2 мс. Эта задержка может накапливаться, поэтому если перезагрузка счетчиков происходит без остановки обновления теньевых регистров, запись в регистр долей секунд следует производить в последнюю очередь.

### **16.3 Генерация прерываний**

Сигнал прерывания RTC\_INT подключен к счетчику секунд. Каждый раз, когда значение обновляется, генерируется прерывание.

### **16.4 Сопряжение с таймерами**

Сигнал RTC1K может использоваться для осуществления тактирования таймеров TMR0 – TMR3. Подробнее об этом можно узнать в разделе 12.

## 17 Сторжевой таймер WDT

Сторжевой таймер позволяет сбросить систему в случае отказа программного обеспечения. Пользователь может включать или выключать сторожевой таймер по собственному усмотрению.

Сторжевой таймер представляет собой 32-битный обратный счетчик, который загружается значением из регистра LOAD. Счетчик уменьшается на единицу по каждому нарастающему фронту тактового сигнала WDTCLK.

По умолчанию сторожевой таймер не тактируется и находится в сбросе. Активировать таймер можно с помощью регистра WDTCFG блока RCU.

Включение счета сторожевого таймера и его прерывания осуществляются установкой бита INTEN в регистре CTRL. Когда счетчик таймера достигает нуля, устанавливается флаг WDTINT в регистре MIS, а в счетчик загружается значение из регистра LOAD.

Далее, если установлен бит RESEN, счетчик продолжает декрементироваться. Если на момент повторного достижения нуля флаг WDTINT установлен, производится сброс микроконтроллера. Алгоритм работы сторожевого таймера показан на рисунке 17.1.

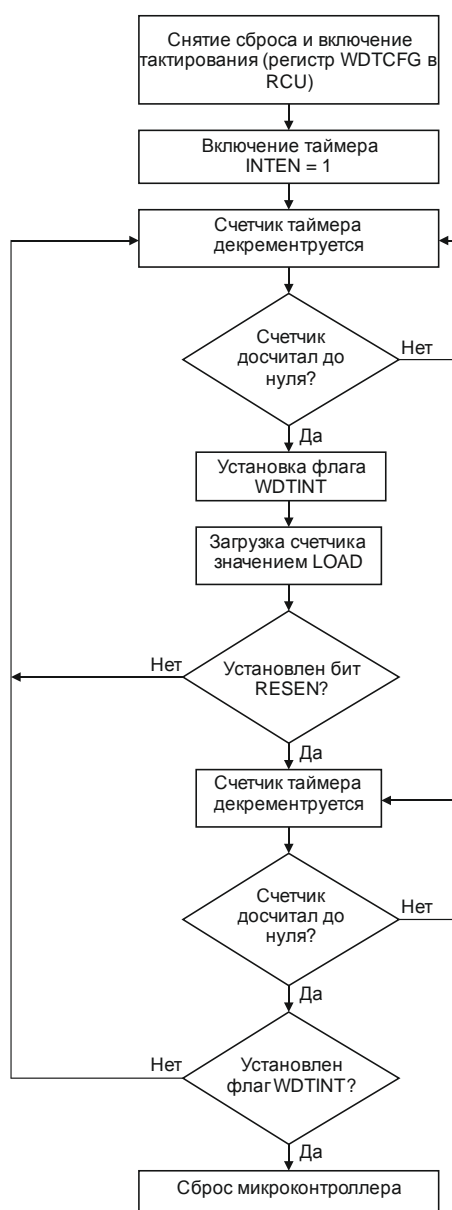


Рисунок 17.1 – Алгоритм работы сторожевого таймера

## 18 Блок АЦП

Блок АЦП объединяет четыре модуля АЦП последовательного приближения (архитектура SAR), схему управления, буферы результатов измерений и схему управления прерываниями. Структурная схема блока АЦП показана на рисунке 18.1.

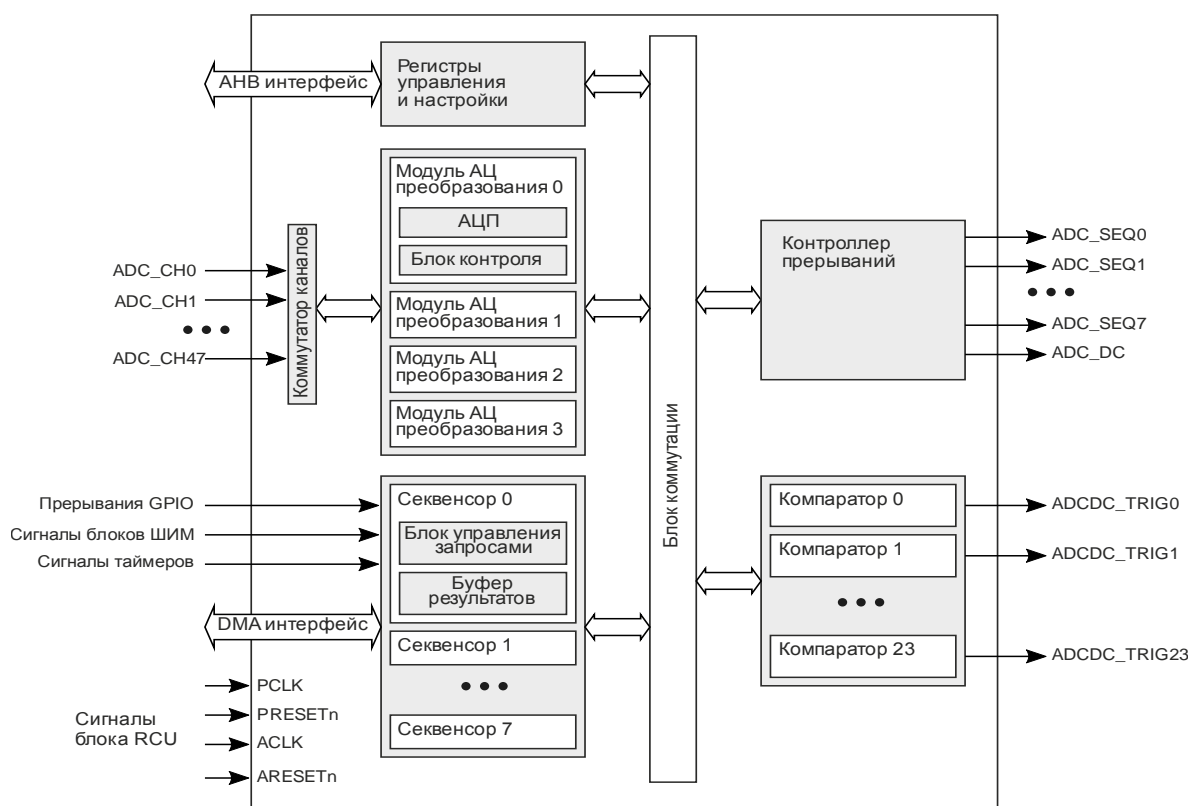


Рисунок 18.1 – Структурная схема блока АЦП

В блок АЦП входят:

- четыре 12-канальных модуля АЦП с настраиваемой разрядностью 6, 8, 10, 12 бит и скоростью измерения по одному каналу в 12-битном режиме до 2,5 миллионов измерений в секунду;
- восемь секвенсоров, каждый из которых позволяет независимо произвести запуск измерений по необходимым каналам АЦП и сгенерировать прерывание;
- 24 независимых цифровых компаратора, отслеживающих и сравнивающих измерения с пороговыми значениями для формирования прерываний и сигналов управления другими блоками микроконтроллера;
- восемь буферов результатов измерений (каждый организован по типу FIFO);
- блок управления прерываниями.

Блок АЦП суммарно имеет 48 аналоговых входных каналов. **Диапазон измеряемых напряжений ограничен AREF и AGND.**

Настройка тактирования и сброса блока АЦП и его модулей осуществляется посредством регистра ADCCFG блока управления тактовыми сигналами RCU. Вся внутренняя логика блока АЦП тактируется частотой ACLK, но запись/чтение контрольно-статусных регистров осуществляется на частоте PCLK.

Для правильной работы блока, необходимо обеспечить тактирование модулей АЦП частотой ACLK от 140 кГц до 35 МГц, которую можно получить, выбрав источник тактового сигнала полем CLKSEL, а также, при необходимости, включив и настроив делитель полями DIVEN и DIVN (поля регистра ADCCFG блока RCU). Частота ACLK не должна быть больше частоты тактового сигнала PCLK.

## 18.1 Секвенсор

Секвенсор представляет собой управляющий блок, позволяющий разгрузить процессор от управления модулями АЦП. Секвенсор управляет запуском модулей АЦП, обработкой полученных результатов измерений и генерацией прерываний. В состав блока АЦП входят  $s$  секвенсоров (где  $s$  от 0 до 7). Структурная схема секвенсора представлена на рисунке 18.2.

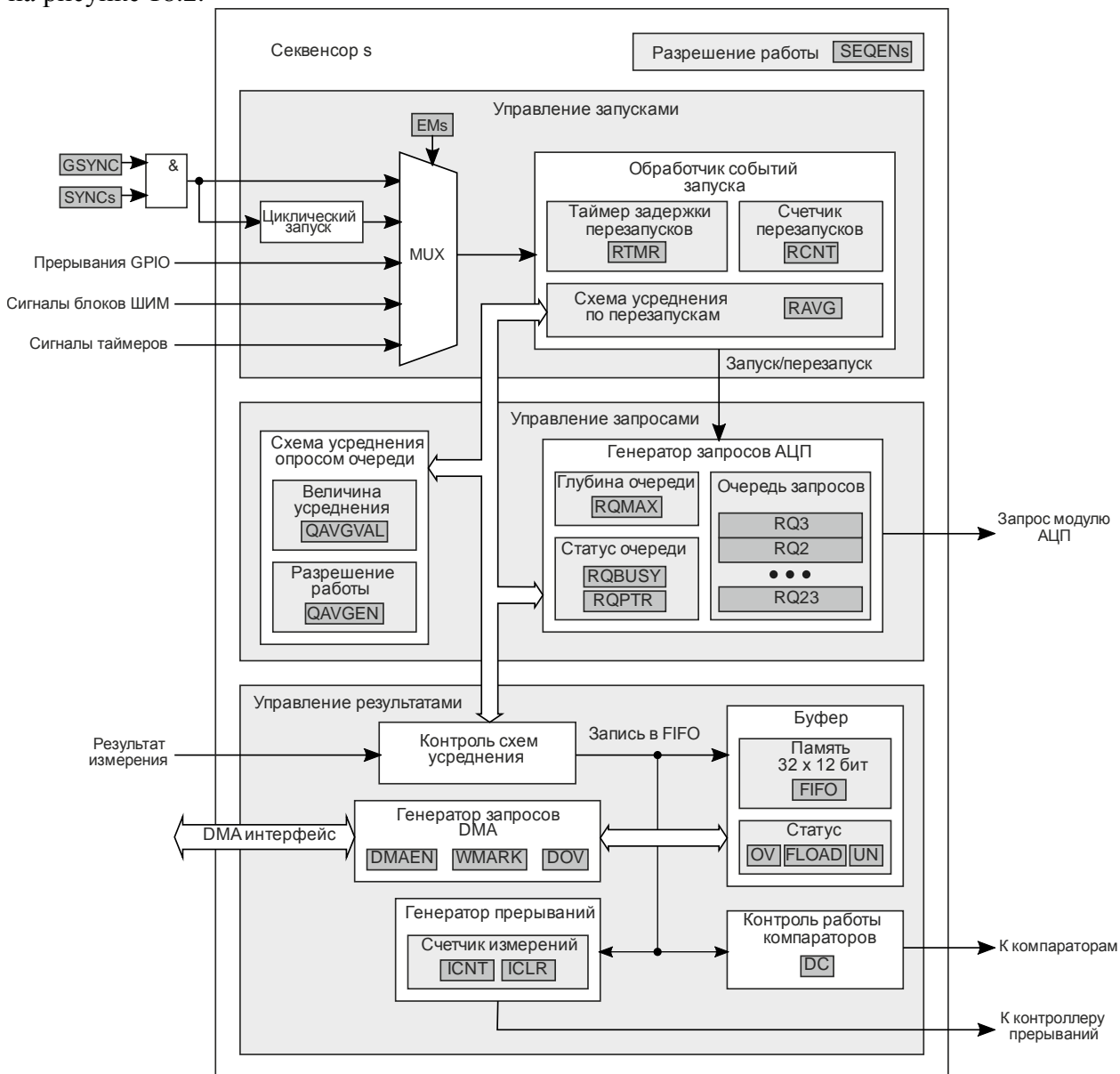


Рисунок 18.2 – Структурная схема секвенсора

### Одиночные запуски по событиям

Разрешение работы секвенсора осуществляется установкой соответствующего бита SEQENs (где  $s$  от 0 до 7) в регистре SEQEN.

Каждый секвенсор может совершать независимые однократные запуски по одному из событий, которое выбирается полем EMs регистра EMUX:

- установка бита GSYNC регистра SEQSYNC (запустятся только секвенсоры, для которых установлены биты SYNCs того же регистра);
- сигналы от таймеров TMR, ETMR;
- сигналы от блоков ШИМ;
- сигналы от GPIO.

Когда секвенсор *s* запускается по одному из сигналов событий, выставляется соответствующий флаг *SEQBUSYs* в регистре *BSTAT*. Также в это же время все настройки секвенсора сохраняются в теневых регистрах, и секвенсор начинает работу согласно полученным настройкам. Изменять настройки секвенсора во время его работы можно, но вступят в силу они лишь при следующем запуске. Флаг занятости держится установленным до тех пор, пока задача, инициированная событием, не будет полностью выполнена секвенсором (будет осуществлена запись последнего результата в *FIFO*).

События запуска не кэшируются, если секвенсор был занят выполнением текущей задачи (установлен *SEQBUSYs*), когда пришло очередное событие запуска, то оно будет проигнорировано. Необходимо учитывать это при настройке запуска по событиям от сторонних периферийных модулей так, чтобы время между возникновением событий было не меньше времени измерений. Диаграммы работы секвенсора при одиночных запусках по событиям показаны на рисунке 18.3.

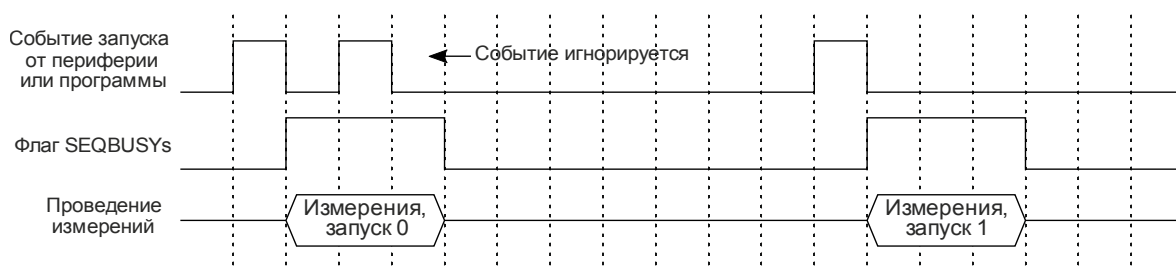


Рисунок 18.3 – Диаграммы работы секвенсора при одиночных запусках по событиям

#### Одиночные запуски по событиям с немедленными перезапусками

Здесь и далее, под перезапусками понимается внутренний механизм работы секвенсора, а под запусками – приход внешнего события, которое переводит секвенсор из ожидающего состояния в активное.

Секвенсор имеет возможность осуществлять как отложенные, так и немедленные автоматические перезапуски серий измерений, после прихода первого «инициирующего» события запуска от периферии. Перезапуск может выполняться до 255 раз (поле *RCNT* регистра *SCCTL*). Текущее состояние счетчика перезапусков можно узнать, прочитав поле *RCNT* регистра *SCVAL*. Диаграммы работы секвенсора при разрешенных, немедленных перезапусках показаны на рисунке 18.4.

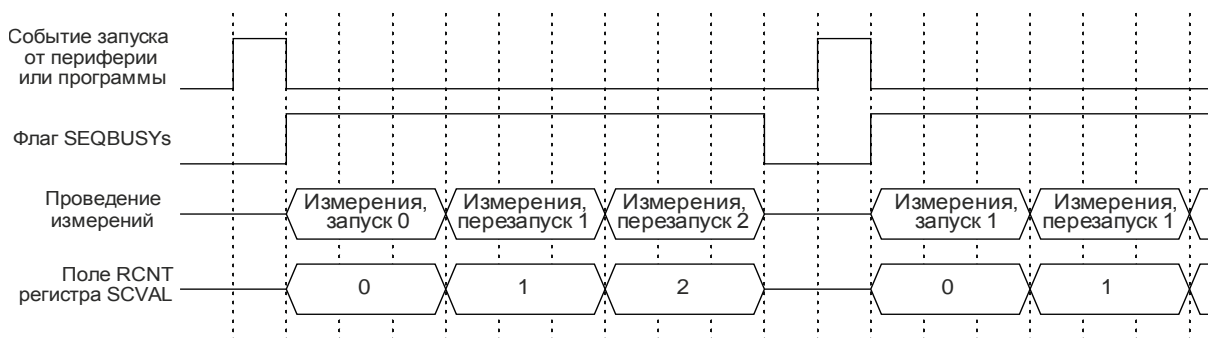


Рисунок 18.4 – Диаграммы работы секвенсора при одиночных запусках по событиям с немедленными перезапусками при  $RCNT = 2$  (регистр *SCCTL*)

#### Одиночные запуски по событиям с отложенными перезапусками

Отложенные перезапуски осуществляются спустя некоторое время от запуска, задаваемое регистром *SRTMR*. Задержка задается в тактах *ACLK* и ведет счет независимо от текущего состояния секвенсора, поэтому, при её выборе необходимо учитывать, что текущие измерения должны завершиться до прихода сигнала перезапуска, иначе он будет пропущен.

В режиме одиночных запусков по событиям счетчик задержки перезапуска не будет считать, если поле RCNT регистра SCCTL равно нулю.

Как говорилось ранее, все настройки секвенсора сохраняются в теневых регистрах при его переходе в режим занятости и их изменение никак не повлияет на текущую работу. Однако существует возможность обновить задержку перезапуска еще во время работы секвенсора. Для этого надо записать новое значение задержки в регистр SRTMR с одновременно установленным последним битом NOWAIT. В этом случае, новая величина задержки вступит в силу после ближайшего события отложенного перезапуска.

Диаграммы работы секвенсора при активных отложенных перезапусках показаны на рисунке 18.5.

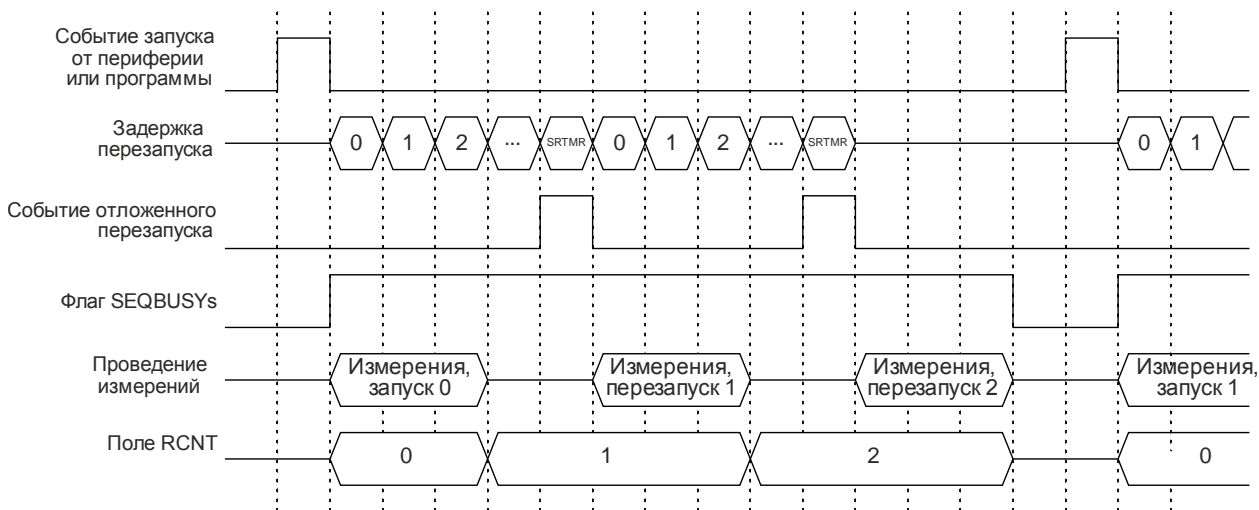


Рисунок 18.5 – Диаграммы работы секвенсора при одиночных запусках по событиям с отложенными перезапусками через определенное время (регистр SRTMR) при RCNT = 2 (регистр SCCTL)

### Одиночные запуски с усреднением по перезапускам

Существует режим усреднения результатов по перезапускам, который включается установкой бита RAVGEN в регистре SCCTL. Главным условием работы этого режима является то, что поле RCNT регистра SCCTL должно содержать любое значение, соответствующее  $2^p - 1$ , где  $p$  равно от 1 до 8. Значение  $2^p$  и является количеством серий измерений, которые будут усреднены (запуск и все перезапуски).

Работа этого режима заключается в том, что пока идут перезапуски, результаты попадут в буфер не сразу, а будут накапливаться во внутренних регистрах (каждому запросу на измерение соответствует такой регистр). Лишь во время последнего перезапуска, будут получены усреднённые значения по каждому из измерений, которые и будут помещаться в FIFO в порядке очереди.

Работа режима усреднения при немедленных перезапусках продемонстрирована на рисунке 18.6. При отложенных перезапусках данный вид усреднения работает аналогично, позволяя распределить равномерно измерения на длительном промежутке времени и получить среднее значение сигнала в конце него.

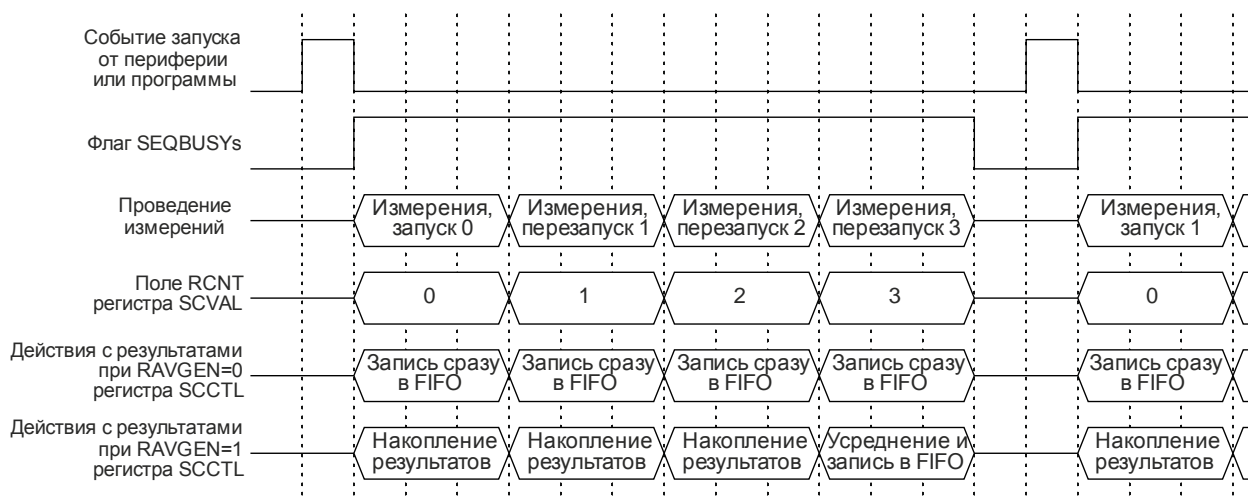


Рисунок 18.6 – Диаграммы работы секвенсора при одиночных запусках по событиям с усреднением по перезапускам при RCNT=3, RAVGEN=1 (регистр SCCTL)

Например, если измерения проводились по четырём каналам при RAVGEN = 0, то на момент снятия флага SEQBUSYs в FIFO было бы 16 результатов, но если усреднение по перезапускам было бы активно, то в FIFO находилось бы четыре усредненных результата по каждому из каналов.

### Циклический запуск

Секвенсор может быть запрограммирован на циклический запуск – он будет запускаться снова каждый раз при завершении предыдущего запуска (имитация постоянно активного внешнего события). Чтобы начать работу в циклическом режиме, необходимо, после соответствующей конфигурации регистра EMUX, установить бит GSYNC регистра SEQSYNC. Чтобы завершить работу в циклическом режиме, необходимо выбрать в поле EMs регистра EMUX любое событие однократного запуска. Флаг занятости секвенсора SEQBUSYs в регистре BSTAT будет установлен сразу же по входу в циклический режим и будет сброшен только лишь по выходу из него. Диаграммы работы секвенсора при циклическом запуске показаны на рисунке 18.7.



Рисунок 18.7 – Диаграммы работы секвенсора при циклическом запуске

### Циклический отложенный запуск

В отличие от режима одиночных запусков, циклический режим позволяет активировать счетчик задержки SRTMR, даже если поле RCNT регистра SCCTL равно нулю. В этом случае измерения будут запускаться не непрерывно, а с некоторой паузой (определяемой регистром SRTMR). Диаграммы работы секвенсора при циклическом запуске показаны на рисунке 18.8.

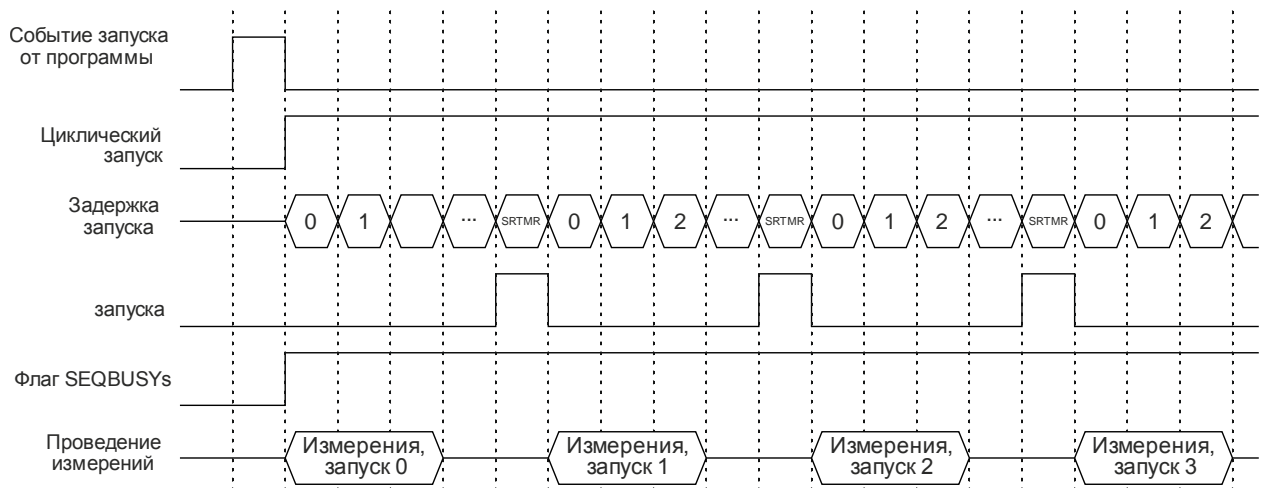


Рисунок 18.8 – Диаграммы работы секвенсора при циклическом отложенном запуске через время SRTMR

### Циклический запуск с усреднением по перезапускам

Если в циклическом режиме установить поле RCNT регистра SCCTL значением, отличным от нуля, то секвенсор между запусками начнет делать нужное количество перезапусков. Но механика такого режима внешне никак не будет отличаться от обычной циклической работы, поэтому данный режим имеет смысл лишь в том случае, когда необходимо скомбинировать циклический режим с усреднением по перезапускам (как немедленным, так и отложенным).

Диаграммы работы секвенсора в циклическом режиме запуска с усреднением по отложенным перезапускам показаны на рисунке 18.9.

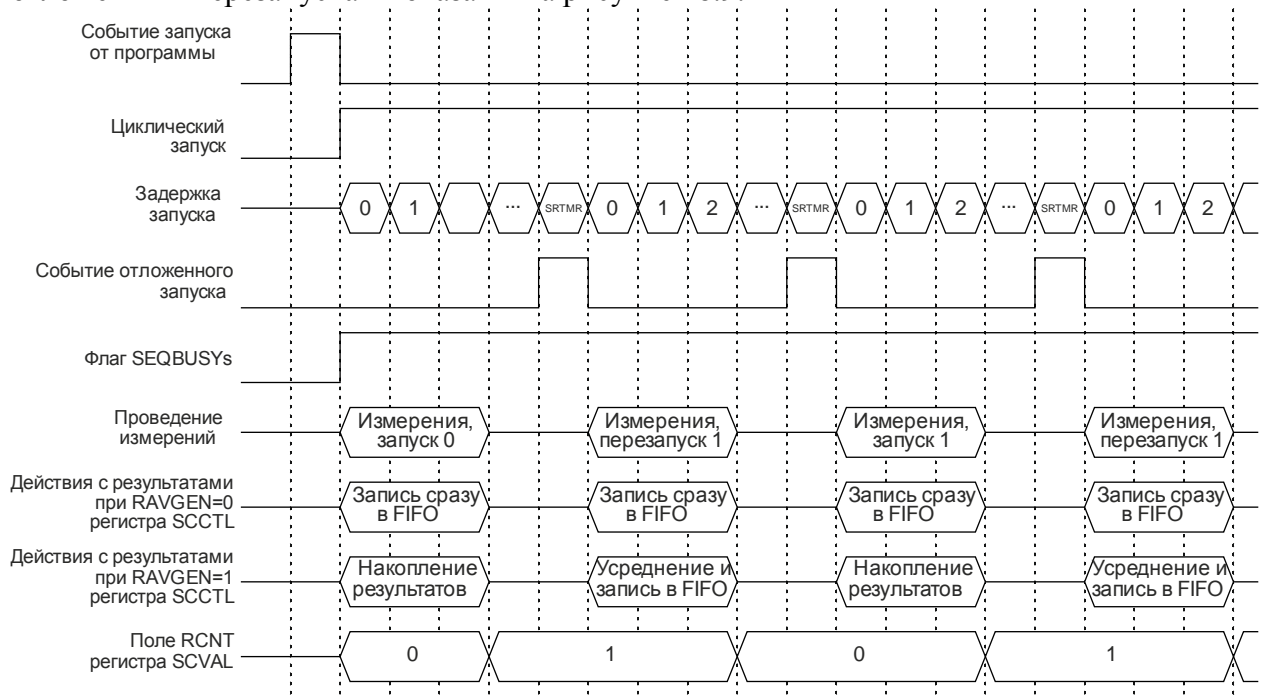


Рисунок 18.9 – Диаграммы работы секвенсора в циклическом запуске с отложенными перезапусками через определенное время (регистр SRTMR) при RCNT=1 (регистр SCCTL)

### Генерация запросов на измерение

После запуска по событию или очередного перезапуска секвенсор начинает формировать запросы на измерения по каналам в порядке очереди, заданной регистром SRQSEL. Конец очереди или её «глубина» задается полем RQMAX регистра SRQCTL. Определить состояние очереди можно с помощью регистра SRQSTAT – в поле RQPTR



находится номер текущего запроса по порядку, а установленный флаг занятости RQBUSY говорит о том, что запрос выставлен и в состоянии обработки.

Иллюстрация к механизму генерации запросов на измерение представлена на рисунке 18.10.

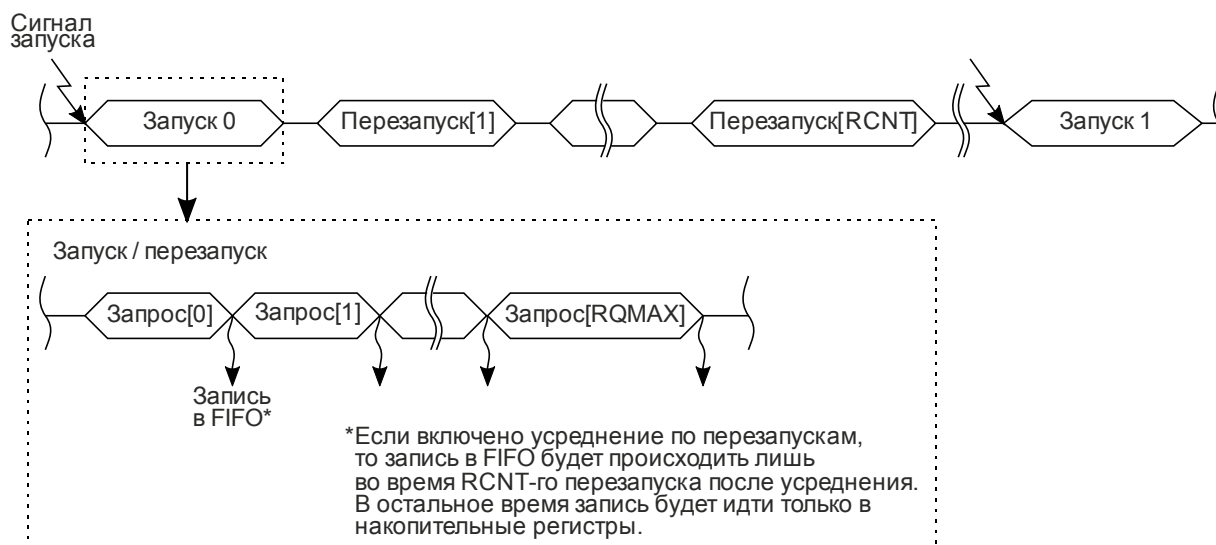


Рисунок 18.10 – Генерация секвенсором запросов на измерение

### Пояснения к рисунку 18.10

1 Как только секвенсор получает сигнал запуска, он выставляет первый запрос из очереди и ожидает его принятия модулем АЦП.

2 Когда запрос будет принят, АЦП проведет измерение. По окончании измерения результат выполнения запроса будет передан секвенсору.

3 Секвенсор сохраняет результат в FIFO и продолжает выставлять запросы до окончания их очереди.

4 Если секвенсор настроен на проведение перезапусков (отложенных или немедленных), то по окончании очереди секвенсор перезапускается и снова выставляет первый в очереди запрос.

5 Лишь когда завершён последний перезапуск, то работа секвенсора, начатая по первичному событию запуска (пояснение 1), считается завершённой.

6 Секвенсор переходит в состояние ожидания следующего сигнала запуска.

Результат каждого запроса сохраняется в кольцевом буфере результатов секвенсора SFIFO. Буферы всех секвенсоров имеют ёмкость в 32 12-битных слова. Текущее количество слов в буфере можно узнать, прочитав регистр SFLOAD.

Контроль состояния буфера осуществляется посредством флагов регистра FSTAT. Установленный флаг OVs, указывает на то, что в FIFO не осталось свободных ячеек. Любая запись в буфер в таком случае будет игнорироваться до появления хотя бы одной свободной ячейки. Сброшенный флаг UNs свидетельствует о наличии в FIFO как минимум одного результата измерения. Соответственно, флаг UNs установится, когда FIFO будет полностью пуст, при этом результатом чтения пустого FIFO будут нули. Сброс флагов осуществляется путем записи в них единицы.

### Усреднение сканированием очереди

Наряду с усреднением по перезапускам секвенсор имеет дополнительный механизм усреднения – усреднение сканированием (опросом) очереди измерений. Оба механизма могут работать как вместе, так и по отдельности.

Включается усреднение сканированием путем установки бита QAVGEN в регистре SRQCTL. Перед включением режима необходимо задать количество усредняемых опросов в поле QAVGVAL того же регистра – оно считается как  $2^{QAVGVAL}$ .

Работа режима заключается в том, что очередь запросов выполняется не один раз (рисунок 18.10) с сохранением результатов в буфер после каждого запроса, а  $2^{QAVGVAL}$  раз с сохранением результатов лишь на последней итерации сканирования после усреднения всех полученных измерений. Результаты измерений накапливаются и усредняются индивидуально для каждого запроса (аналогично усреднению по перезапускам).

Диаграммы работы усреднения сканированием показаны на рисунке 18.11.

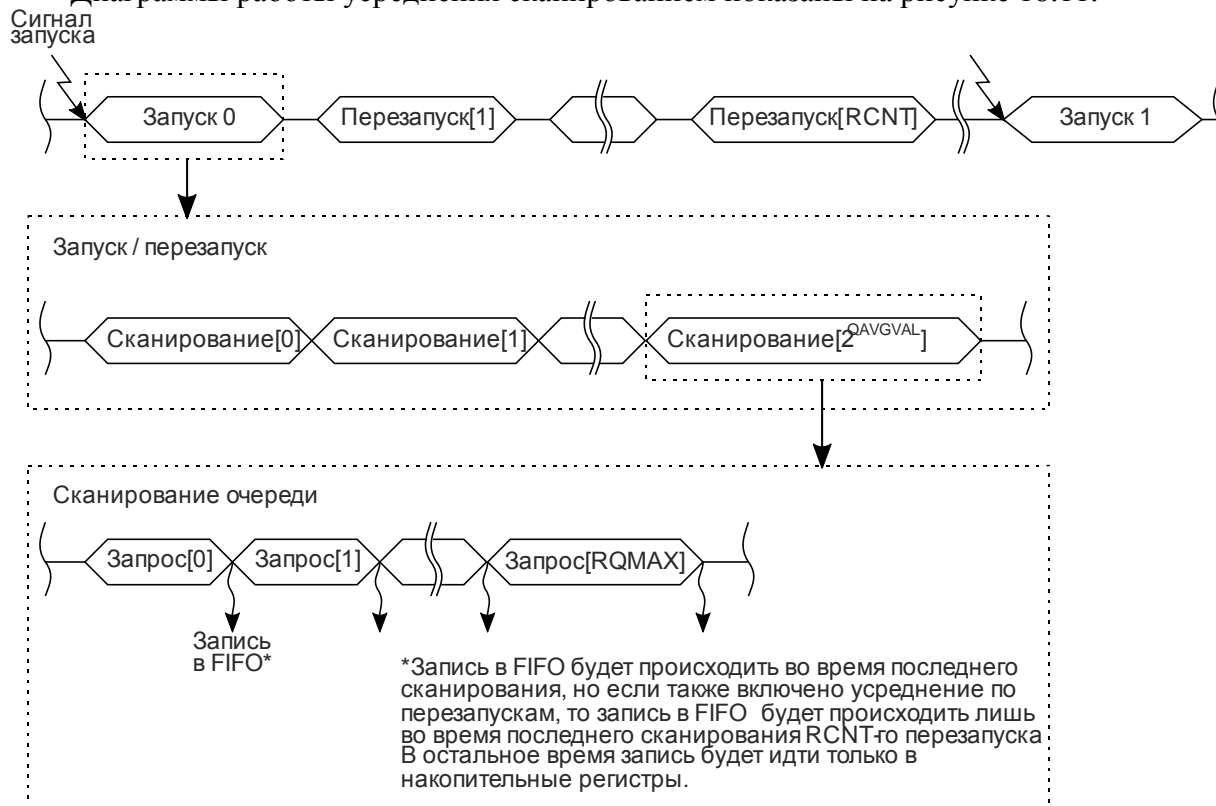


Рисунок 18.11 – Режим усреднения сканированием

Усреднение сканированием позволяет с относительно малой фазовой задержкой измерить уровень сигнала по нескольким каналам в пределах одной временной точки, а усреднение по перезапускам дает возможность усреднить значения, полученных таким образом точек, на достаточно большом интервале времени. В совокупности, оба механизма предлагают довольно гибкий инструментарий для автоматизации и усреднения измерений.

### Генерация прерываний

Секвенсор может генерировать прерывания с заданной периодичностью. По завершении каждой записи в FIFO инкрементируется счетчик измерений. Как только было зафиксировано ICNT+1 записей (поле регистра SCCTL), генерируется прерывание. Стоит отметить, что даже если FIFO заполнено полностью (установлен OVs в регистре FSTAT), а измерения продолжают проводиться – счетчик измерений все равно будет считать последующие попытки записи секвенсора в FIFO (хотя они и будут игнорироваться самим FIFO). Текущее состояние счетчика можно узнать, прочитав поле ICNT регистра SCVAL. Сброс счетчика запросов происходит в следующих случаях:

- зафиксировано ICNT+1 записей;
- при запуске секвенсора по событию, если сброшен бит ICNTs в регистре CICNT;
- бит разрешения работы секвенсора ENs регистра SEQEN сброшен;
- программно – при каждой записи единицы в поле ICLR регистра SCVAL.

Для каждого секвенсора выделена линия прерываний – ADC\_SEQs.

## Использование прямого доступа к памяти

Для разрешения использования DMA секвенсором, необходимо установить бит DMAEN в регистре SDMACTL. Поле WMARK того же регистра задает уровень заполнения буфера секвенсора, по достижении которого будет запущен DMA. Перенос данных будет выполняться, пока не будет передано число результатов измерений, соответствующее состоянию поля WMARK.

Если очередной запрос на запуск DMA пришел раньше, чем закончился предыдущий цикл DMA от того же секвенсора, то будет выставлен флаг ошибки DOVs в регистре FSTAT.

## 18.2 Модуль АЦП

Структурная схема одного 12-канального модуля АЦП показана на рисунке 18.12. Суммарное количество модулей АЦП равняется  $m$  ( $m$  равно 0 – 3). Каналы ADC\_CH[0 – 11] подключены к модулю 0, ADC\_CH[12 – 23] – к модулю 1, ADC\_CH[24 – 35] – к модулю 2, ADC\_CH[36 – 47] – к модулю 3.

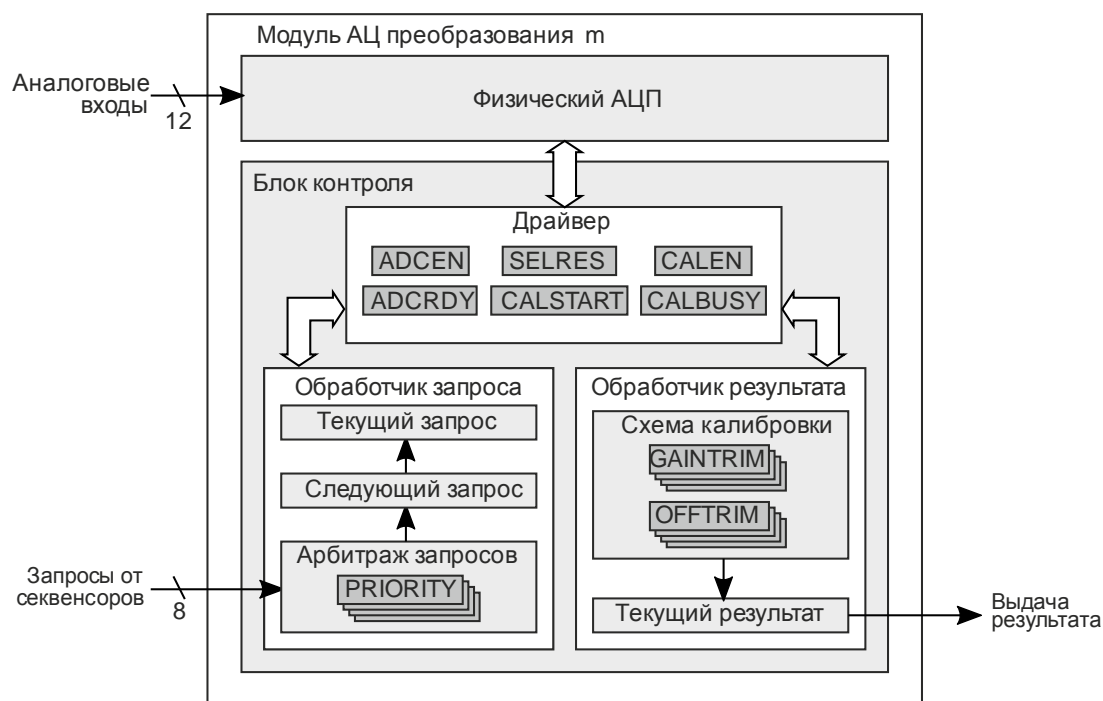


Рисунок 18.12 – Структурная схема 12-канального модуля АЦП

Каждый физический АЦП содержит внутренний линейный регулятор напряжения (LDO), который должен быть включен (блок PMU регистр ADCPC) перед разрешением работы модуля.

Разрешение работы модуля АЦП $m$  (где  $m$  от 0 до 3) осуществляется установкой бита ADCEN в регистре ACTL $m$ . При каждом переключении ADCEN в единицу запускается процедура инициализации АЦП, которая завершается установкой бита ADCRDY регистра ACTL $m$ . АЦП готов к работе, когда ADCRDY = 1. Для правильной работы блока, необходимо обеспечить тактирование модулей АЦП (сигнал ACLK) частотой от 140 кГц до 35 МГц.

Физический блок АЦП имеет внутреннюю схему калибровки смещения нуля. Схема включается при установке бита CALEN регистра ACTL $m$ . Ручной запуск калибровки запускается при установленном бите CALEN по событию записи единицы в бит CALSTART того же регистра. В начале процесса калибровки устанавливается статусный бит CALBUSY, который сбросится по её окончании. Также по завершению процесса обновится поле CALOUT, которое будет содержать полученное поправочное значение.

Длительность процедуры – 82 такта ACLK.

Поправочное значение можно загрузить вручную – величина записывается в поле CALIN регистра ACTLm, затем устанавливается бит CALLOAD.

Процесс внутренней калибровки также может быть запущен автоматически в процессе инициализации АЦП, если на момент переключения ADCEN в единицу бит CALEN был установлен.

Разрядность результата настраивается с помощью поля SELRES регистра ACTLm – можно осуществить выбор между 6, 8, 10, 12 бит. Выравнивается результат по младшему биту. В таблице 18.1 показаны соответствия режима разрядности и длительности преобразования.

Таблица 18.1 – Зависимость скорости преобразования одного канала от разрядности данных

Поле SELRES регистра ACTL	Разрядность данных, бит	Количество тактов сигнала ACLK на преобразование	Максимальная скорость, выборки/с
00b	6	8	$4.38 \times 10^6$
01b	8	10	$3.50 \times 10^6$
10b	10	12	$2.92 \times 10^6$
11b	12	14	$2.50 \times 10^6$

Общая процедура включения модуля АЦПm:

1 Включить LDO внутри АЦП, установив единицу в соответствующее поле LDOENm регистра PMU – ADCPC.

2 Дождаться установки флага готовности LDORDYm в том же регистре.

3 Настроить и включить тактирование АЦП частотой ACLK от 140 кГц до 35 МГц, снять сброс (регистр RCU – ADCCFG).

4 Если необходимо, активировать автоматическую внутреннюю калибровку по включению – установить бит CALEN регистра ACTLm.

5 Разрешить работу модуля АЦП с помощью установки бита ADCEN в регистре ACTLm.

6 Если на момент разрешения работы был установлен CALEN, то запустится процедура калибровки. Если этот бит был сброшен, то выполнится одиночное пустое преобразование.

7 По окончании иницилирующих процедур установится бит ADCRDY регистра ACTLm. Модуль АЦП готов к работе.

Процедура деинициализации модуля АЦПm для уменьшения потребления (например, при переходе в DEEPSLEEP):

1 Модуль АЦП должен находиться в состоянии ожидания – измерения не проводятся.

2 Сбрасывается бит ADCEN регистра ACTLm.

3 Сбрасывается бит LDOENm регистра PMU – ADCPC.

### Правила арбитража

Когда АЦП начинает выполнять измерения по запросам, он устанавливает флаг ADCBUSY в регистре BSTAT. Флаг занятости будет сброшен лишь при полном отсутствии запросов, при последовательном выполнении запросов флаг не сбрасывается. Во время установки флага все настройки каналов АЦП сохраняются в теневых регистрах. Изменять их настройки во время работы можно, но вступят в силу они лишь при следующей установке флага ADCBUSY после сброса.

Секвенсоры могут выставлять запросы как одновременно, так и независимо друг от друга по разным событиям запуска. Для определения порядка выполнения запросов модулем АЦП реализована схема арбитража, со следующими правилами работы:

1 Если модуль АЦП был в режиме ожидания (включен и измерения не проводятся), то при поступлении запроса незамедлительно начинается его обслуживание.

2 Секвенсоры выставляют «ждущие» запросы – запрос будет активен до тех пор, пока модуль АЦП не начнет его обработку. Как только его обслуживание началось, запрос сбрасывается.

3 Как только текущий запрос секвенсора был сброшен, он сразу выставляет следующий запрос (если таковой имеется в наличии), чтобы успеть принять участие в ближайшей процедуре арбитража и чтобы АЦП не тратил такты на переход из активного режима в режим ожидания и обратно.

4 Секвенсор, во время работы, может не выставить следующий запрос после сброса текущего, но лишь в случае ожидания отложенного события перезапуска. Во всех остальных режимах запросы выставляются неразрывно друг за другом.

5 Если несколько секвенсоров выставили одинаковые запросы, то обслуживаться они будут параллельно. Результат запроса попадет в буферы всех соответствующих секвенсоров.

6 Арбитраж происходит перед началом обработки первого запроса (если АЦП был в ожидании) и в конце каждого обрабатываемого в текущий момент.

7 Если несколько секвенсоров одновременно выставят запросы, то запросы по каналам с меньшим номером имеют приоритет выше, чем каналы с большим.

8 Каналы с установленным битом PRIORITY в регистрах CHCTLn (где n – номер канала от 0 до 47) имеют более высокий приоритет, чем те, у которых бит сброшен.

9 Если одновременно будут выставлены запросы по каналам с установленным PRIORITY, то запросы по каналам с меньшим номером имеют приоритет выше, чем каналы с большим.

10 Необходимо с осторожностью производить частый опрос более высокоприоритетных каналов – это может значительно затруднить опрос остальных каналов.

Иллюстрация работы схемы арбитража запросов приведена на рисунке 18.13.

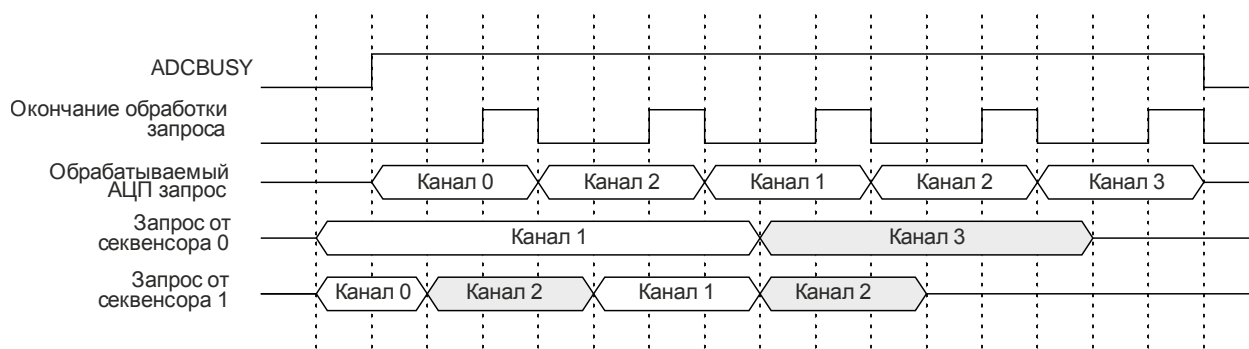


Рисунок 18.13 – Пример арбитража запросов, серым выделены запросы по каналам с установленным битом PRIORITY

### Схема коррекции результатов измерений

Результат преобразования передается на схему коррекции, которая нивелирует ошибку усиления и смещения нуля и работа которой описывается формулой

$$D_C = \frac{D_R \times (D_{MAX} + GAINTRIM)}{D_{MAX}} + OFFTRIM, \quad (18.1)$$

где  $D_R$  – данные, полученные непосредственно с АЦП;

$D_C$  – скорректированные данные, выдаваемые секвенсору;

$D_{MAX}$  – максимальная величина данных в зависимости от разрядности результата;  
 GAINTRIM – коэффициент корректировки усиления;  
 OFFTRIM – коэффициент корректировки смещения нуля.

Значения GAINTRIM и OFFTRIM заносятся в одноименные поля регистров CHCTLn (где n – номер канала от 0 до 47) в дополнительном коде. По умолчанию результат измерения проходит через схему коррекции, не изменяясь, т. к. коэффициенты равны нулю.

Реализована математика «насыщения» – когда значение OFFTRIM отрицательное и больше дроби, то результат будет равен нулю, если сумма OFFTRIM и дроби больше  $D_{MAX} - 1$ , то результат равен  $D_{MAX} - 1$ .

В зависимости от разрядности результата диапазоны коэффициентов коррекции отличаются. Все значения вносятся в дополнительный код и выравниваются по младшему биту.

Таблица 18.2 – Зависимость диапазонов коэффициентов коррекции от разрядности данных

Поле SELRES регистра ACTL	Разрядность данных, бит	Диапазон значений коэффициентов коррекции OFFTRIM, GAINTRIM
00b	6	от -4 до 3
01b	8	от -16 до 15
10b	10	от -64 до 63
11b	12	от -256 до 255

### 18.3 Цифровой компаратор

В состав блока АЦП входят d компараторов (где d от 0 до 23). Структурная схема компаратора показана на рисунке 18.14.

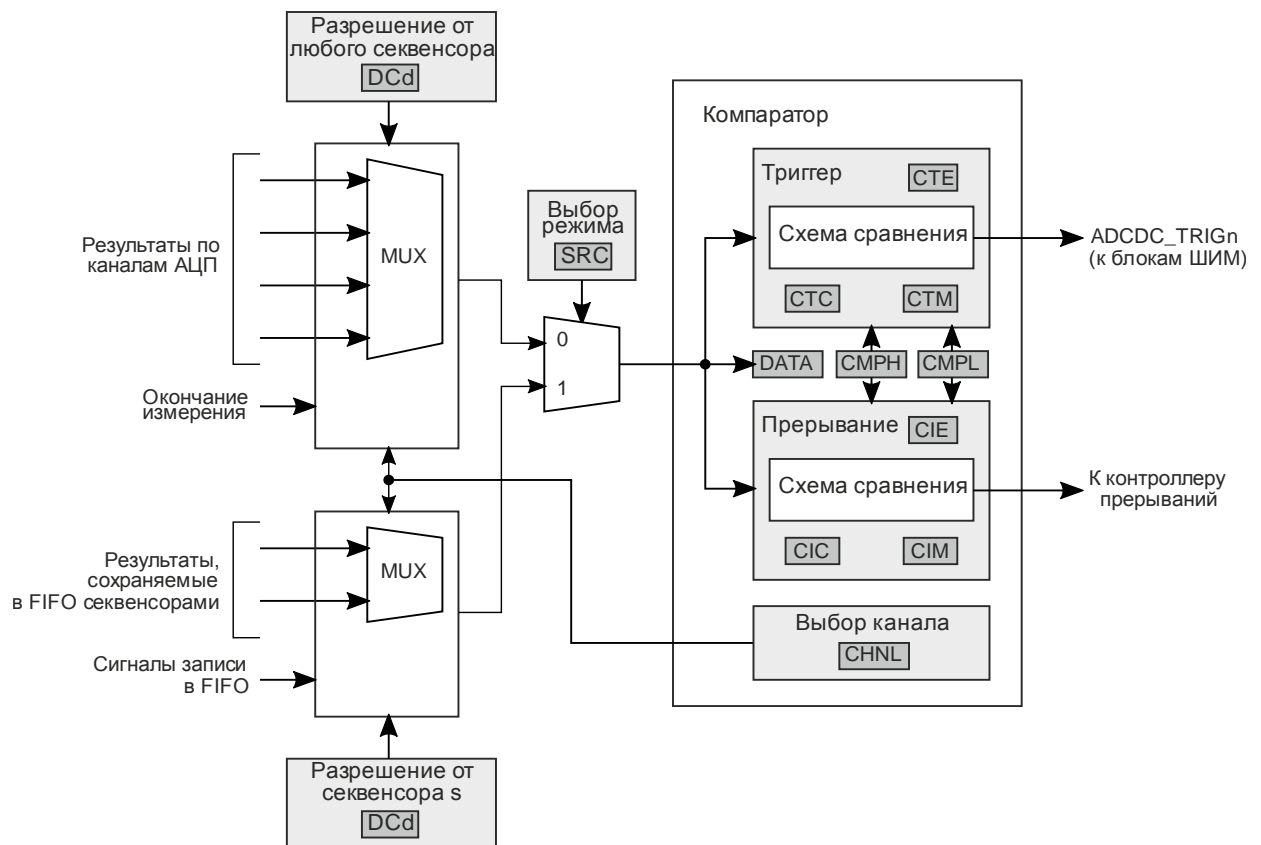


Рисунок 18.14 – Структурная схема компаратора

Все компараторы блока АЦП независимы. Каждый компаратор может обрабатывать результат измерения любого канала.

По умолчанию, когда модуль АЦП завершает обработку запроса по каналу, он выставляет результат, который захватывает как секвенсор, так и разрешенный (любым из секвенсоров) и настроенный на этот канал компаратор.

Возможен и другой режим работы, когда на компаратор будет подаваться результат, который секвенсор записывает в FIFO, но только в том случае, если канал, соответствующий сохраняемому результату, также совпадает с каналом, на который настроен компаратор. Включение этого режима осуществляется установкой бита SRC в регистре DCTL.

Правила настройки цифровых компараторов:

1 Посредством регистра SRQSEL секвенсора  $s$  выбираются каналы для измерений.

2 Посредством регистра SDC выбираются (разрешаются) компараторы для обработки полученных результатов запросов (установкой битов DCd). Запрещенные компараторы не обрабатывают полученные результаты.

3 Для каждого компаратора  $d$  в его регистре DCTL в поле CHNL указывается номер канала, результат измерения которого будет передан на него. По умолчанию значение CHNL = 0h, т. е. все компараторы настроены на работу с нулевым каналом.

4 Битом SRC в регистре DCTL выбирается источник данных для компаратора – результаты непосредственно с АЦП или результаты, записываемые секвенсором в FIFO (которые могут быть уже усреднены).

Результат измерения, полученный компаратором  $d$ , передается во внутреннюю схему сравнения и одновременно с этим сохраняется в регистре DDATA. Схема сравнения выполняет проверку соответствия результата измерения заданному условию (поля CTC, CTM регистра DCTL и поля CMP<sub>L</sub>, CMP<sub>H</sub> регистра DCM<sub>P</sub>) и в зависимости от результата проверки переключает выходной триггер и устанавливает флаг события сравнения DCEV<sub>d</sub> в регистре DCTRIG. Работа триггера разрешается установкой бита CTE регистра DCTL.

#### **Сравнение по условию «Измерение $\leq$ CMP<sub>L</sub>» (CTC = 00b):**

- В однократном режиме (CTM = 01b) выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме (CTM = 00b) выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом (CTM = 11b) выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «CMP<sub>H</sub>  $\leq$  Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (CTM = 10b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «CMP<sub>H</sub>  $\leq$  Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «CMP<sub>H</sub>  $\leq$  Измерение».

Пример функционирования триггера показан на рисунке 18.15.

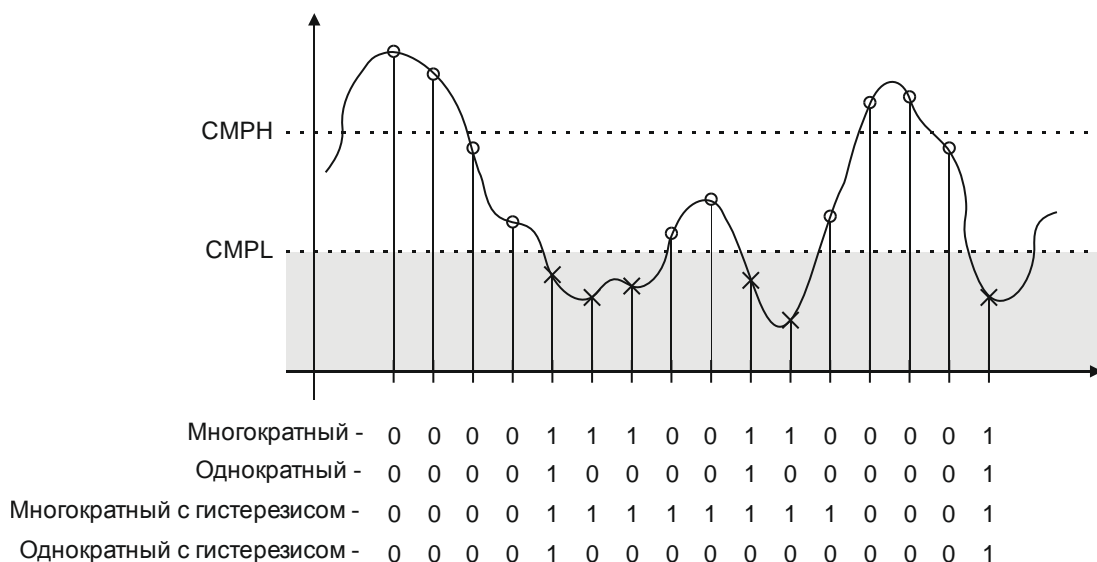


Рисунок 18.15 – Функционирование триггера при CTC = 00b

**Сравнение по условию «CMPL ≤ Измерение ≤ CMPH» (CTC = 01b):**

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- Однократный и многократный режимы с гистерезисом не поддерживаются.

Пример функционирования триггера показан на рисунке 18.16.

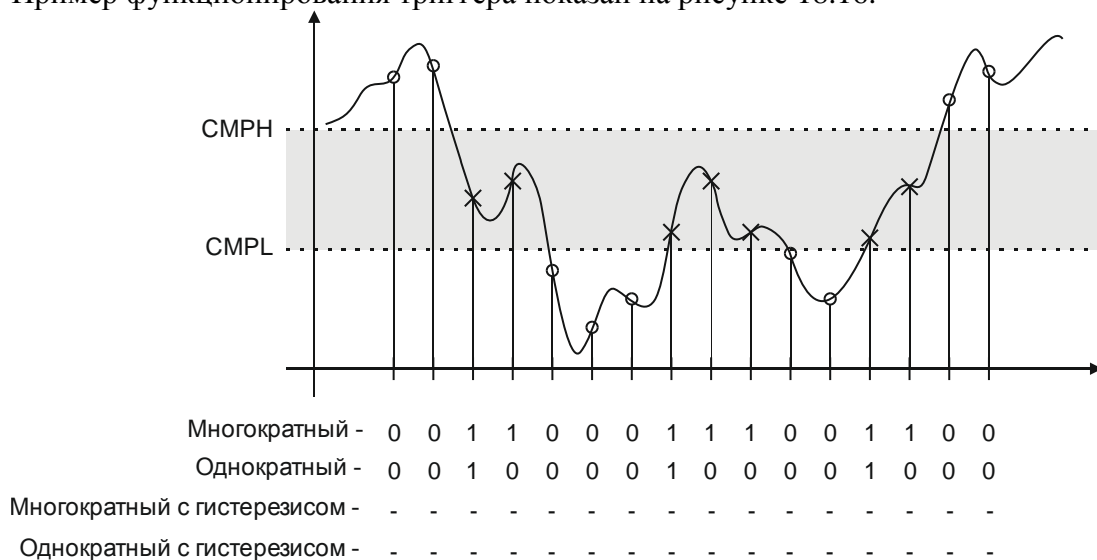


Рисунок 18.16 – Функционирование триггера при CTC = 01b

**Сравнение по условию «CMPH ≤ Измерение» (CTC = 10b):**

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «Измерение ≤ CMPL», результат сравнения окажется положительным при том, что результат предыдущего



сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (СТМ = 10b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «Измерение  $\leq$  CMPL», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «Измерение  $\leq$  CMPL».

Пример функционирования триггера показан на рисунке 18.17.

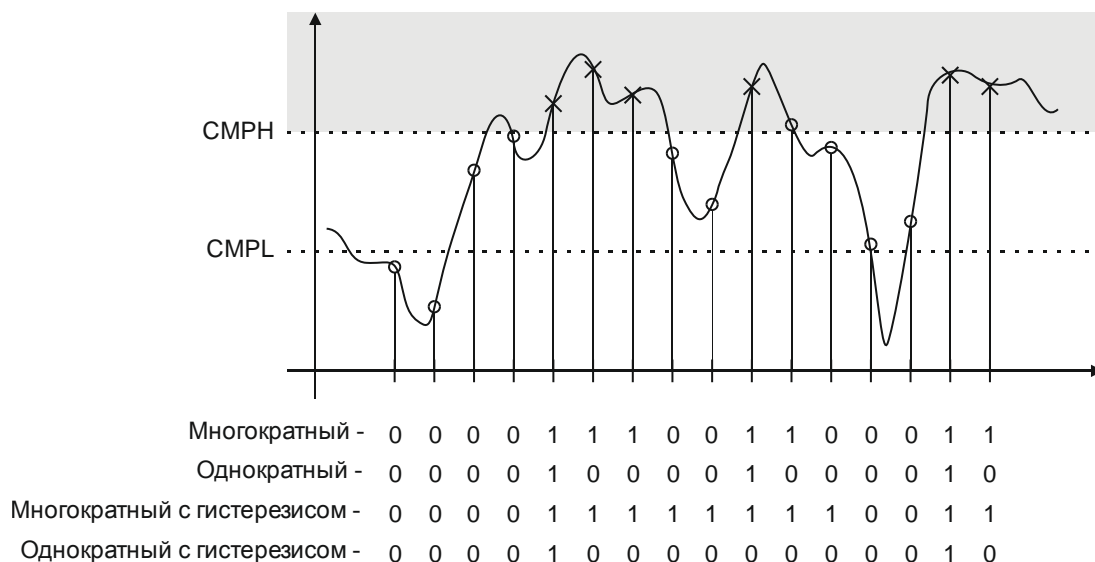


Рисунок 18.17 – Функционирование триггера при СТМ = 10b

Переключение выходного триггера в единицу устанавливает соответствующий флаг TOSd в регистре DCTRIG и генерирует управляющий сигнал для пороговых выключателей блоков ШИМ. Вне зависимости от состояния флагов TOSd по каждому событию сравнения устанавливается соответствующий флаг DCEVd того же регистра. Флаги события сравнения сбрасываются записью единицы. Сброс самого триггера и его статусного флага выполняется также записью единицы в соответствующий бит TOSd регистра DCTRIG.

Независимо от состояния триггера (разрешен или запрещен) в случае положительного результата сравнения компаратор может генерировать прерывание. Для этого следует установить бит CIE регистра DCTL и задать условия CIC и CIM (аналогичны по функционалу СТМ и СТМ).

Примечание – Условия срабатывания выходного триггера компаратора и условия генерирования прерываний могут не совпадать.

## 18.4 Прерывания

При генерировании прерываний устанавливаются флаги SEQRISs и DCRISd в регистре RIS. Если были установлены маски прерываний в регистре IM (поля SEQIMs и DCIMd), то также устанавливаются соответствующие маскированные флаги прерываний SEQMISs и DCMISd. Сброс флагов (маскированных и немаскированных) осуществляется записью единицы в соответствующие поля регистра IC (поля SEQICs и DCICd).

Установка маскированных флагов SEQMISs вызывает формирование соответствующих прерываний ADC\_SEQs блока АЦП.

Флаги DCMISd компараторов объединены по ИЛИ, и установка любого из них вызывает формирование прерывания ADC\_DC блока АЦП.

Структурная схема контроллера прерываний показана на рисунке 18.18.

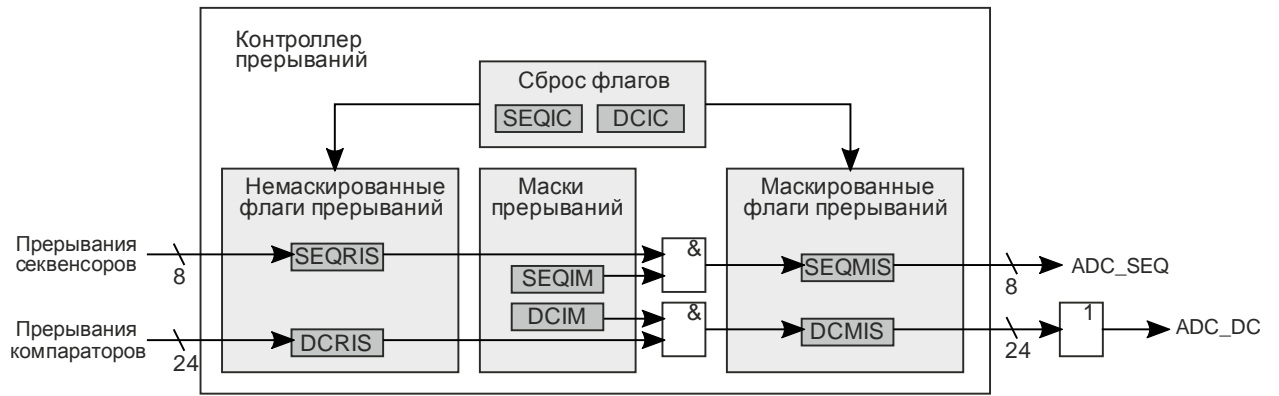


Рисунок 18.18 – Контроллер прерываний

## 18.5 Примеры работы блока АЦП

В этом подразделе представлены разнообразные примеры настройки блока АЦП для осуществления измерений в различных режимах. Для всех примеров подразумевается, что АЦП тактируется частотой в 33,3 МГц.

### Включение LDO

Каждый модуль АЦП имеет внутренний LDO, который должен быть включен в первую очередь.

1 Для включения LDO внутри АЦП необходимо установить единицу в соответствующее поле LDOENm регистра PMU - ADCPC.

2 Дожидаемся установки соответствующих флагов готовности LDORDYm в том же регистре.

```
PMU->ADCPC_bit.LDOEN0 = 1;
PMU->ADCPC_bit.LDOEN1 = 1;
PMU->ADCPC_bit.LDOEN2 = 1;
PMU->ADCPC_bit.LDOEN3 = 1;
//ждем готовности LDO в АЦП
while (!(PMU->ADCPC_bit.LDORDY0 &&
        PMU->ADCPC_bit.LDORDY1 &&
        PMU->ADCPC_bit.LDORDY2 &&
        PMU->ADCPC_bit.LDORDY3));
```

### Настройка тактирования

Один из примеров настройки тактирования АЦП совместно с настройкой системного тактового сигнала представлен ниже.

1 С помощью регистра PLLCFG блока RCU настраиваем выходную частоту PLL 200 МГц. Осуществляем процедуру перевода системной частоты на источник PLL. Таким образом, частота системного тактового сигнала SYSCLK будет равна 200 МГц.

2 Настройка рабочего тактового сигнала АЦП ACLK производится с помощью регистра ADCCFG блока RCU. Выбираем в качестве источника выходную частоту PLL (поле CLKSEL = 1), включаем делитель на 6 (поле DIVN=2, бит DIVEN=1). Таким образом, f<sub>ACLK</sub> = 33,3 МГц.

```
RCU->ADCCFG_bit.CLKSEL = 1;
RCU->ADCCFG_bit.DIVN = 2;
RCU->ADCCFG_bit.DIVEN = 1;
```

3 Включаем тактирование блока (бит CLKEN=1) и снимаем сброс (бит RSTDIS=1).

Блок АЦП готов к дальнейшим конфигурациям.

```
RCU->ADCCFG_bit.CLKEN = 1;
RCU->ADCCFG_bit.RSTDIS = 1;
```

### Пример 1 – Программный запуск одного секвенсора

Требуемый режим работы:

- программный запуск;
- секвенсор 0
- однократное измерение каналов 0-3;
- без прерываний (опрос флагов).

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
//Настройка модуля АЦП – 12 бит и калибровка при включении
ADC->ADCCTL[0].ADCCTL_bit.SELRES = 3;
ADC->ADCCTL[0].ADCCTL_bit.CALEN = 1;
```

```

ADC->ADCCTL[0].ADCCTL_bit.ADCEN = 1;
//Настройка секвенсора 0
ADC->EMUX_bit.EM0 = 0;
ADC->SEQ[0].RQCTL_bit.RQMAX = 0x3;
ADC->SEQ[0].RQSEL0_bit.RQ0 = 0x0;
ADC->SEQ[0].RQSEL0_bit.RQ1 = 0x1;
ADC->SEQ[0].RQSEL0_bit.RQ2 = 0x2;
ADC->SEQ[0].RQSEL0_bit.RQ3 = 0x3;
ADC->SEQEN_bit.SEQEN0 = 1;
// Запуск
while(!ADC->ADCCTL[0].ADCCTL_bit.ADCRDY);
ADC->SEQSYNC_bit.SYNC0 = 1;
ADC->SEQSYNC_bit.GSYNC = 1;

```

1 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ADCCTL.

2 Для работы выбран секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно 0, т. к. запуск программный.

3 В поле RQMAX регистра SRQCTL необходимо внести значение 3h, т. к. измерения будут проводиться по четырем каналам.

4 Настраиваем каналы для запросов. Допустим, необходимо опросить каналы последовательно от нулевого к третьему, значит, в регистр SRQSEL0 необходимо внести значения в поля: RQ0=0h, RQ1=1h, RQ2=2h, RQ3=3h.

5 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

6 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы начать измерения, необходимо установить бит SYNC0 в регистре SEQSYNC и записать в бит GSYNC единицу.

7 Проводим опрос флагов, чтобы установить окончание измерений. Например, можно опрашивать регистр SFLOAD, ожидая, пока он не станет равен 4h.

8 Считываем четыре результата измерения из буфера SFIFO.

Работа режима проиллюстрирована на рисунке 18.19.

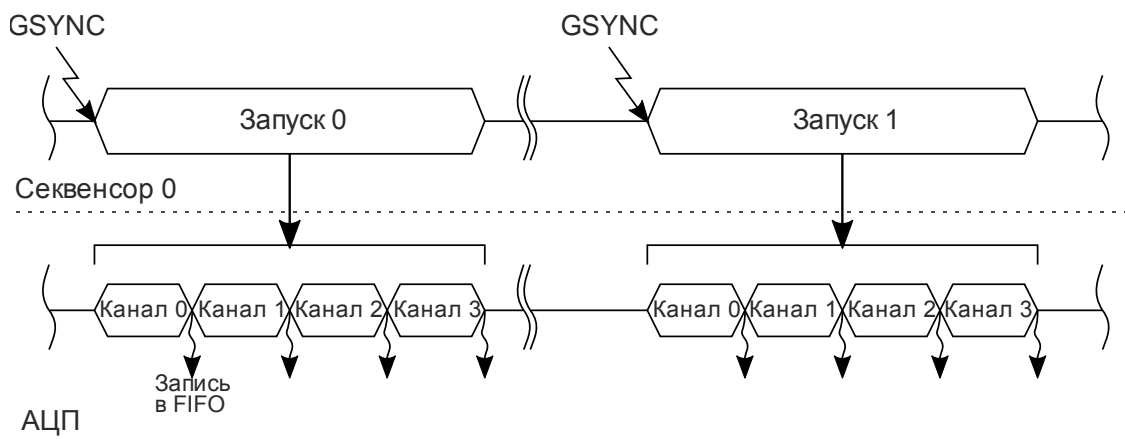


Рисунок 18.19 – Диаграммы программного запуска одного секвенсора

## Пример 2 – Циклический опрос канала с задержками

Требуемый режим работы:

- циклическая работа;
- секвенсор 0
- опрос канала номер 2 каждую 1мс;
- коррекция канала 2;
- каждый опрос из 16 последовательных измерений и усреднения;
- прерывание по каждой записи в FIFO.

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
//Настройка модуля АЦП – 12 бит и калибровка при включении
ADC->ADCCTL[0].ADCCTL_bit.SELRES = 3;
ADC->ADCCTL[0].ADCCTL_bit.CALEN = 1;
ADC->ADCCTL[0].ADCCTL_bit.ADCEN = 1;
//Коррекция канала 2
ADC->CHCTL[2].CHCTL_bit.GAINTRIM = 5; // значения для примера
ADC->CHCTL[2].CHCTL_bit.OFFTRIM = (uint32_t) (-5);
//Настройка секвенсора 0
ADC->EMUX_bit.EM0 = 0xF;
ADC->SEQ[0].CCTL_bit.ICNT = 0;
ADC->SEQ[0].RTMR = 33332;
ADC->SEQ[0].RQCTL_bit.RQMAX = 0;
ADC->SEQ[0].RQCTL_bit.QAVGVAL = 4;
ADC->SEQ[0].RQCTL_bit.QAVGEN = 1;
ADC->SEQ[0].RQSEL0_bit.RQ0 = 2;
ADC->SEQEN_bit.SEQEN0 = 1;
// NVIC прерывание
ADC->IM_bit.SEQIM0 = 1;
NVIC_EnableIRQ(ADC_SEQ0_IRQn);
// Запуск
while(!ADC->ADCCTL_bit.ADCRDY);
ADC->SEQSYNC_bit.SYNC0 = 1;
ADC->SEQSYNC_bit.GSYNC = 1;
```

1 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ADCCTL.

2 Если предварительно была произведена процедура коррекции канала 2 и были вычислены поправочные коэффициенты, их необходимо внести в поля GAINTRIM и OFFTRIM регистра CHCTL2.

3 Для работы выберем секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно Fh, т. к. запуск циклический.

4 В поле RQMAX регистра SRQCTL необходимо внести значение 0h, т. к. измерения будут проводиться по одному каналу.

5 Настраиваем канал для запроса. В регистр SRQSEL0 необходимо внести значение RQ0=2h.

6 Включаем усреднение сканированием по 16 опросам очереди. В регистре SEQ0\_RQCTL нужно установить поля QAVGVAL=4, QAVGEN=1.

7 Для того чтобы опрос проводился каждую 1 мс (при  $f_{ACLK} = 33,3$  МГц), необходимо внести в регистр SEQ0\_RTMR значение  $(1000000 \text{ нс}/30 \text{ нс}) - 1 = 33332$ .

8 Разрешаем генерацию прерываний по каждой записи в FIFO – нужно установить бит SEQIM0 в регистре IM и проследить, чтобы поле ICNT регистра SCCTL оставалось

равным нулю.

9 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

10 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы начать измерения, необходимо установить бит SYNC0 в регистре SEQSYNC и записать в бит GSYNC единицу.

11 Измерения будут сразу же запущены и будут запускаться каждую 1 мс далее. После каждого запуска будет проведено 16 измерений, результат усреднения которых будет записан в FIFO, и будет вызвано прерывание.

Работа режима проиллюстрирована на рисунке 18.20.

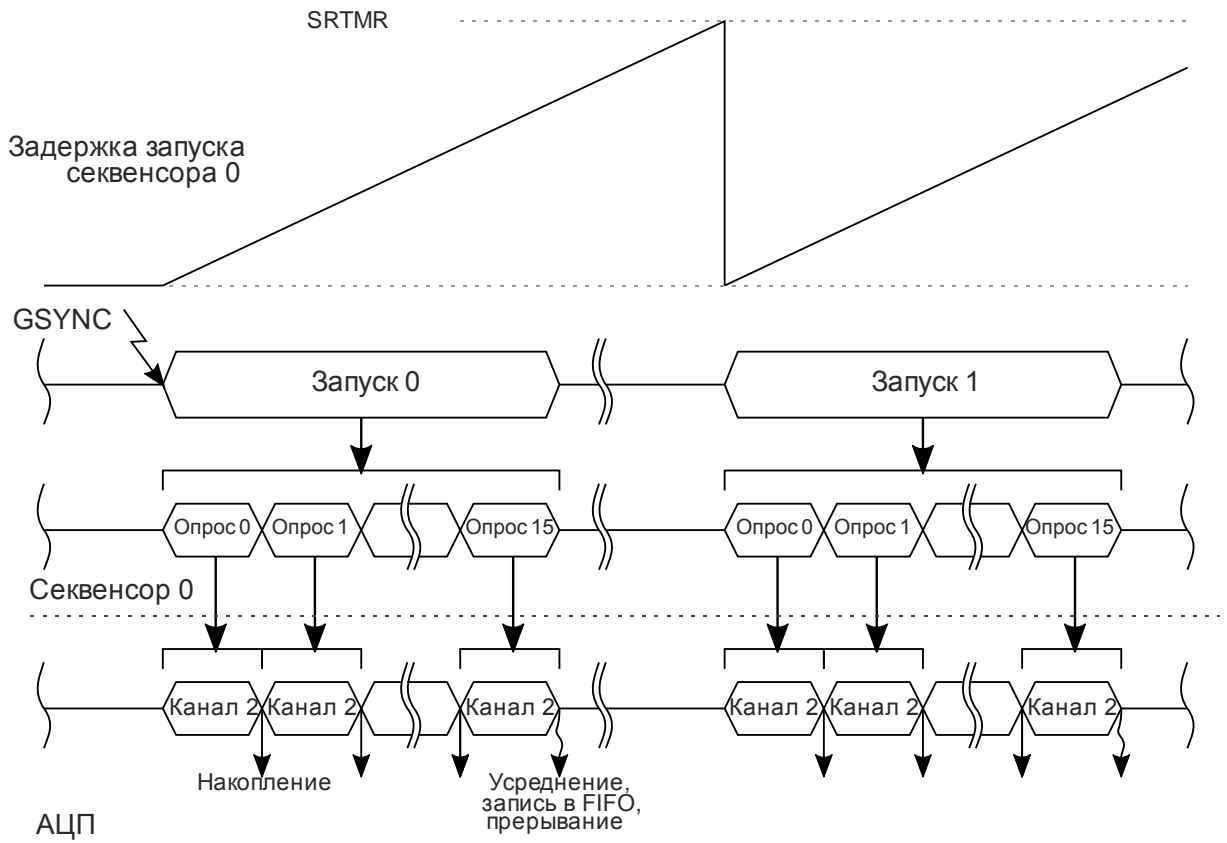


Рисунок 18.20 – Диаграммы циклического опроса канала

### Пример 3 – Равномерно распределенные измерения по периоду таймера

Требуемый режим работы:

- запуск по таймеру 0 с частотой 10 кГц;
- четыре точки измерений равномерно распределенные по периоду;
- секвенсор 0;
- опросы по каналам 3 и 0;
- каждый опрос из четырех последовательных измерений и усреднения;
- прерывание каждые восемь записей в FIFO (по окончании измерений в текущем периоде таймера).

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
// Настройка таймера
RCU->APBCFG_bit.DIV = 1;
RCU->PCLKCFG0_bit.TMR0EN = 1;
RCU->PRSTCFG0_bit.TMR0EN = 1;
TMR0->LOAD = 9599;
```

```

TMR0->ADCSOC_bit.EN = 1;
//Настройка модуля АЦП - 12 бит и калибровка при включении
ADC->ADCCTL[0].ADCCTL_bit.SELRES = 3;
ADC->ADCCTL[0].ADCCTL_bit.CALEN = 1;
ADC->ADCCTL[0].ADCCTL_bit.ADCEN = 1;
//Настройка секвенсора
ADC->EMUX_bit.EM0 = 5;
ADC->SEQ[0].CCTL_bit.ICNT = 7;
ADC->SEQ[0].CCTL_bit.RCNT = 3;
ADC->SEQ[0].RTMR = 799;
ADC->SEQ[0].RQCTL_bit.RQMAX = 1;
ADC->SEQ[0].RQCTL_bit.QAVGVAL = 2;
ADC->SEQ[0].RQCTL_bit.QAVGEN = 1;
ADC->SEQ[0].RQSEL0_bit.RQ0 = 3;
ADC->SEQ[0].RQSEL0_bit.RQ1 = 0;
ADC->SEQEN_bit.SEQEN0 = 1;
// NVIC прерывание
ADC->IM_bit.SEQIM0 = 1;
NVIC_EnableIRQ(ADC_SEQ0_IRQn);
// Запуск
while(!ADC->ADCCTL_bit.ADCRDY);
TMR0->CTRL_bit.ON = 1;

```

1 Включаем делитель на 2 для APB тактового сигнала –  $f_{CLK}=100$  МГц, установив бит DIV в регистре RCU APBCFG. Включаем тактирование таймера 0 и снимаем сброс – установим бит TMR0EN в регистрах PCLKCFG0 и PRSTCFG0 соответственно.

2 Для того чтобы таймер опустошался с частотой 10 кГц (при  $f_{CLK}=100$  МГц), необходимо внести в регистр таймера LOAD значение  $(100000 \text{ кГц}/10 \text{ кГц}) - 1 = 9999$ .

3 Разрешаем генерацию таймером запросов на старт преобразования, установив бит EN в регистре ADCSOC таймера.

4 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений ( $SELRES=3$ ) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ADCCTL.

5 Для работы выберем секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно 5h, т. к. запуск по опустошению таймера 0.

6 Необходимое количество записей в FIFO для генерации прерывания – 8h, поэтому запишем в поле ICNT регистра SEQ0\_CCTL значение  $8 - 1 = 7h$ .

7 По каждому запуску должно совершиться три перезапуска с паузой в четверть периода опустошения таймера 0. В поле количества перезапусков RCNT регистра SCCTL внесем 3h.

В регистр задержки перезапуска SRTMR внесем значение  $33333 \text{ кГц}/(10 \text{ кГц} \times 4) - 1 = 832$ .

8 В поле RQMAX регистра SRQCTL необходимо внести значение 1h, т. к. измерения будут проводиться по двум каналам.

9 Настраиваем каналы для запроса. В регистр SRQSEL необходимо внести значения  $RQ0=3h$ ,  $RQ1=0h$ .

10 Включаем усреднение сканированием по четырем опросам очереди. В регистре SRQCTL нужно установить поля  $QAVGVAL=2$ ,  $QAVGEN=1$ .

11 Разрешаем генерацию прерываний после каждой восьмой записи в FIFO – нужно установить бит SEQIM0 в регистре IM.

12 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

13 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы

их начать, необходимо включить таймер 0, установив бит ON в регистре CTRL таймера 0.

14 Измерения будут запускаться по каждому опустошению таймера 0. После каждого запуска будет проведено по три отложенных перезапуска. Т. к. после каждого перезапуска в FIFO будет попадать по 2 усредненных результата, то прерывание будет сгенерировано после записи последнего результата в последнем третьем перезапуске.

Работа режима проиллюстрирована на рисунке 18.21.

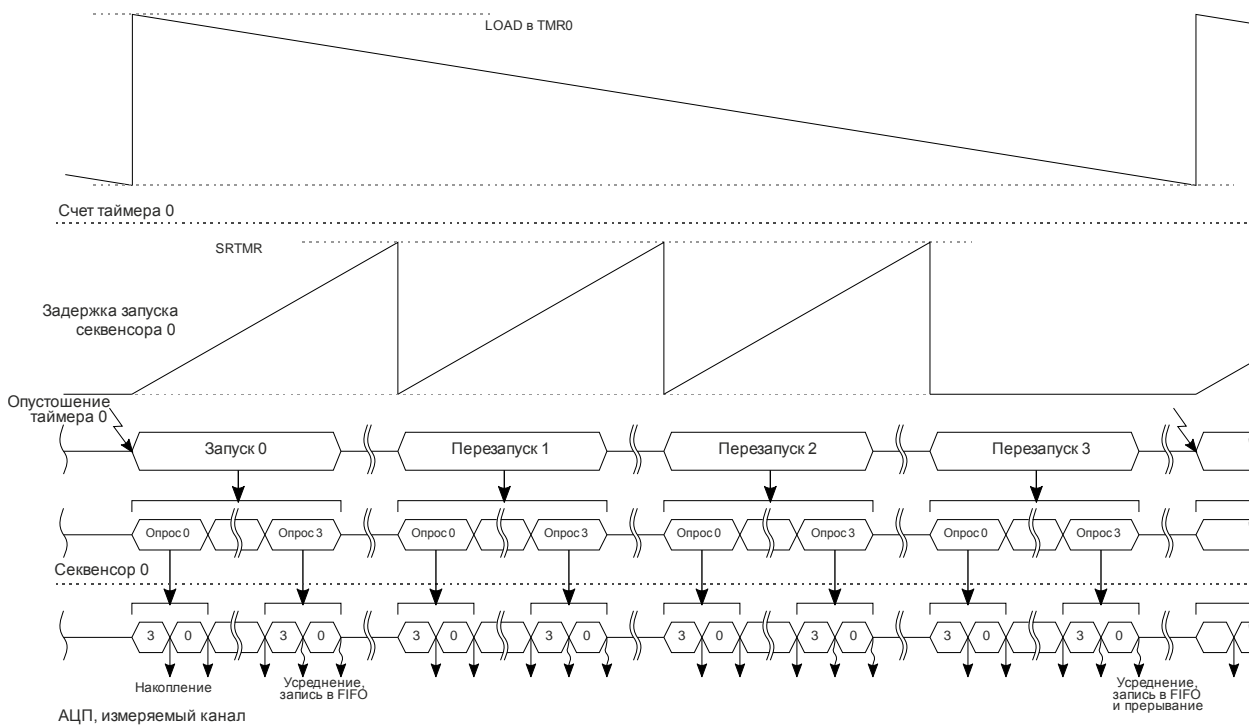


Рисунок 18.21 – Диаграммы равномерно распределенных измерений по периоду таймера

#### Пример 4 – Запуск нескольких секвенсоров

Требуемый режим работы:

- таймер 0 опустошается с частотой 10 кГц;
- ШИМ0 считает в двунаправленном режиме, достигает нуля с частотой 10 кГц;
- секвенсор 0 – запускается по ШИМ;
- секвенсор 0 – усреднение результатов по четырем точкам измерений, равномерно распределенным по периоду;
- секвенсор 0 – измерения по каналам 0, 1, каналы имеют повышенный приоритет;
- секвенсор 0 – каждое измерение из двух последовательных опросов и усреднения;
- секвенсор 0 – прерывание каждые две записи в FIFO (по окончании измерений в текущем периоде);
- секвенсор 1 – запускается по таймеру;
- секвенсор 1 – измерение канала 2;
- секвенсор 1 – каждое измерение из четырех последовательных опросов и усреднения;
- секвенсор 1 – прерывание каждую запись в FIFO.

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
RCU->APBCFG_bit.DIV = 1;
// Настройка таймера
RCU->PCLKCFG0_bit.TMR0EN = 1;
RCU->PRSTCFG0_bit.TMR0EN = 1;
TMR0->LOAD = 9999;
TMR0->ADCSOC_bit.EN = 1;
```



```

//Инициализация ШИМ
RCU->PCLKCFG0_bit.PWM0EN = 1;
RCU->PRSTCFG0_bit.PWM0EN = 1;
PWM0->TBCTL_bit.CLKDIV = 1;
PWM0->TBCTL_bit.HSPCLKDIV = 5;
PWM0->TBPRD = 250;
PWM0->TBCTL_bit.CTRMODE = 2;
PWM0->ETSEL_bit.SOCASEL = 1;
PWM0->ETSEL_bit.SOCAEN = 1;
//Настройка модуля АЦП - 12 бит и калибровка при включении
ADC->ADCCTL[0].ADCCTL_bit.SELRES = 3;
ADC->ADCCTL[0].ADCCTL_bit.CALEN = 1;
ADC->ADCCTL[0].ADCCTL_bit.ADCEN = 1;
ADC->CHCTL[0].CHCTL_bit.PRIORITY = 1;
ADC->CHCTL[1].CHCTL_bit.PRIORITY = 1;
//Настройка секвенсора 0
ADC->EMUX_bit.EM0 = 9;
ADC->SEQ[0].CCTL_bit.ICNT = 1;
ADC->SEQ[0].CCTL_bit.RCNT = 3;
ADC->SEQ[0].CCTL_bit.RAVGEN = 1;
ADC->SEQ[0].RTMR = 832;
ADC->SEQ[0].RQCTL_bit.RQMAX = 1;
ADC->SEQ[0].RQCTL_bit.QAVGVAL = 1;
ADC->SEQ[0].RQCTL_bit.QAVGEN = 1;
ADC->SEQ[0].RQSEL0_bit.RQ0 = 0;
ADC->SEQ[0].RQSEL0_bit.RQ1 = 1;
ADC->SEQEN_bit.SEQEN0 = 1;
ADC->IM_bit.SEQIM0 = 1;
NVIC_EnableIRQ(ADC_SEQ0_IRQn);
//Настройка секвенсора 1
ADC->EMUX_bit.EM1 = 5;
ADC->SEQ[1].RQCTL_bit.RQMAX = 0;
ADC->SEQ[1].RQCTL_bit.QAVGVAL = 2;
ADC->SEQ[1].RQCTL_bit.QAVGEN = 1;
ADC->SEQ[1].RQSEL0_bit.RQ0 = 2;
ADC->SEQEN_bit.SEQEN1 = 1;
ADC->IM_bit.SEQIM1 = 1;
NVIC_EnableIRQ(ADC_SEQ1_IRQn);
// Запуск
while(!ADC->ADCCTL_bit.ADCRDY);
TMR0->CTRL_bit.ON = 1;
SIU->PWMSYNC_bit.PRESCRST = 1<<0;

```

1 Включаем делитель на 2 для APB тактового сигнала –  $f_{PCLK}=100$  МГц, установив бит DIV в регистре RCU APBCFG. Включаем тактирование таймера 0 и снимаем сброс – установим бит TMR0EN в регистрах PCLKCFG0 и PRSTCFG0 соответственно.

2 Для того чтобы таймер опустошался с частотой 10 кГц (при  $f_{PCLK}=100$  МГц), необходимо внести в регистр таймера LOAD значение  $(100000 \text{ кГц}/10 \text{ кГц}) - 1 = 9999$ .

3 Разрешаем генерацию таймером запросов на старт преобразования, установив бит EN в регистре ADCSOC таймера.

4 Включаем тактирование ШИМО и снимаем сброс – установим бит PWM0EN в регистрах PCLKCFG0 и PRSTCFG0 соответственно.

5 Настроим делители тактового сигнала ШИМ, таким образом, чтобы частота счета TBCLK была равна 5 МГц (при  $f_{PCLK}=100$  МГц). Для этого запишем в поля регистра ШИМ

ТВCTL значения CLKDIV=1 (коэффициент 1/2), HSPCLKDIV=5 (коэффициент 1/10).

6 Режим счета двусторонний – поле STRMODE=2 регистра ТВCTL, а период счета (регистр ТВPRD) соответственно равен  $5000 \text{ кГц}/(10 \text{ кГц})/2 = 250$ .

7 Разрешаем генерацию ШИМ0 запросов по каналу А на старт преобразования, установив бит SOCAEN в регистре ETSEL и записав единицу в поле ETSEL того же регистра.

8 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ADCCTL.

9 Устанавливаем высокий приоритет для каналов, которые будут обрабатываться секвенсором 0 – каналов 0, 1, установив бит PRIRORITY в соответствующих регистрах СНCTL.

10 Проинициализируем секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно 9h, т. к. запуск по сигналу канала А ШИМ0.

11 Необходимое количество записей в FIFO для генерации прерывания – 2, поэтому запишем в поле ICNT регистра SEQ0->SCCTL значение 2 – 1=1h.

12 По каждому запуску должно совершиться три перезапуска с паузой в четверть периода ШИМ. В поле количества перезапусков RCNT регистра SEQ0->SCCTL внесем 3h. В регистр задержки перезапуска SEQ0->SRTMR внесем значение  $33333 \text{ кГц}/(10 \text{ кГц}*4) - 1 = 832$ .

13 В поле RQMAX регистра SEQ0->SRQCTL необходимо внести значение 1h, т. к. измерения будут проводиться по двум каналам.

14 Настраиваем каналы для запроса. В регистр SEQ0->SRQSEL необходимо внести значения RQ0=0h, RQ1=1h.

15 Включаем усреднение сканированием по 2 опросам очереди. В регистре SEQ0->SRQCTL нужно установить поля QAVGVAL=1, QAVGEN=1.

16 Разрешаем генерацию прерываний – нужно установить бит SEQIM0 в регистре IM.

17 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

18 Проинициализируем секвенсор 1. Настроим его источник запуска – поле EM1 регистра EMUX должно быть равно 5h, т. к. запуск по сигналу таймера 0.

19 В поле RQMAX регистра SEQ1->SRQCTL необходимо внести значение 0h, т. к. измерения будут проводиться по одному каналу.

20 Настраиваем каналы для запроса. В регистр SEQ1->SRQSEL0 необходимо внести значение RQ0=2h.

21 Включаем усреднение сканированием по четырем опросам очереди. В регистре SEQ1->RQCTL нужно установить поля QAVGVAL=2, QAVGEN=1.

22 Разрешаем генерацию прерываний – нужно установить бит SEQIM1 в регистре IM.

23 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN1 в регистре SEQEN.

24 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы их начать, необходимо включить таймер 0, установив бит ON в регистре CTRL таймера 0. А также разрешить работу предделителя ШИМ, записав 1h в поле PRESCRST регистра PWMSYNC блока SIU.

25 Измерения секвенсора 0 будут запускаться по каждому равенству счетного регистра ШИМ0 нуля. После каждого запуска будет проведено по три отложенных перезапуска. Т. к. дополнительно включено усреднение по перезапускам, то два усреднённых результата попадут в FIFO в последнем третьем перезапуске и будет вызвано прерывание. Измерения секвенсора 1 будут производиться по каждому опустошению таймера 0. После усреднения сканированием результат будет записан в FIFO и будет вызвано прерывание.

Работа режима проиллюстрирована на рисунке 18.22.

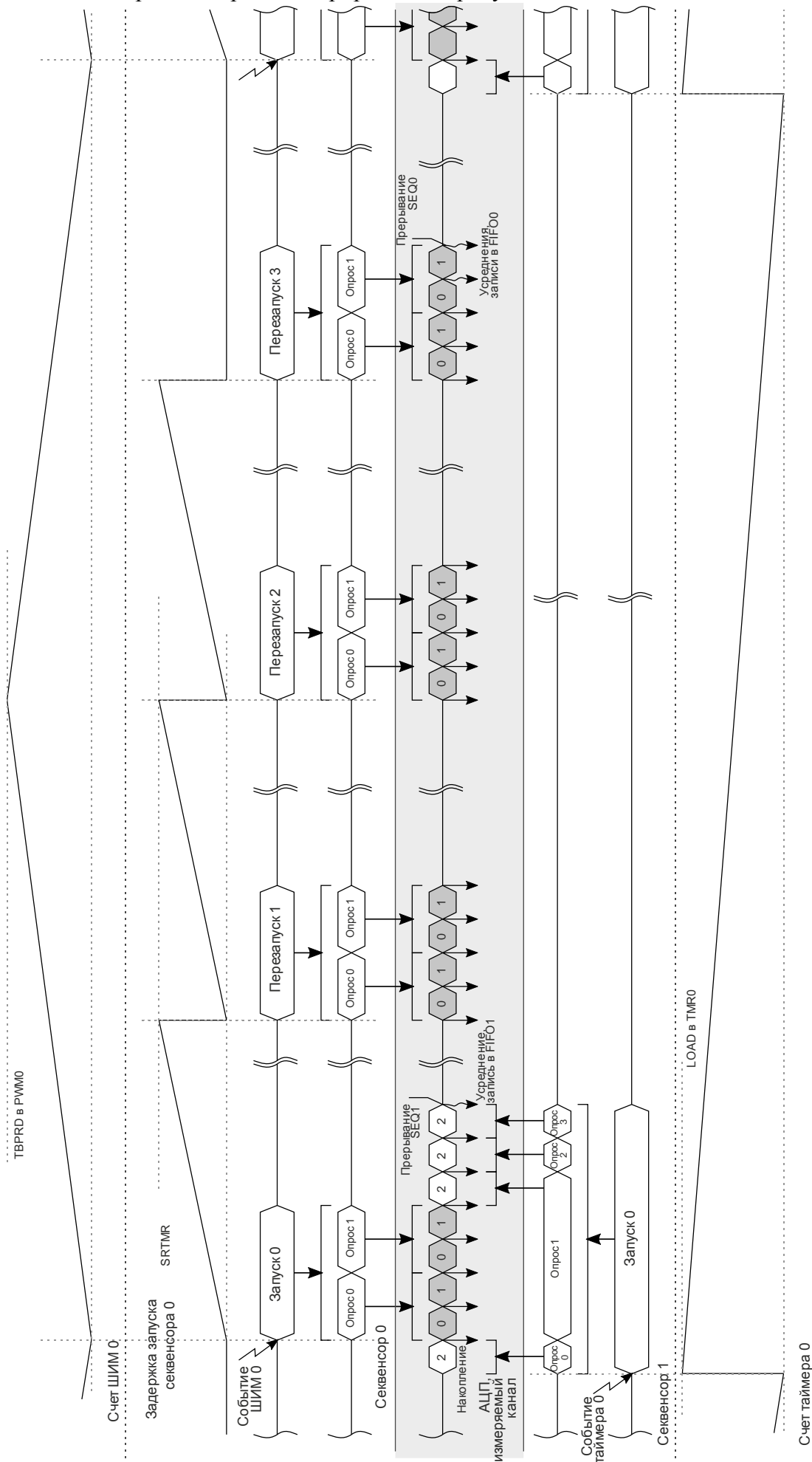


Рисунок 18.22 – Диаграммы работы примера 4

## 19 Блоки захвата

В микроконтроллере реализованы шесть блоков захвата. Все блоки идентичны.

Блоки захвата используются для:

- вычисления скорости вращения вала ротора (с использованием датчиков Холла);
- вычисления промежутков времени между срабатыванием позиционных датчиков;
- вычисления периода и скважности импульсов.

Возможности блока захвата:

- 32-разрядный таймер, с разрешающей способностью 10 нс (на 100 МГц);
- четыре 32-разрядных регистра захвата времени;
- выбор полярности фронта для обработки каждого из четырех последовательных событий;
- источники прерываний по каждому из четырех событий;
- однократный захват значений времени до четырех событий;
- режим циклической работы по событиям, с переписыванием значений (кольцевой буфер);
- режимы захвата абсолютного и относительного значений времени;
- альтернативный режим работы, если не задействована функция захвата времени – одноканальный выход ШИМ.

Функциональная схема блока захвата представлена на рисунке 19.1.

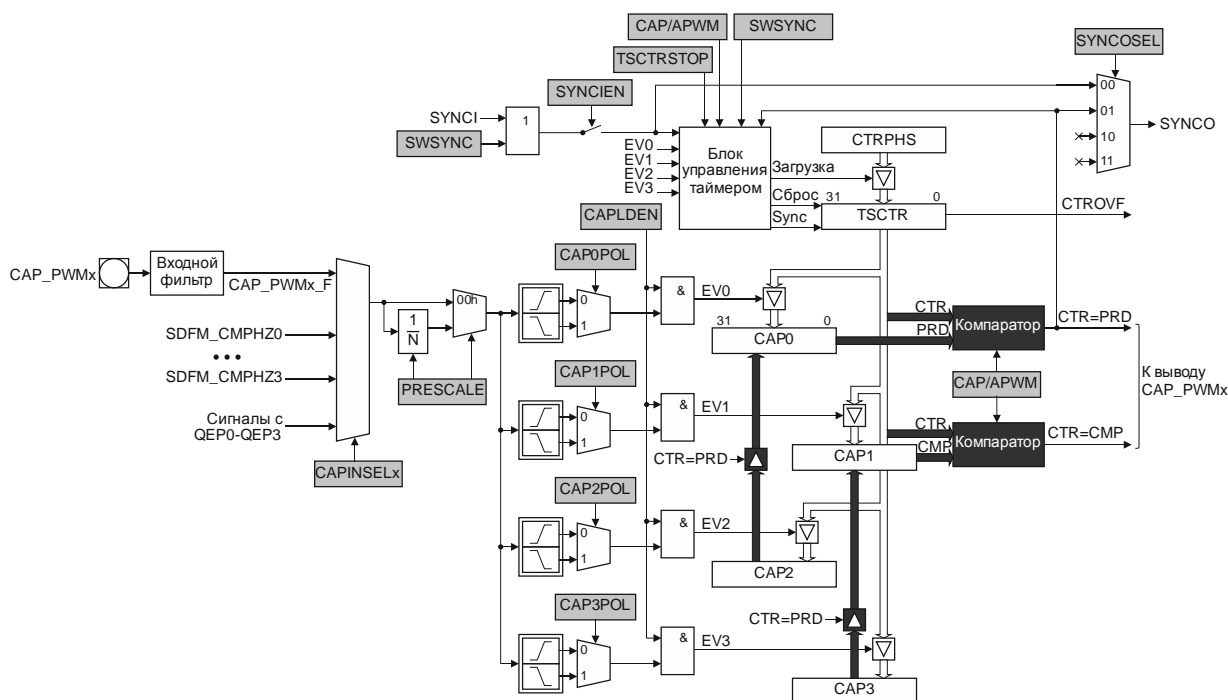


Рисунок 19.1 – Функциональная схема блока захвата

Для начала работы с блоками захвата необходимо подать тактирование – установить соответствующие биты в регистре PCLKCFG блока RCU и снять сброс – установить соответствующие биты регистра PRSTCFG.

Каждый блок захвата ECAPx имеет один вывод CAP\_PWMx (где x от 0 до 5), соединенный с выводом микроконтроллера (альтернативная функция). В зависимости от режима работы блока захвата вывод является входом захвата внешнего события или выходом генерируемого сигнала ШИМ. Вход захвата пропускается через дополнительный входной фильтр (настраивается регистрами QCQUALCTL и QCQUALSAMPLE блока SIU).

Дополнительно, блок захвата может использовать в качестве входов сигналы от других блоков (поле CAPINSELx регистра CAPMUX блока SIU).

## 19.1 Режим захвата времени

Режим захвата времени выбран по умолчанию. Вывод CAP\_PWMx функционирует как вход.

### Предварительный делитель

В случае если события на входе CAP\_PWMx приходят слишком часто и требуется уменьшить их частоту, используется предварительный делитель событий (импульсов), состоящий из собственно делителя и мультиплексора. В предварительном делителе используется счетчик, который производит выборку одного события из каждых 2 – 63 входных. Значение делителя задается полем PRESCALE регистра ECCTL0. В случае если задано значение 00h, то делитель выключен и входной сигнал поступает на детекторы фронта напрямую.

Для примера на рисунке 19.2 показаны несколько вариантов сигналов на выходе делителя, в зависимости от заданного значения N.

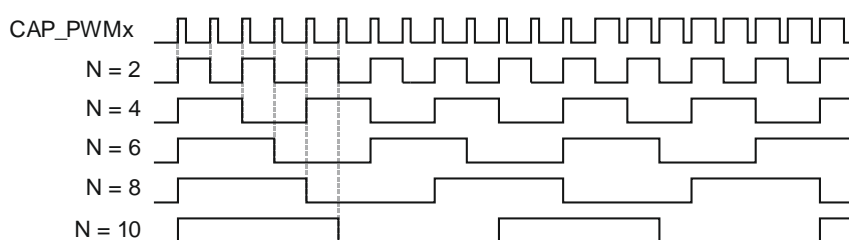


Рисунок 19.2 – Формы сигналов на выходе делителя в зависимости от значения N

### Захват значения таймера

С выхода делителя сигнал поступает на четыре детектора фронта, каждый из которых управляется соответствующим битом CAPnPOL. Далее, если установлен бит CAPLDEN и обнаружен заданный фронт сигнала, формируется соответствующее событие. Возможно одновременное формирование до четырех событий (EV0 – EV3), и по переднему фронту каждого события происходит захват значения таймера TSCTR в соответствующий регистр захвата (CAP0 – CAP3).

Регистр захвата перезаписывается новым захваченным значением каждый раз при возникновении соответствующего события.

### Однократный захват

Однократный захват выбирается битом CONTOST регистра ECCTL1 и включается записью единицы в бит REARM с последующей установкой бита CAPLDEN (установится аппаратно). В этом режиме происходит запуск двухразрядного счетчика событий EV0 – EV3. Количество подсчитываемых событий от одного до четырех задается полем STOPWRAP. Подсчитывается каждое из сформированных событий и одновременно происходит захват значения таймера в соответствующие регистры захвата. Как только количество событий совпадет со значением STOPWRAP, события EV0 – EV3 больше не формируются, значение таймера захватывается регистрами CAP0 – CAP3, и далее регистры не перезаписываются.

Для повторного запуска следует записать единицу в бит REARM (это обнулит счетчик и включит режим), после чего – разрешить формирование событий EV0 – EV3 установкой бита CAPLDEN.

### Циклический захват

Циклический захват выбран по умолчанию. После установки бита CAPLDEN начинается обработка событий EV0 – EV3 и захват значения таймера. Количество подсчитываемых событий от одного до четырех задается полем STOPWRAP.

Регистр захвата перезаписывается новым захваченным значением каждый раз при

возникновении соответствующего события.

Примечание – В обоих режимах захвата значение поля STOPWRAP не оказывает никакого влияния на счетчик и состояние бита CAPLDEN.

### Таймер

Таймер представляет собой 32-разрядный счетчик, работающий на системной частоте. Контроль работы таймера осуществляет блок управления таймером. Счетчик таймера включается битом TSCTRSTOP и инкрементируется, начиная со значения 0000\_0000h до значения FFFF\_FFFFh, после чего сбрасывается.

Чтобы синхронизировать работу таймера с другими блоками счетчик таймера может быть в любой момент загружен новым значением, которое предварительно записывается в теневой регистр CTRPHS. Загрузка может быть активирована как программно – запись единицы в бит SWSYNC регистра ECCTL1, так и аппаратно – приход синхроимпульса по входу SYNCI. Разрешение синхронизации осуществляется установкой бита SYNCIEN.

Блок захвата также может генерировать сигнал синхронизации SYNCO. Источник выбирается полем SYNCOSSEL либо используется SYNCI или SWSYNC, либо событие CTR = PRD генерации ШИМ.

Входы и выходы синхронизации блоков захвата подключены по цепочке, см. рисунок 19.3.

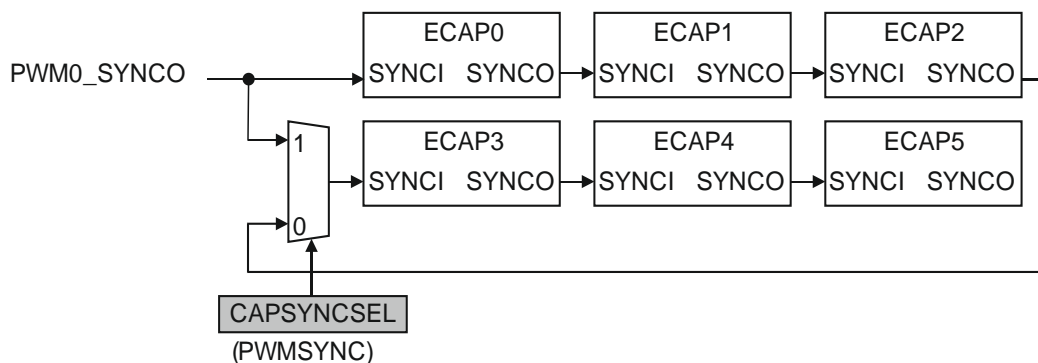


Рисунок 19.3 – Подключение сигналов синхронизации блоков захвата

Таймер может быть сброшен (с предварительным захватом его значения) при формировании событий EV0 – EV3. Указать событие можно установкой соответствующего бита CTRRSTn (где n от 0 до 3) в регистре ECCTL0. Так, например, если установлен бит CTRRST2, то при формировании события EV2 произойдет захват значения таймера в регистр CAP2 и сброс таймера.

### Регистры CAP0 – CAP3

32-разрядные регистры CAP0 – CAP3, сохраняющие (захватывающие) значение счетчика таймера в момент появления положительного фронта сигнала события EVn (где n от 0 до 3), доступны только для чтения.

## 19.2 Режим работы «генератор ШИМ»

Режим работы «генератор ШИМ» выбирается установкой бита CAPAPWM в регистре ECCTL1. Вывод CAP\_PWMx (где x от 0 до 5) функционирует как выход. Блок захвата в этом случае используется как одноканальный 32-разрядный генератор сигнала ШИМ.

### Таймер и регистры захвата

Таймер функционирует как 32-разрядный инкрементный счетчик, работающий на системной частоте. После включения счетчик таймера считает от значения 0000\_0000h до значения, которое задается регистром CAP0. Как только значения счетчика и регистра совпадают, счетчик сбрасывается.

Регистр CAP0 является регистром периода таймера, а регистр CAP1 – регистром сравнения. Регистры CAP2 и CAP3 являются регистрами отложенной загрузки для регистров CAP0 и CAP1, соответственно. Все регистры доступны как для записи, так и для чтения.

Запись в регистр CAP0 является мгновенной загрузкой, которая аппаратно дублируется записью в регистр CAP2. Аналогично для пары CAP1, CAP3.

Запись в регистры CAP2 и CAP3 является отложенной загрузкой. Как только значение счетчика таймера достигает значения периода CAP0, возникает событие  $CTR = PRD$ , по которому происходит сброс таймера и перегрузка значений из CAP2 и CAP3 в регистры CAP0 и CAP1 (на рисунке 19.1 отмечено стрелками черного цвета).

Регистры CAP0 и CAP1 должны быть обязательно инициализированы до начала запуска таймера. При дальнейшей работе можно изменять значения только регистра отложенной загрузки.

### Генерация ШИМ

После инициализации регистров CAP0 и CAP1 запускается счетчик таймера. Текущее значение счетчика CTR посредством двух компараторов сравнивается одновременно со значением PRD регистра периода CAP0 и значением CMP регистра сравнения CAP1 (на рисунке 19.1 отмечено черным цветом).

Как только возникает событие  $CTR = CMP$ , сигнал на выходе CAP\_PWMx переводится в ноль. Далее сигнал удерживается в нуле до тех пор, пока счетчик таймера не достигнет значения периода. При возникновении события  $CTR = PRD$  сигнал переводится в единицу. Одновременно с этим происходит сброс таймера и перезагрузка регистров CAP0 и CAP1. Управлять полярностью сигнала можно битом APWMPOL. На рисунке 19.4 представлен пример формирования сигнала ШИМ с активным высоким уровнем сигнала (по умолчанию, APWMPOL = 0).

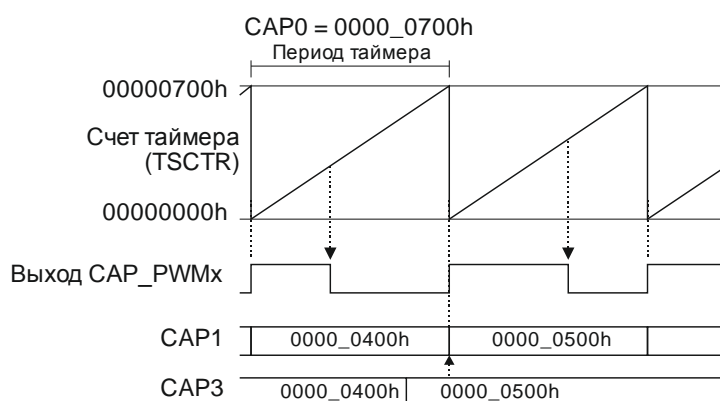


Рисунок 19.4 – Пример формирования сигнала ШИМ

На рисунке 19.4 период таймера задан как CAP0 = 0000\_0700h. Начальное значение сравнения CAP1 = 0000\_0400h. Пока таймер считает, в регистр CAP3 загружается новое значение 0000\_0500h для отложенной загрузки. По достижении значения сравнения сигнал на выходе CAP\_PWMx переводится в низкий уровень. По окончании периода происходит сброс таймера и загрузка значения 0000\_0500h (из регистра CAP3) в регистр CAP1 и перевод сигнала на выходе CAP\_PWMx в высокий уровень.

Таким образом, можно достаточно гибко управлять как длительностью импульсов, изменяя период работы таймера, так и скважностью при постоянном периоде.

### 19.3 Прерывания

Источники прерываний блока захвата:

- события EV0 – EV3;
- переполнение счетчика таймера CTROVF;
- события CTR = PRD;
- события CTR = CMP.

Каждое из семи прерываний имеет бит маски в регистре ECEINT, флаг прерывания в регистре ECFLG, бит сброса флага в регистре ECCLR и бит программного прерывания в регистре ECFRC. На рисунке 19.5 показан пример для прерывания по событию EV1.

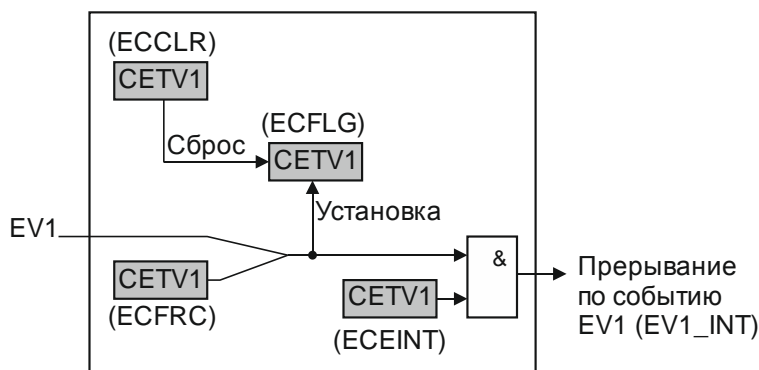


Рисунок 19.5 – Пример управления прерыванием EV1

Все прерывания по событиям поступают на блок управления прерываниями и обрабатываются, как сказано выше. При возникновении любого из этих прерываний в регистре PEINT устанавливается флаг PEINT, и генерируется прерывание блока захвата, см. рисунок 19.6.

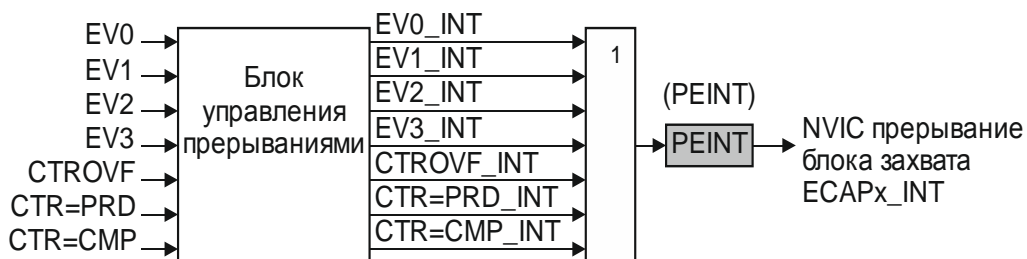


Рисунок 19.6 – Общая схема управления прерываниями

Примечание – Программа обслуживания прерывания должна сбрасывать флаг PEINT во избежание повторного обслуживания прерывания от блока захвата. Для сброса флага следует записать единицу в нулевой бит регистра PEINT.



## 20 Блоки ШИМ

Архитектура блоков ШИМ разработана по принципу минимальной нагрузки на процессор, что достигается автоматизацией формирования выходных импульсов с настраиваемыми пользователем параметрами. Так, после минимальных настроек, эти блоки способны работать самостоятельно, формируя выходные сигналы на выводах PWMx\_A и PWMx\_B (где x от 0 до 9) микроконтроллера.

Микроконтроллер содержит 10 блоков ШИМ, объединенных схемой синхронизации. Блоки ШИМ0 - ШИМ5 являются блоками ШИМ высокого разрешения.

Каждый блок ШИМ поддерживает следующую функциональность:

- 16-разрядный таймер;
- выходы PWMx\_A и PWMx\_B могут работать в режиме фронтальной и центрированной модуляции как полностью независимо, так и комплементарно с разделением генератором «мертвого» времени;
- выходы PWMx\_A и PWMx\_B могут управляться в зависимости от событий цифровых компараторов блока АЦП, компараторов SDFM, обеспечивая автоматический релейный режим поддержания заданной величины;
- программное управление выходами ШИМ;
- программное задание фазы счетчиков таймера для координации работы нескольких блоков ШИМ;
- аппаратный контроль фазы при координации работы нескольких блоков ШИМ;
- предотвращение наложения фронтов за счет генератора «мертвого» времени с независимой схемой задержки переднего и заднего фронтов выходного сигнала;
- сигнал аварии может переводить выходы PWMx\_A и PWMx\_B в высокое, низкое или Z-состояние;
- однократная и циклическая обработка сигналов аварии;
- все события могут инициировать прерывания;
- при обработке прерываний программируемый предделитель событий позволяет снизить нагрузку на процессор;
- сигнал ШИМ может модулироваться высокочастотным сигналом при использовании драйверов ключей с импульсным трансформатором.

Для начала работы с блоками ШИМ необходимо снять с них сброс и разрешить тактирование – необходимо установить биты PWMxEN регистров PCLKCFG и PRSTCFG.

Описание сигналов и выводов блока ШИМ:

- PWMx\_A и PWMx\_B (где x от 0 до 9) – выходы ШИМ;
- PWM\_TZ0- PWM\_TZ5 – входы, с которых принимаются сигналы аварии (общие для всех блоков ШИМ, и каждый блок может использовать или не использовать эти сигналы);
- PWM\_SYNCI – вход микроконтроллера, служащий для приема синхросигнала;
- SYNCI – вход блока ШИМ, служащий для приема синхросигнала;
- SYNCO – выход блока ШИМ, использующийся для синхронизации блоков;
- PWM\_INT – прерывание обработчика событий;
- PWM\_TZINT – прерывание по сигналу аварии;
- PWM\_HDINT – прерывание от порогового выключателя.

Функциональная схема блока ШИМ показана на рисунке 20.1, где заштрихованные области соответствуют функциям высокого разрешения, доступным для ШИМ0-ШИМ5.

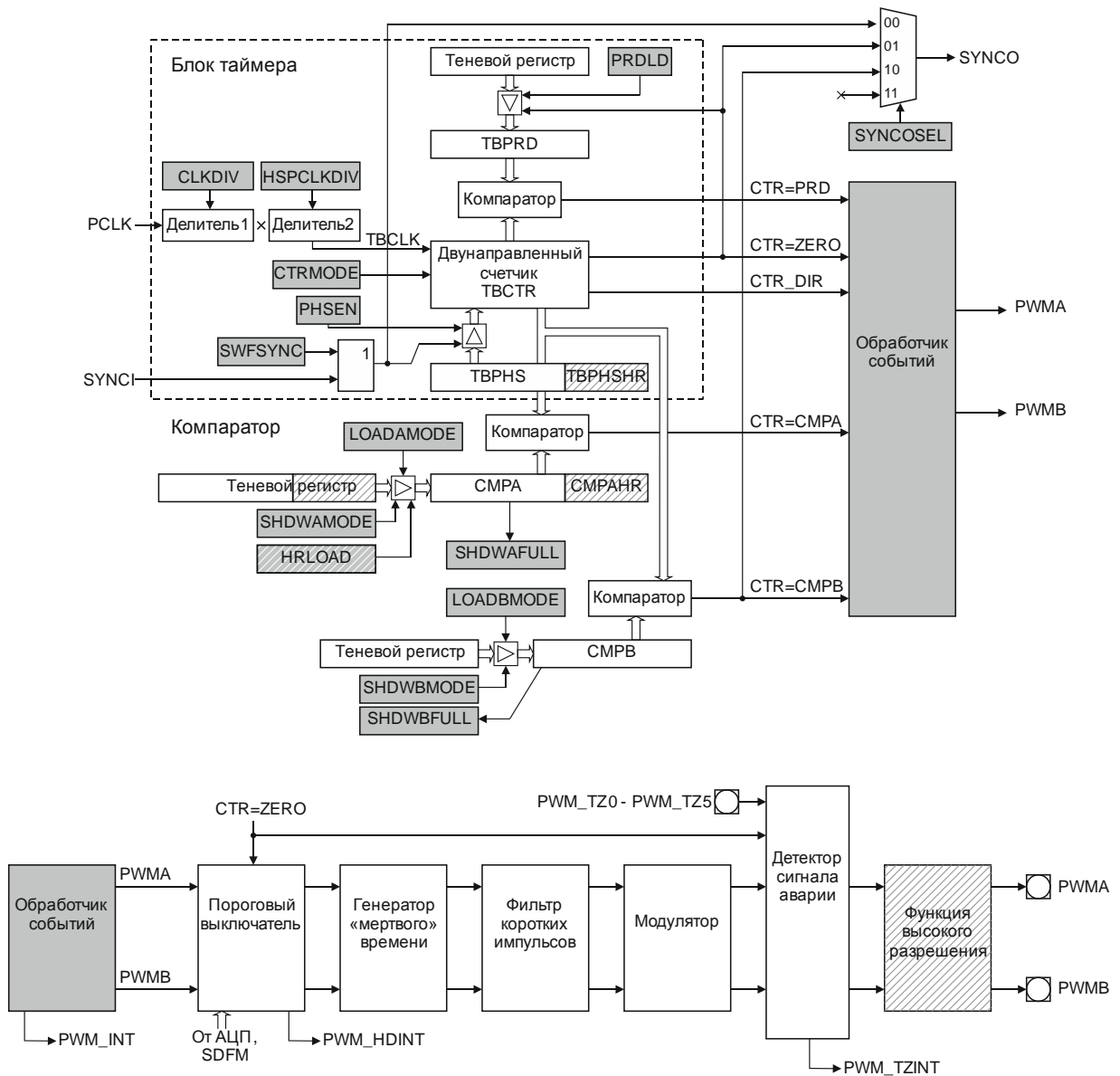


Рисунок 20.1 – Функциональная схема блока ШИМ

## 20.1 Таймер

Таймер представляет собой двунаправленный счетчик TBCTR, тактируемый сигналом TBCLK, который формируется на основе синхросигнала PCLK. Частота сигнала TBCLK задается произведением коэффициентов двух делителей. Коэффициенты задаются полями CLKDIV и HPCLKDIV регистра TBCTL. Для работы других блоков ШИМ счетчик позволяет формировать события такие, как совпадение по периоду  $CTR = PRD$  ( $TBCTR = TBPRD$ ), совпадение с нулем  $CTR = Zero$  ( $TBCTR = 0000h$ ), совпадение с регистрами  $CTR = CMPA$  и  $CTR = CMPB$  ( $TBCTR = CMPA$  и  $TBCTR = CMPB$ , соответственно). Событие  $TBCTR = FFFFh$  влияет только на флаг CTRMAX регистра TBSTS.

Всеми настройками работы счетчика таймера управляет регистр TBCTL.

Состояние счетчика отражают флаги регистра TBSTS.

На выходе первого блока ШИМ формируется сигнал SYNCO, который является синхросигналом для остальных блоков ШИМ, см. рисунок 20.3а. Сигнал SYNCO имеет три источника – программно сгенерированный синхроимпульс, посредством записи

единицы в бит SWFSYNC, события  $CTR = Zero$  и  $CTR = CMPB$ . Выбор источника осуществляется посредством поля SYNCOSSEL.

В блоке таймера находятся регистры начальной фазы счета  $TBPHS$  и периода (максимального значения счетчика)  $TBPRD$ . Регистр периода имеет теневой регистр для синхронной загрузки значения, до которого счетчик осуществляет счет. Управление загрузкой осуществляется битом  $PRDLD$ .

Счетчик может работать в трех режимах счета, см. рисунок 20.2:

- вверх (от  $0000h$  до значения  $TBPRD$ , затем сброс в  $0000h$  и т. д.);
  - вниз (от значения  $TBPRD$  до  $0000h$ , затем загрузка значения  $TBPRD$  и т. д.);
  - вверх-вниз (от  $0000h$  до значения  $TBPRD$ , затем от значения  $TBPRD$  до  $0000h$  и т. д.).
- Параметры счета задаются полем  $CTRMODE$ .

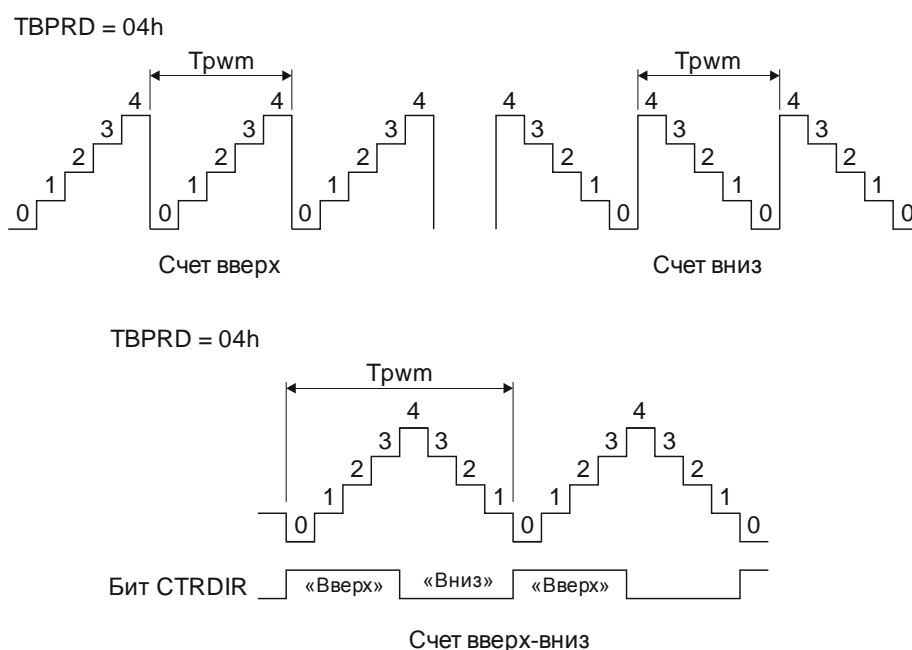


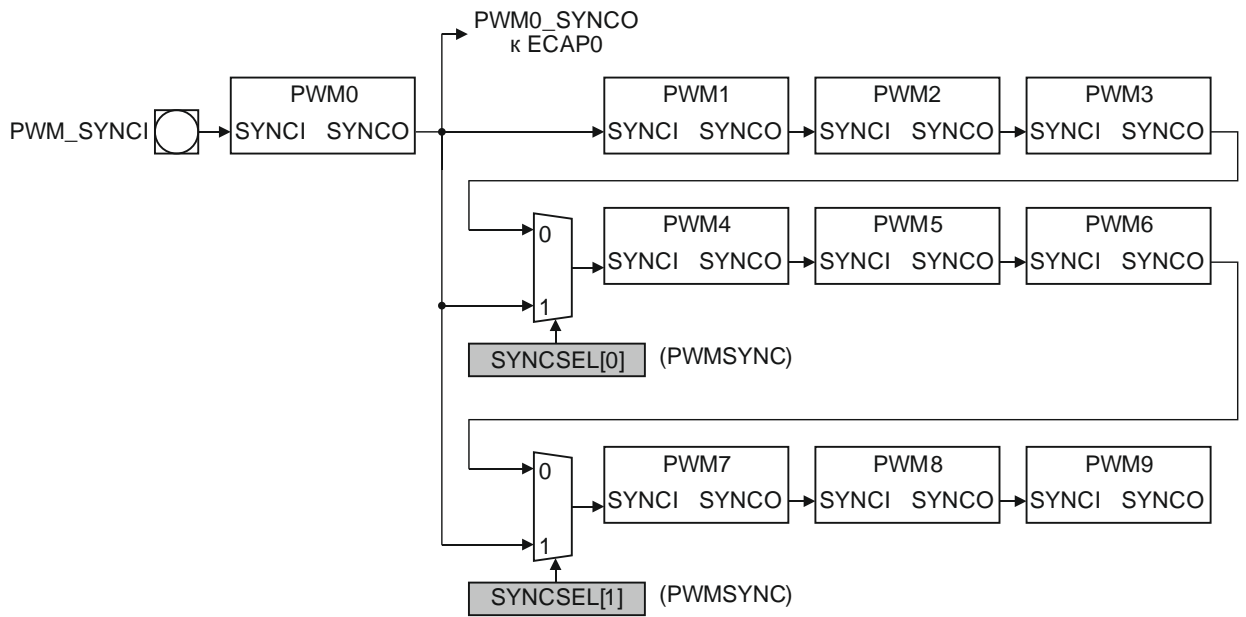
Рисунок 20.2 – Режимы работы счетчика при значении периода  $0004h$  ( $Trwm$ ); для режима счета «вверх-вниз» дополнительно указано поведение флага  $CTRDIR$

### Теневая загрузка

Регистры  $TBPRD$ ,  $CMRA$ ,  $CMRANR$ ,  $CMRB$  имеют соответствующие теневые регистры и управляющие биты, регулирующие режим и событие загрузки. Помимо индивидуальной настройки теневого режима есть бит  $SHDWGLOB$  регистра  $TBCTL$ . По умолчанию он установлен, и теневая загрузка работает согласно настройкам. Но возможны ситуации, когда необходимо обеспечить одновременную загрузку всех теневых регистров. Тогда перед загрузкой необходимо сбросить  $SHDWGLOB$ , в результате значения будут попадать в теневые регистры, но перезапись в активные будет блокирована. Когда все необходимые регистры будут записаны, следует установить этот бит, и тогда все активные регистры будут перезаписаны в соответствии с заданными настройками по соответствующим событиям. Данный бит не оказывает влияния, если для регистров выбрана прямая загрузка, без участия теневых.

## Синхронизация таймеров блоков ШИМ

Реализована схема синхронизации блоков ШИМ, см. рисунок 20.3а.



а)

Событие синхронизации (при счете вверх-вниз)

Счетчик TBCTRL блока 0 ШИМ

2	1	0	1	2	3	4
---	---	---	---	---	---	---

Счетчик TBCTRL блока 1 ШИМ

x	x	x	1	2	3	4
---	---	---	---	---	---	---

Такт записи нового значения

Событие синхронизации (при счете вверх)

Счетчик TBCTRL блока 0 ШИМ

98	99	0	1	2	3	4
----	----	---	---	---	---	---

Счетчик TBCTRL блока 1 ШИМ

x	x	x	1	2	3	4
---	---	---	---	---	---	---

Такт записи нового значения

Событие синхронизации (при счете вниз)

Счетчик TBCTRL блока 0 ШИМ

2	1	0	99	98	97	96
---	---	---	----	----	----	----

Счетчик TBCTRL блока 1 ШИМ

x	x	x	99	98	97	96
---	---	---	----	----	----	----

Такт записи нового значения

Счетчик TBCTRL блока 0 ШИМ

x	x	0	x	x	x	x
---	---	---	---	---	---	---

Счетчик TBCTRL блока 1 ШИМ

x	x	x	0	x	x	x
---	---	---	---	---	---	---

б)

в)

Рисунок 20.3 – Схема синхронизации модулей ШИМ

Система синхронизации таймеров включает в себя таймеры всех блоков ШИМ. Каждый блок ШИМ имеет вход синхронизации SYNCI и выход синхронизации SYNCO.

Если бит PHSEN установлен, то в счетчик таймера будет автоматически загружаться значение регистра TBRHS, при выполнении каждого из условий:

- изменение входного сигнала SYNCI (в этом случае загрузка значения TVPHS в регистр TVCTR происходит на следующий такт TVCLK после поступления импульса на вход SYNCI с задержкой в два системных такта, если TVCLK = PCLK, или один такт, если TVCLK ≠ PCLK);

- запись единицы в бит SWFSYNC (программная синхронизация), которая генерирует импульс синхронизации, аналогичный импульсу с входа SYNCI.

В режиме счета вверх-вниз необходимо запрограммировать бит PHSDIR, чтобы задать направление счета таймера после синхронизации.

Если бит PHSEN сброшен, блок ШИМ не будет реагировать на входной сигнал синхронизации, а только передавать напрямую этот сигнал на выход SYNSO, чтобы тактировать другие блоки ШИМ. Следующая особенность схемы: генерация и распространение сигнала синхронизации от блока ШИМ, – занимает один такт TVCLK.

К примеру, если по событию синхронизации блока 0 в счетчик блока 1 должен быть записан ноль, то этот ноль запишется только на следующий такт после события, см. рисунок 20.3б. Таким образом, при синхронизации от другого блока ШИМ нужно всегда учитывать этот такт и записывать значение фазы, следующее по порядку, в соответствии с режимом счета, см. рисунок 20.3в.

## 20.2 Компаратор

Компаратор – это блок, сравнивающий значение счетчика таймера с заданными значениями порогов срабатывания. Значения хранятся в регистрах CMPA и CMPB. Значения, записываемые по адресам регистров CMPA и CMPB, предварительно размещаются в теневых регистрах. Это нужно для синхронной загрузки новых значений. Управление загрузкой регистров CMPA и CMPB осуществляется битами SHDWAMODE и SHDWBMODE, а также полями LOADAMODE и LOADBMODE регистра CMPCTL.

Блок компаратора формирует на выходах два события CTR = CMPA и CTR = CMPB, возникающие в случае совпадения значения счетчика с регистром CMPA и/или регистром CMPB, соответственно.

Для каждого компаратора событие может возникать:

- один раз за период, если счетчик считает вверх или вниз;
- один раз за период, если счетчик считает вверх-вниз, но при этом значение в регистре CMPA/CMPB равно 0000h или значению TVPRD;
- два раза за период, если счетчик считает вверх-вниз и при этом значение в регистре CMPA/CMPB лежит в диапазоне 0001h – (TVPRD – 1).

На рисунках 20.4 – 20.7 приведены примеры формирования сигналов событий.

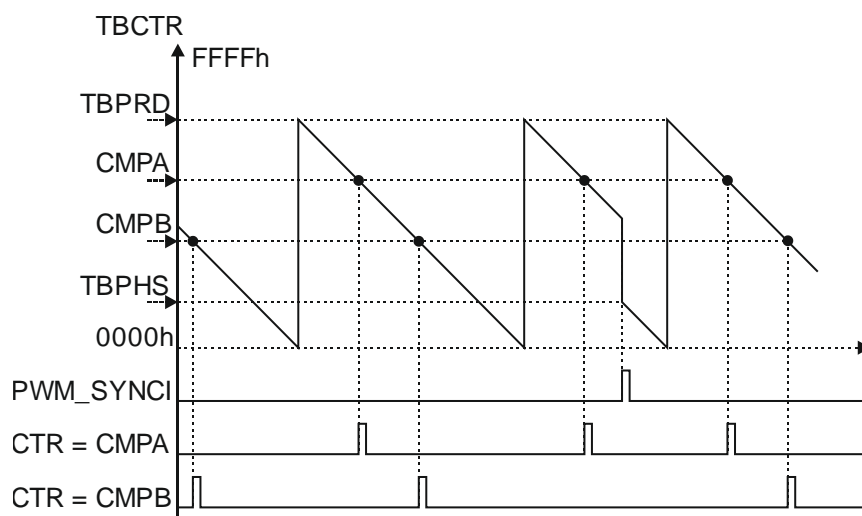


Рисунок 20.4 – Диаграмма работы при счете вниз

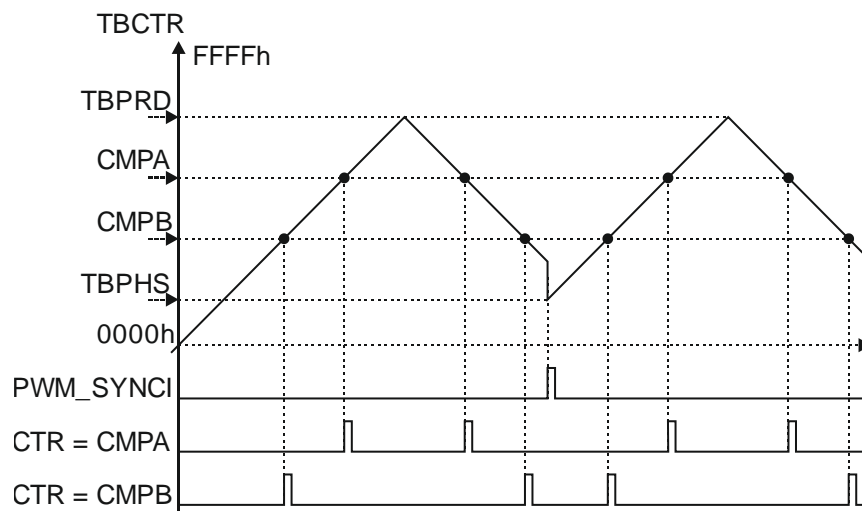


Рисунок 20.5 – Диаграмма работы при счете вверх-вниз.  
Синхронизация при счете вверх

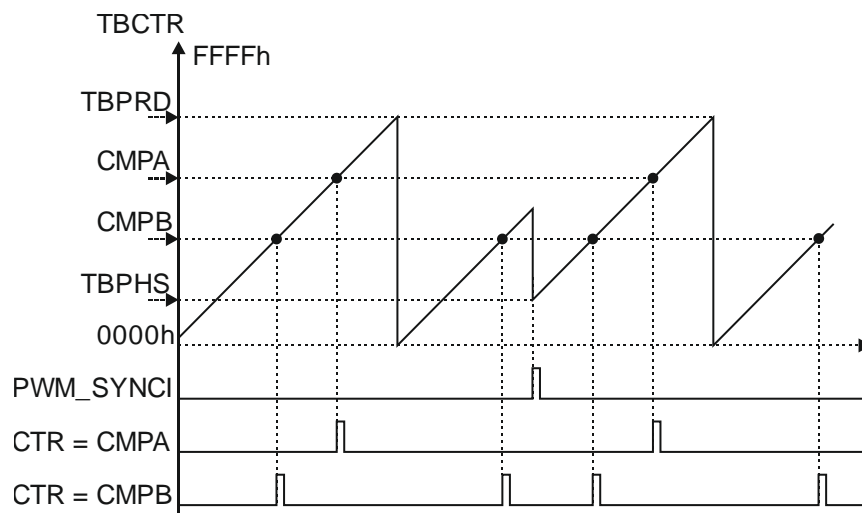


Рисунок 20.6 – Диаграмма работы при счете вверх

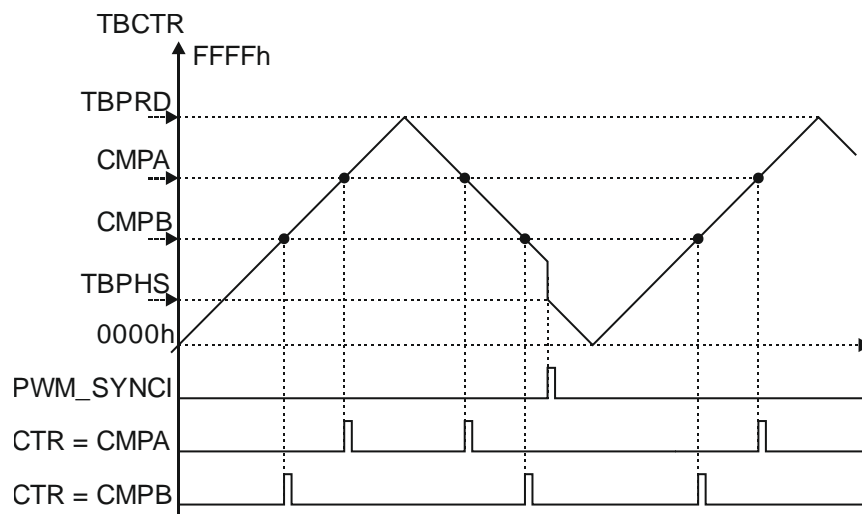


Рисунок 20.7 – Диаграмма работы при счете вверх-вниз.  
Синхронизация при счете вниз

### 20.3 Обработчик событий

Обработчик событий – блок, управляющий поведением сигналов на линиях PWMA и PWMB, см. рисунок 20.1, в зависимости от возникающих событий на входе блока и направления счета счетчика таймера. На поведение выходных сигналов влияют импульсы входных сигналов при возникновении событий: CTR = PRD, CTR = Zero, CTR = CMPA, CTR = CMPB.

Основные действия с сигналами PWMA и PWMB:

- переключение в единицу или ноль;
- инверсия (переключение в противоположное состояние);
- сохранение без изменений.

Поведение сигналов задается независимо друг от друга. Кроме этого, обработчик событий позволяет программно задавать состояние сигналов PWMA и PWMB и величину «мертвого» времени ШИМ. Управление работой блока производится посредством регистров AQCTLA, AQCTLB, AQSFRC, AQCSFRC.

Существует вероятность того, что несколько событий могут произойти одновременно. Для таких ситуаций обработчик событий использует систему приоритетов событий.

Таблица 20.1 – Распределение приоритетов событий при счете вверх

Событие	Приоритет
Программное	1 (самый высокий)
CTR = TBPRD	2
CTR = CMPB (счет вверх) при счете вверх	3
CTR = CMPA (счет вверх) при счете вверх	4 (самый низкий)

Таблица 20.2 – Распределение приоритетов событий при счете вниз

Событие	Приоритет
Программное	1 (самый высокий)
CTR = Zero	2
CTR = CMPB (счет вниз) при счете вниз	3
CTR = CMPA (счет вниз) при счете вниз	4 (самый низкий)

Таблица 20.3 – Распределение приоритетов событий при счете вверх-вниз

Событие	Приоритет
Программное	1 (самый высокий)
CTR = CMPB (счет вверх) при счете вверх или CTR = CMPB (счет вниз) при счете вниз	2
CTR = CMPA (счет вверх) при счете вверх или CTR = CMPA (счет вниз) при счете вниз	3
CTR = Zero или CTR = PRD	4
CTR = CMPB (счет вверх) при счете вниз или CTR = CMPB (счет вниз) при счете вверх	5
CTR = CMPA (счет вверх) при счете вниз или CTR = CMPA (счет вниз) при счете вверх	6 (самый низкий)

В режиме счета вверх:

- если компаратор запрограммирован так, что  $СМРА/СМРВ \leq ТВРД$  (счет вверх), то событие произойдет при  $СТР = СМРА/СМРВ$ ;
- если компаратор запрограммирован так, что  $СМРА/СМРВ > ТВРД$  (счет вверх), то событие не произойдет;
- если компаратор запрограммирован на срабатывание при счете вниз, то событие не произойдет.

В режиме счета вниз:

- если компаратор запрограммирован так, что  $СМРА/СМРВ \leq ТВРД$  (счет вниз), то событие произойдет при  $СТР = СМРА/СМРВ$ ;
- если компаратор запрограммирован так, что  $СМРА/СМРВ \geq ТВРД$  (счет вниз), то событие произойдет при  $СТР = ТВРД$ ;
- если компаратор запрограммирован на срабатывание при счете вверх, то событие не произойдет.

В режиме счета вверх-вниз, см. рисунок 20.8:

- если счетчик считает вверх, а компаратор запрограммирован так, что  $СМРА/СМРВ < ТВРД$  (счет вверх), то событие произойдет при  $СТР = СМРА/СМРВ$ ;
- если  $СМРА/СМРВ \geq ТВРД$  (счет вверх), то событие произойдет при  $СТР = ТВРД$ ;
- если счетчик считает вниз, а компаратор запрограммирован так, что  $СМРА/СМРВ < ТВРД$  (счет вниз), то событие произойдет при  $СТР = СМРА/СМРВ$ ;
- если  $СМРА/СМРВ \geq ТВРД$  (счет вверх), то событие произойдет при  $СТР = ТВРД$ .

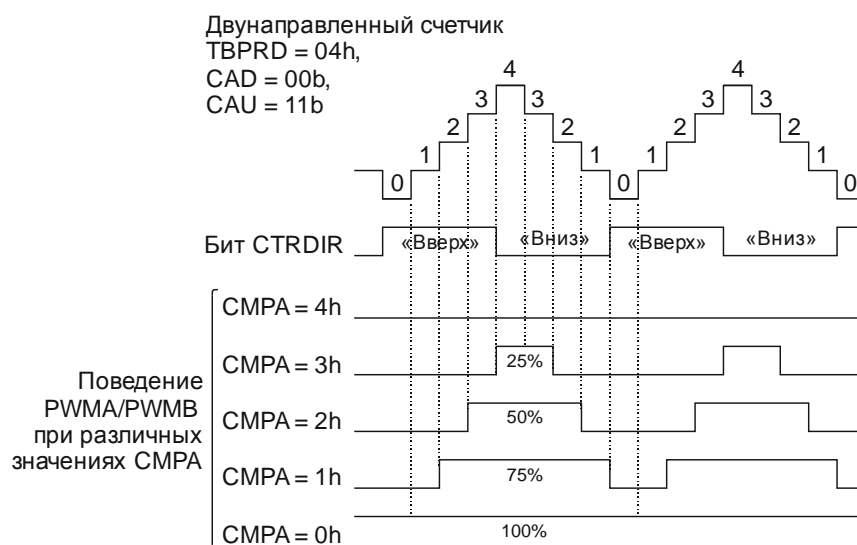


Рисунок 20.8 – Работа таймера при счете вверх-вниз с симметричным выходом (центрированная модуляция)

На рисунках 20.9 – 20.11 показано поведение линий PWMA и PWMB при различных видах модуляции, принятые обозначения и пояснения приведены в таблице 20.4.



Таблица 20.4 – Пояснения к обозначениям на рисунках 20.9 – 20.11

Обозначение			Пояснение	
P ×	CA ×	CB ×	События CTR = PRD, CTR = CMPA, CTR = CTRB, соответственно. Символ «×» указывает на то, что при возникновении этого события сигнал на линии PWMA/PWMB остается без изменений. Пунктирными линиями отмечены моменты возникновения события. Так, например, см. рисунок 20.9, при возникновении события CTR = CTRB, сигнал на линии PWMA остается без изменения, а сигнал на линии PWMB переключается в ноль	
Z ↑	Z ↓	Событие CTR = Zero		Символы «↑»/«↓» указывают на то, что при возникновении этого события сигнал на линии PWMA/PWMB переключается в единицу/ноль
CA ↑	CA ↓	Событие CTR = CMPA		
CB ↑	CB ↓	Событие CTR = CMPB		

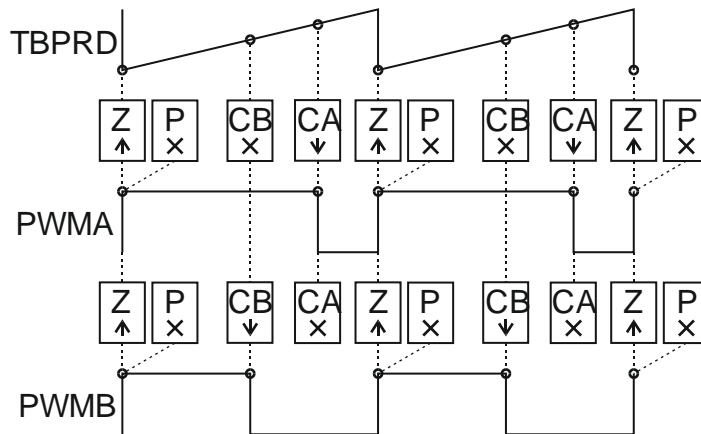


Рисунок 20.9 – Независимый режим работы выходов (фронтная модуляция)

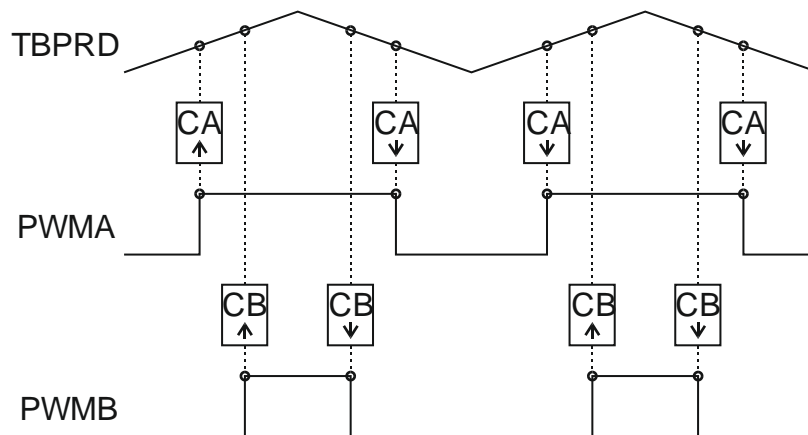


Рисунок 20.10 – Симметричный режим работы при счете вверх-вниз (центрированная модуляция)

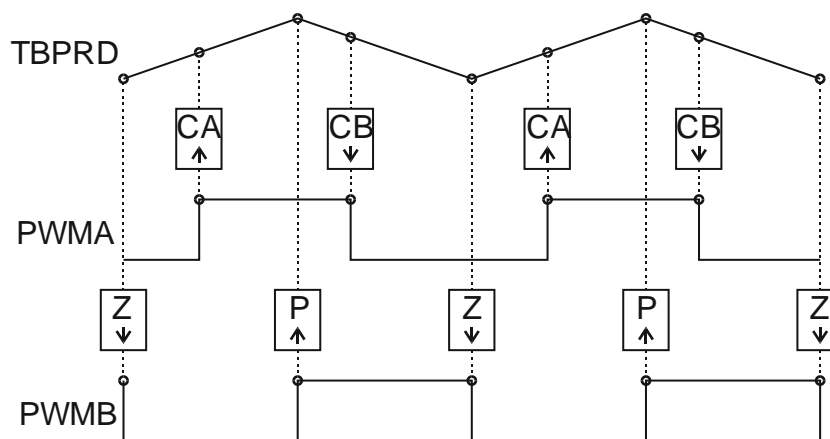


Рисунок 20.11 – Ассиметричный режим работы при счете вверх-вниз

## 20.4 Пороговый выключатель

Пороговый выключатель контролирует выходные сигналы PWMA и PWMB обработчика событий и позволяет удерживать их в определенном заданном пользователем состоянии в случае прихода сигнала триггера от цифровых компараторов блока АЦП или компараторов и фильтров SDFM. Этот блок удобен для организации релейного поддержания заданного уровня какой-либо физической величины, например для организации контура тока. В этом случае цифровой компаратор, к каналу которого подключен сигнал датчика тока контура, формирует сигнал о превышении током задания, а соответствующий пороговый выключатель реагирует на это превышение и включает/отключает соответствующий силовой транзистор посредством влияния на выход ШИМ.

Функциональные возможности:

- входные события от компараторов блока АЦП и компараторов и фильтров SDFM могут использоваться всеми блоками ШИМ;
- при регистрации события от компаратора выходные сигналы обработчика событий могут быть переведены в состояние логической единицы, нуля или оставлены без изменений;
- поддерживаются однократное и циклическое срабатывания для удержания выхода;
- входное событие от компаратора может анализироваться в однократном и циклическом режимах;
- событие срабатывания компаратора может быть сгенерировано программно;
- пороговый выключатель может быть отключен, если он не требуется;
- может генерироваться прерывание по событиям порогового выключателя.

Управление пороговым выключателем осуществляется посредством регистров HDSEL0, HDSEL1, HDCTL и HDFRC.

### Компараторы блока АЦП

Когда выходные сигналы компараторов блока АЦП переходят в состояние высокого уровня, формируется событие. Каждый пороговый выключатель блока ШИМ может использовать, а может не использовать эти события в своей работе; выбор, по сигналу какого компаратора блока АЦП формировать событие удержания, задается с помощью регистра HDSEL0. Длительность импульса на входном сигнале от компаратора блока АЦП не должна быть меньше периода системного синхросигнала. Каждый входной сигнал компаратора блока АЦП должен быть настроен на однократное или циклическое формирование события, выбор режима задается битами CBC и OST регистра HDCTL, а источник события – полем ADCDC регистра HDSEL0.

При получении события от компаратора блока АЦП в режиме циклической обработки немедленно формируется реакция на основе содержимого регистра HDCTL, в результате чего меняется состояние сигналов на выходе порогового выключателя взамен полученных от обработчика событий PWMA и/или PWMB на заданное пользователем в регистре HDCTL. Дополнительно устанавливается флаг CBC в регистре HDFLG, и генерируется прерывание PWM\_HDINT. Удержание выходных сигналов PWMA и PWMB заканчивается по событию TVCTR = 0000h, при условии, что событие компаратора блока АЦП уже не активно. Таким образом, в режиме циклической обработки состояние удержания сбрасывается в каждом периоде ШИМ. При этом флаг CBC остается активным до его программного сброса. Если после сброса флага CBC вновь будет получено событие компаратора блока АЦП, то флаг установится вновь.

При получении события компаратора блока АЦП в режиме однократной обработки, также немедленно формируется реакция на основе содержимого регистра HDCTL, которая меняет состояние выходных сигналов PWMA и/или PWMB. В дополнение, устанавливается флаг OST, и генерируется прерывание PWM\_HDINT. Удержание выходных сигналов будет производиться до программного сброса записью единицы в бит OST регистра HDCLR.

Способ удержания выходных сигналов при получении события компаратора блока АЦП программируется индивидуально для выходных сигналов PWMA и PWMB в регистр HDCTL.

### **Компараторы и фильтры SDFM**

Функционирование происходит способом, аналогичным взаимодействию с компараторами АЦП, но с одним отличием - источники события определяются полями SDFML, SDFMH, SDFMZ регистра HDSEL1.

## **20.5 Генератор задержки ШИМ**

Блок имеет на входе сигналы PWMA и PWMB с выходов обработчика событий, а на выходах повторяет эти сигналы, но со вставкой задержки («мертвое» время) в момент переключения сигналов (если это необходимо).

Задержку можно учесть при программировании обработчика событий, но чтобы с высокой вероятностью избежать ошибок, желательно использовать генератор задержки ШИМ.

### **Основные функции генератора задержки ШИМ:**

- генерация пары выходных сигналов PWMA и PWMB с выдержкой интервалов (задержек) времени относительно входных сигналов PWMA, PWMB;
- программирование задержки для активного высокого и активного низкого уровня сигналов каналов PWMA и PWMB;
- добавление программируемой задержки передних фронтов сигналов;
- добавление программируемой задержки для задних фронтов сигналов;
- возможность передачи сигналов с входов на выходы без изменений.

Генератор задержки ШИМ программируется посредством регистров DBCTL, DBRED и DBFED. Структурная схема генератора «мертвого» времени ШИМ представлена на рисунке 20.12.

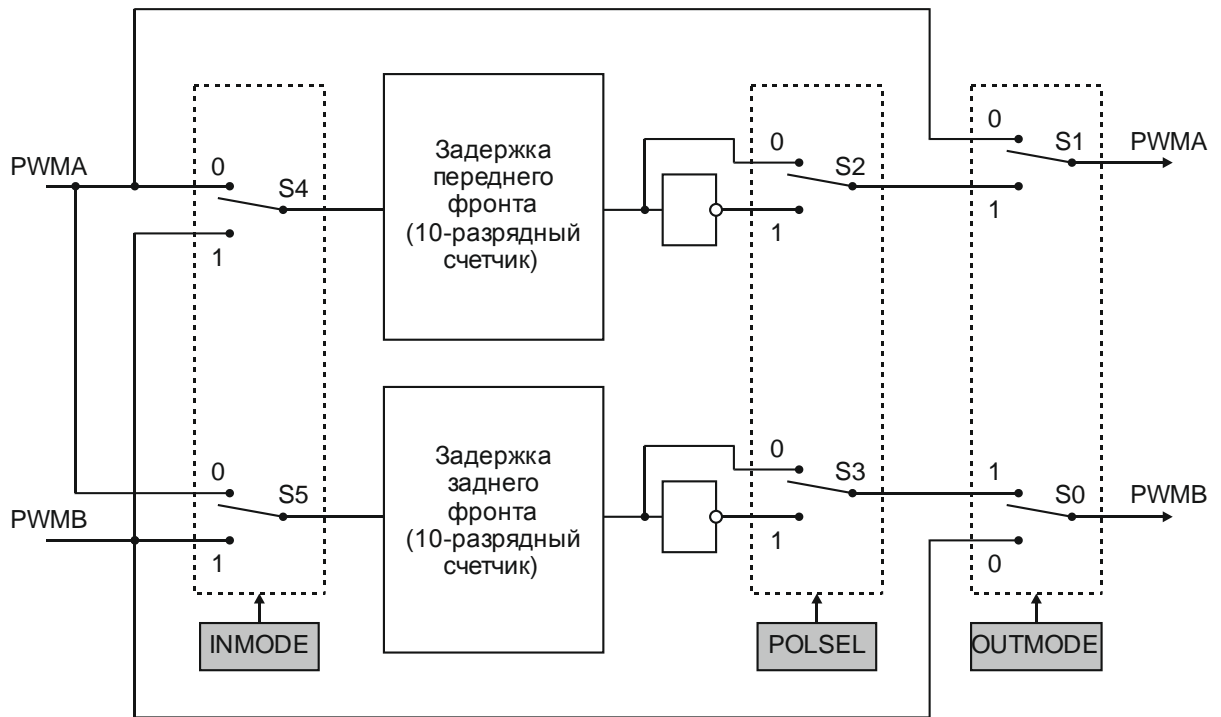


Рисунок 20.12 – Структурная схема генератора «мертвого» времени ШИМ

### Функционирование

Генератор задержки ШИМ может работать с четырьмя источниками (фронты сигналов PWMA и PWMB). Выбор источника задается полем MODE регистра DBCTL.

Поле POLSEL позволяет задать инверсию (переключение значения на противоположное) сигнала после внесения задержки, см. рисунок 20.13.

Величины задержек по переднему и заднему фронту программируются отдельно посредством регистров DBRED и DBFED, соответственно.

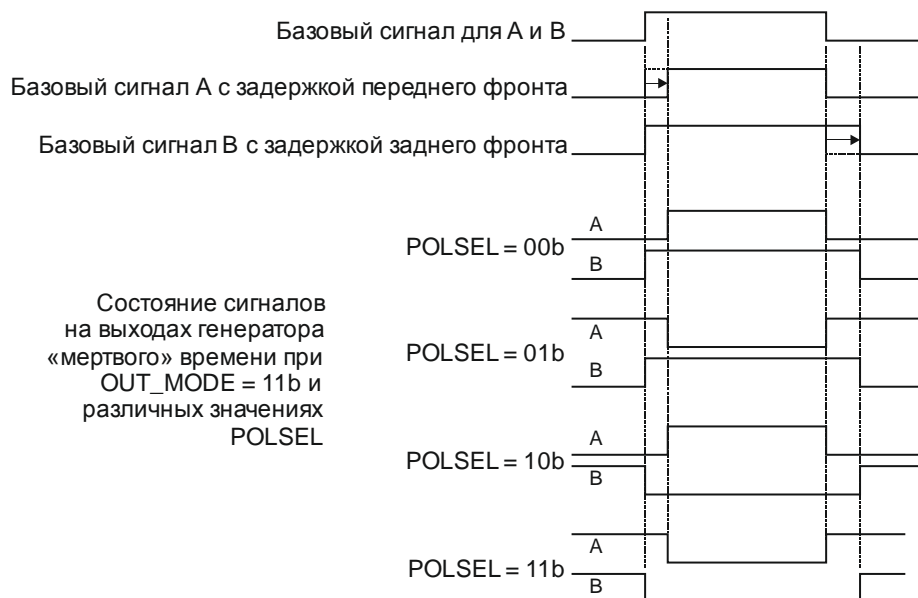


Рисунок 20.13 – Временные диаграммы работы генератора «мертвого» времени в типовой конфигурации

## 20.6 Фильтр коротких импульсов

Фильтр коротких импульсов предназначен для блокирования прохождения на выход импульсов с длительностью меньше заданной. Этот блок может применяться, если драйвер силового ключа инвертора не имеет такой функции, а для обеспечения правильного режима работы транзистора необходимо запретить открытие/закрытие транзистора на очень короткое время.

### Основные функции фильтра:

- программируемая ширина минимального пропускаемого импульса;
- фильтр может быть отключен, если он не требуется.

Ширина минимального импульса, допускаемого к прохождению на выход, задается в регистре FWDTH в тактах PCLK и может принимать значение от 00h (фильтр выключен) до FFh. Импульсы длительностью меньше заданной пропускаться не будут.

## 20.7 Модулятор

Блок позволяет модулировать выходной сигнал ШИМ с помощью высокочастотных импульсов программируемой скважности. Модулирование требуется для управления силовыми ключами через импульсный трансформатор.

### Основные функции модулятора:

- программируемая частота;
- программируемая ширина первого импульса;
- программируемый коэффициент заполнения второго и последующего импульсов;
- модулятор может быть отключен (бит CHPEN регистра PCCTL).

Модулятор программируется посредством регистра PCCTL. Структурная схема модулятора приведена на рисунке 20.14.

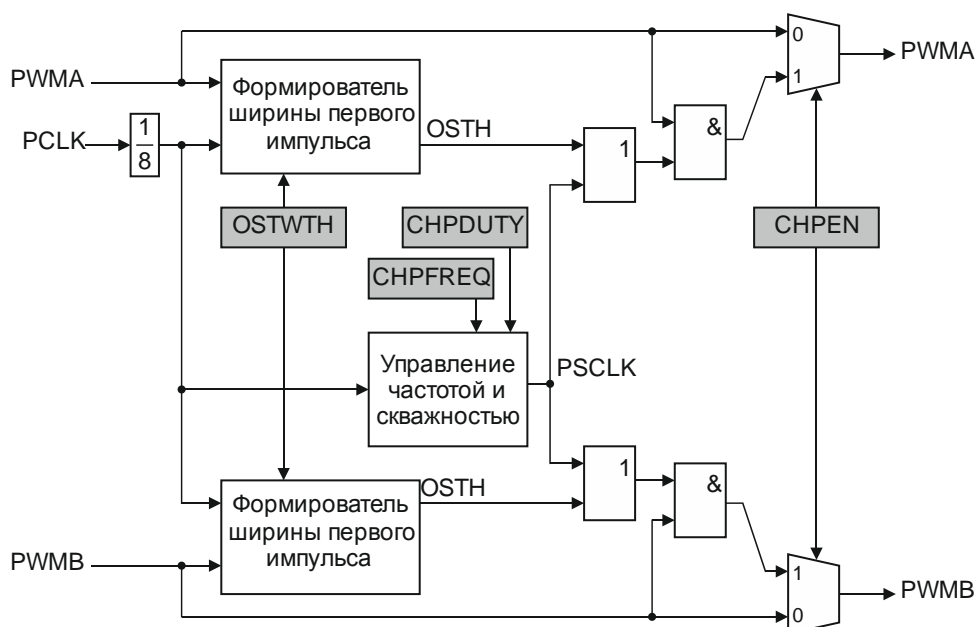


Рисунок 20.14 – Структурная схема модулятора

### Функционирование

Частота модуляции формируется на основе системной частоты при помощи делителя, программируемого полем CHPFREQ.

На рисунке 20.15 приведен пример временных диаграмм работы модулятора.

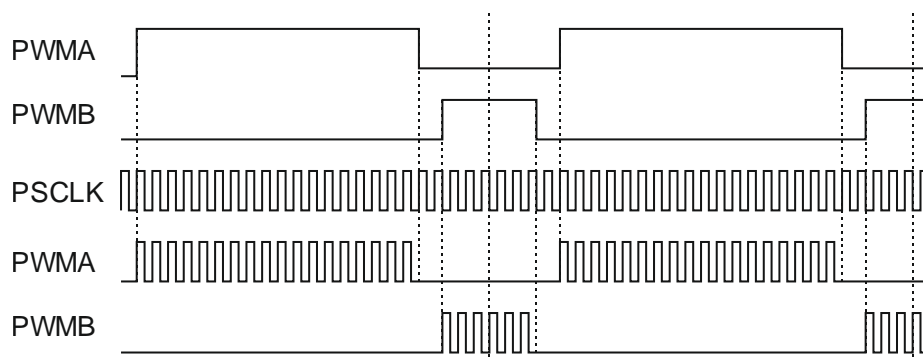


Рисунок 20.15 – Временные диаграммы работы модулятора

Ширина первого импульса может программироваться независимо с помощью поля OSTWTH, это требуется для открывания ключа. Значения поля OSTWTH лежат в диапазоне 0h – Fh.

Ширина L первого импульса определяется по формуле

$$L = T \times 8 \times (\text{OSTWTH} + 1), \quad (20.1)$$

где T – период синхросигнала PCLK.

На рисунке 20.16 приведен пример формирования расширенного первого импульса.

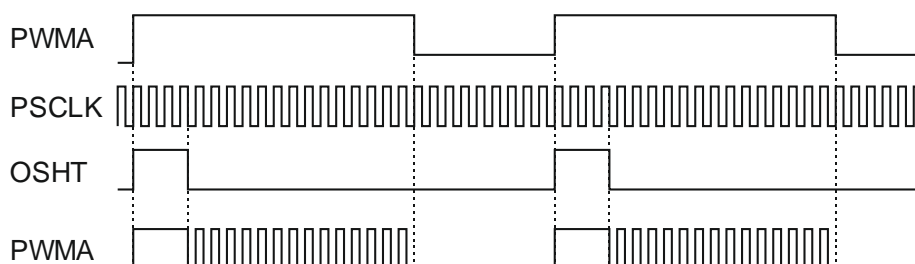


Рисунок 20.16 – Временные диаграммы работы модулятора с расширенным первым импульсом

Также существует возможность регулировать коэффициент заполнения импульсов посредством поля CHPDUTY. Значения поля CHPDUTY лежат в диапазоне 0h – 7h.

Коэффициент заполнения D (с шагом 12,5 %) последующих импульсов определяется по формуле

$$D = 12,5 \times (\text{CHPDUTY} + 1). \quad (20.2)$$

На рисунке 20.17 приведена иллюстрация регулировки коэффициента заполнения.

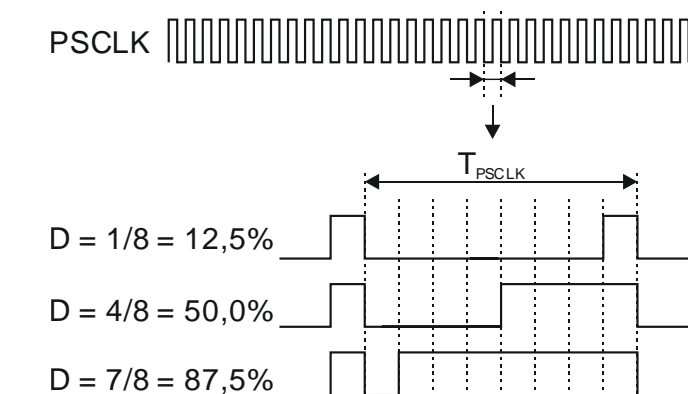


Рисунок 20.17 – Регулировка коэффициента заполнения D

## 20.8 Детектор сигнала аварии

Детектор сигнала аварии контролирует выходы PWMx\_A и PWMx\_B и может переводить их в определенное (запрограммированное) состояние в случае, если поступит сигнал аварии.

Основные функции детектора сигнала аварии:

- входные сигналы аварии с выводов микроконтроллера PWM\_TZ0- PWM\_TZ5 могут использоваться любым блоком ШИМ;

- в случае если поступит сигнал аварии, выходы ШИМ могут быть переведены в одно из состояний: логического нуля, логической единицы, высокоимпедансное или оставлены без изменения;

- поддерживается однократная блокировка выводов для ситуации короткого замыкания или перегрузки по току;

- поддерживается циклическая блокировка для режима ограничения тока;

- каждый входной источник сигнала аварии может быть обработан в однократном и циклическом режимах;

- поддерживается программная генерация сигнала аварии;

- детектор сигнала аварии может быть отключен, если он не требуется.

Детектор сигнала аварии программируется посредством регистров TZSEL, TZCTL, TZEINT, TZFLG, TZCLR и TZFRC.

### Функционирование

Структурная схема детектора сигналов аварии показана на рисунке 20.18.

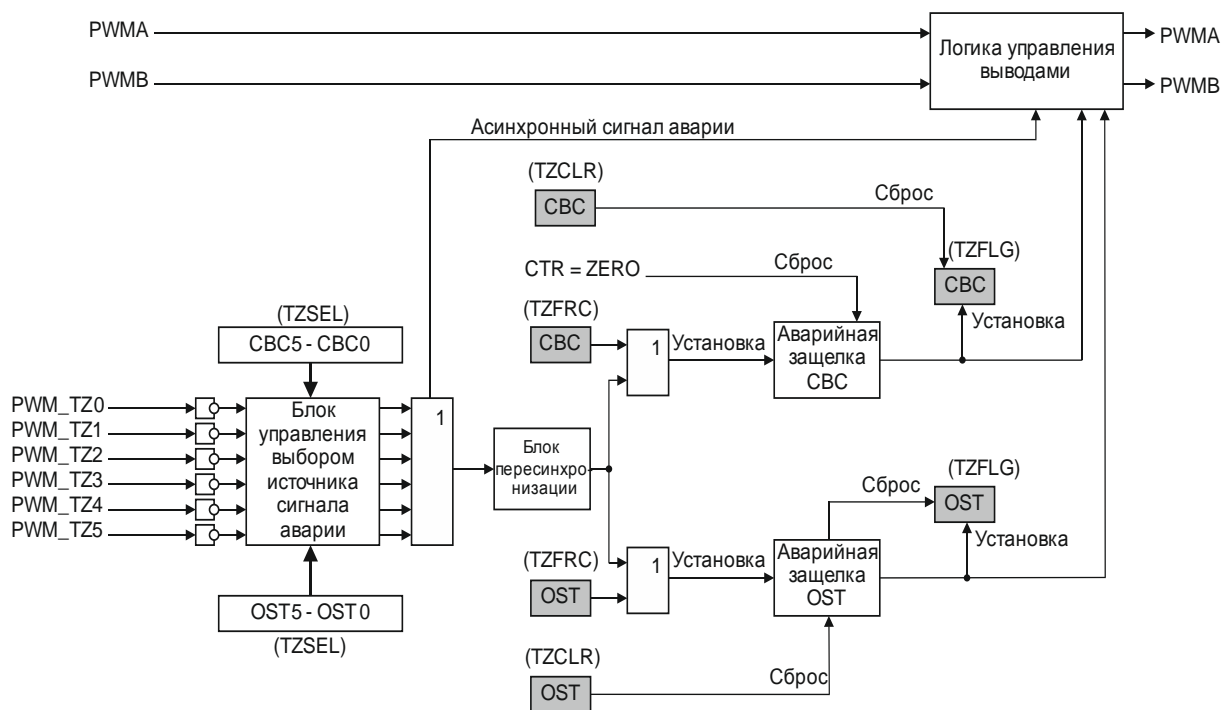


Рисунок 20.18 – Структурная схема детектора сигналов аварии

Переход входного сигнала аварии из состояния логической единицы в состояние логического нуля формирует событие аварии. Каждый блок ШИМ может использовать или не использовать это событие в своей работе (программируется посредством регистра TZSEL). События могут формироваться синхронно (с цифровым фильтром помех) или асинхронно (программируется через регистры GPIO микроконтроллера). При синхронной обработке длительность импульса на входном сигнале сбоя должна быть не меньше периода синхросигнала TVCLK. Если же обработка производится в асинхронном режиме, то событие формируется и обрабатывается даже в том случае, если по какой-либо причине

отключилось тактирование. Каждый входной сигнал аварии должен быть настроен на однократное или циклическое формирование события аварии (программируется посредством регистра TZSEL).

При получении события аварии в режиме циклической обработки немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг CBC в регистре TZFLG, а также генерируется прерывание PWM\_TZINT, если разрешено в регистре TZEINT и контроллером прерываний. Аварийное удержание выводов заканчивается по событию TVCTR = 0000h при условии, что событие аварии уже неактивно. Таким образом, в режиме циклической обработки событие аварии сбрасывается в каждом периоде ШИМ, хотя флаг аварии CBC остается установленным до принудительного программного сброса. Если после сброса регистра флага CBC вновь будет получено событие аварии, то флаг установится вновь. На рисунке 20.19 показана схема формирования прерывания.

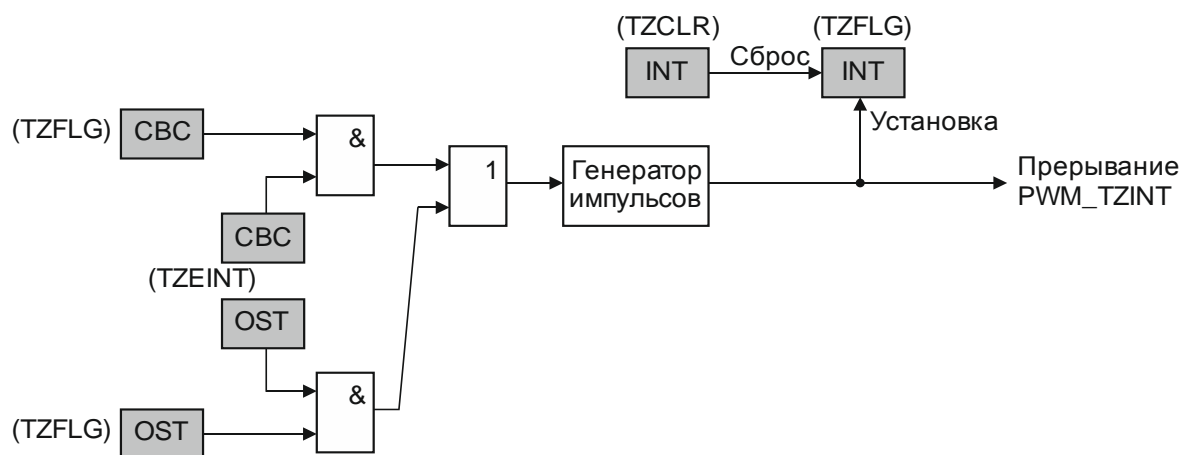


Рисунок 20.19 – Схема формирования прерывания

При получении события аварии в режиме однократной обработки немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг OST в регистре TZFLG, а также генерируется прерывание PWM\_TZINT, если разрешено в регистре TZEINT и контроллером прерываний. Аварийное удержание выводов заканчивается после принудительного программного сброса записью в бит OST регистра TZCLR.

Аварийное состояние выводов при получении события сбоя программируется индивидуально для выходов PWM\_A и PWM\_B полями TZA и TZB регистра TZCTL.

## 20.9 Триггер событий

### Основные функции триггера событий:

- получение событий, сформированных таймером и компаратором;
- использование информации о направлении счета (вверх-вниз);
- использование делителя событий для формирования сигнала прерывания, запросов к блокам контроллера DMA, АЦП и SDFM;
- предоставление доступа процессора к содержимому регистра флагов событий и счетчикам событий.

Триггер событий программируется посредством регистров ETSEL, ETPS, ETFLG, ETCLR и ETFRC.

### Прерывания

Функциональная схема триггера событий для генерации прерываний показана на рисунке 20.20. Триггер может генерировать прерывания (если разрешено битом INTEN регистра ETSEL) по каждому событию, а также в два, три и четыре раза реже.



Коэффициент деления событий задается полем INTPRD. Источник события выбирается полем INTSEL. Количество возникших событий отражается в поле INTCNT. Счетчик INTCNT считает от 00b до INTPRD и сбрасывается только вместе с отправкой активного прерывания.

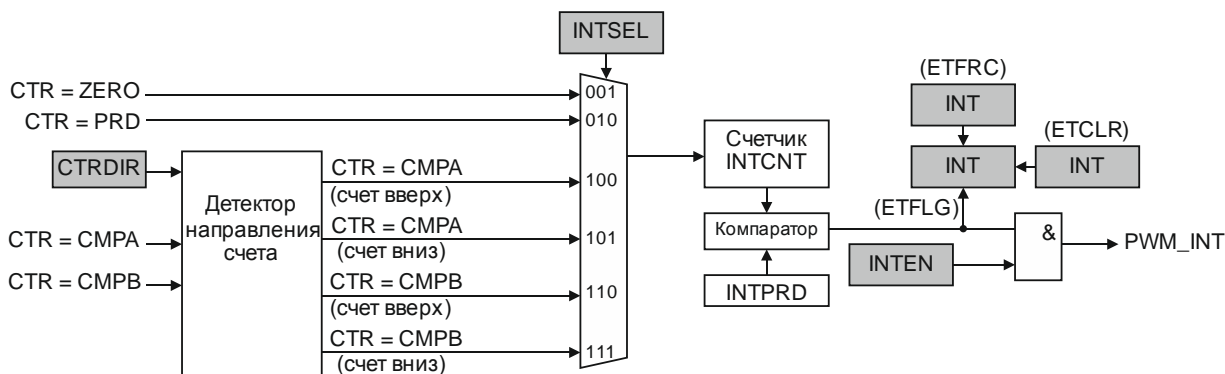


Рисунок 20.20 – Структурная схема триггера событий для генерации прерываний

Когда возникает совпадение значений счетчиков INTCNT и INTPRD, то возможны варианты:

- если прерывание разрешено и сброшен флаг INT (регистр ETFLG), то генерируется прерывание и устанавливается флаг INT, а счетчик INTCNT сбрасывается в 00b и начинает считать заново;
- если прерывание запрещено или флаг INT установлен, то счетчик перестает считать события;
- если прерывание разрешено, но флаг от предыдущего прерывания еще не сброшен, то счетчик хранит свое максимально достигнутое значение  $INTCNT = INTPRD$  до сброса флага INT. Это позволяет обработать еще прерывание, пришедшее за то время, пока обрабатывалось предыдущее.

Каждая запись в INTPRD сбрасывает счетчик INTCNT. Запись единицы в бит INT регистра ETFRC увеличит значение счетчика на единицу.

### Сопряжение с блоком DMA

Механизм сопряжения модуля ШИМ с блоком DMA реализован аналогично механизму сопряжения модуля ШИМ с триггером событий. Управление запросами блока DMA: сопряжение с сигналами PWM\_DRQA, PWM\_DRQB осуществляется через регистры ETSEL, ETPS, ETFRC аналогично механизму управления сигналом прерывания PWM\_INT. Функциональная схема триггера событий для генерации запросов блока DMA представлена на рисунке 20.21.

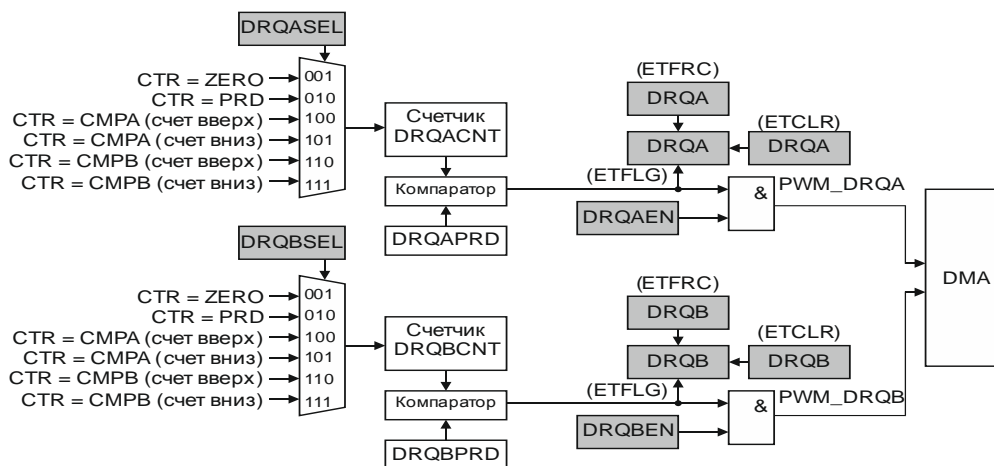


Рисунок 20.21 – Структурная схема триггера событий для генерации запросов контроллера DMA

### Сопряжение с блоками АЦП и SDFM

Управление сигналами запуска блока АЦП и SDFM PWM\_SOCА, PWM\_SOCВ осуществляется аналогично сигналу прерывания PWM\_INT и запросам PWM\_DRQA, PWM\_DRQB через регистры ETSEL, ETPS, ETFRC.

Функциональная схема триггера событий для генерации сигналов запуска АЦП или SDFM показана на рисунке 20.22.

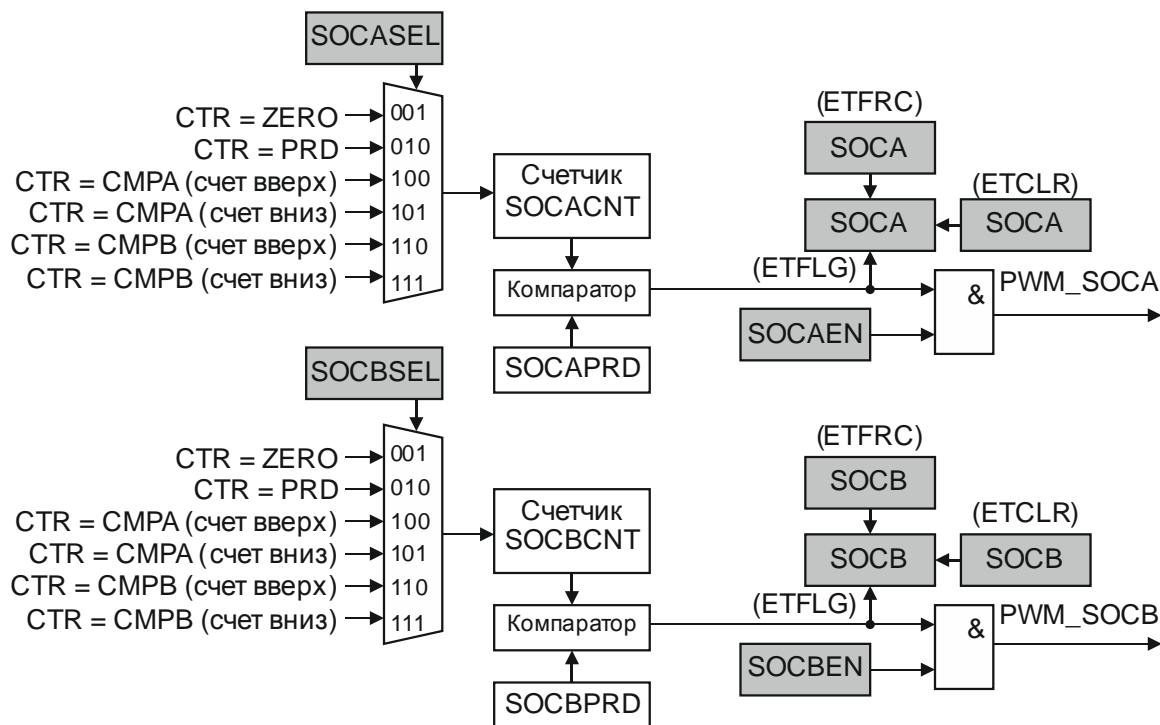


Рисунок 20.22 – Структурная схема триггера событий для генерации сигналов запуска АЦП

Выходы всех блоков ШИМ0 – ШИМ9 (PWM<sub>x</sub>\_SOCA и PWM<sub>x</sub>\_SOCB каждого блока, где x от 0 до 9) объединяются по ИЛИ, как показано на рисунке 20.21. Сигналы с выходов элементов ИЛИ защелкиваются в триггерах и формируют импульсы запуска.

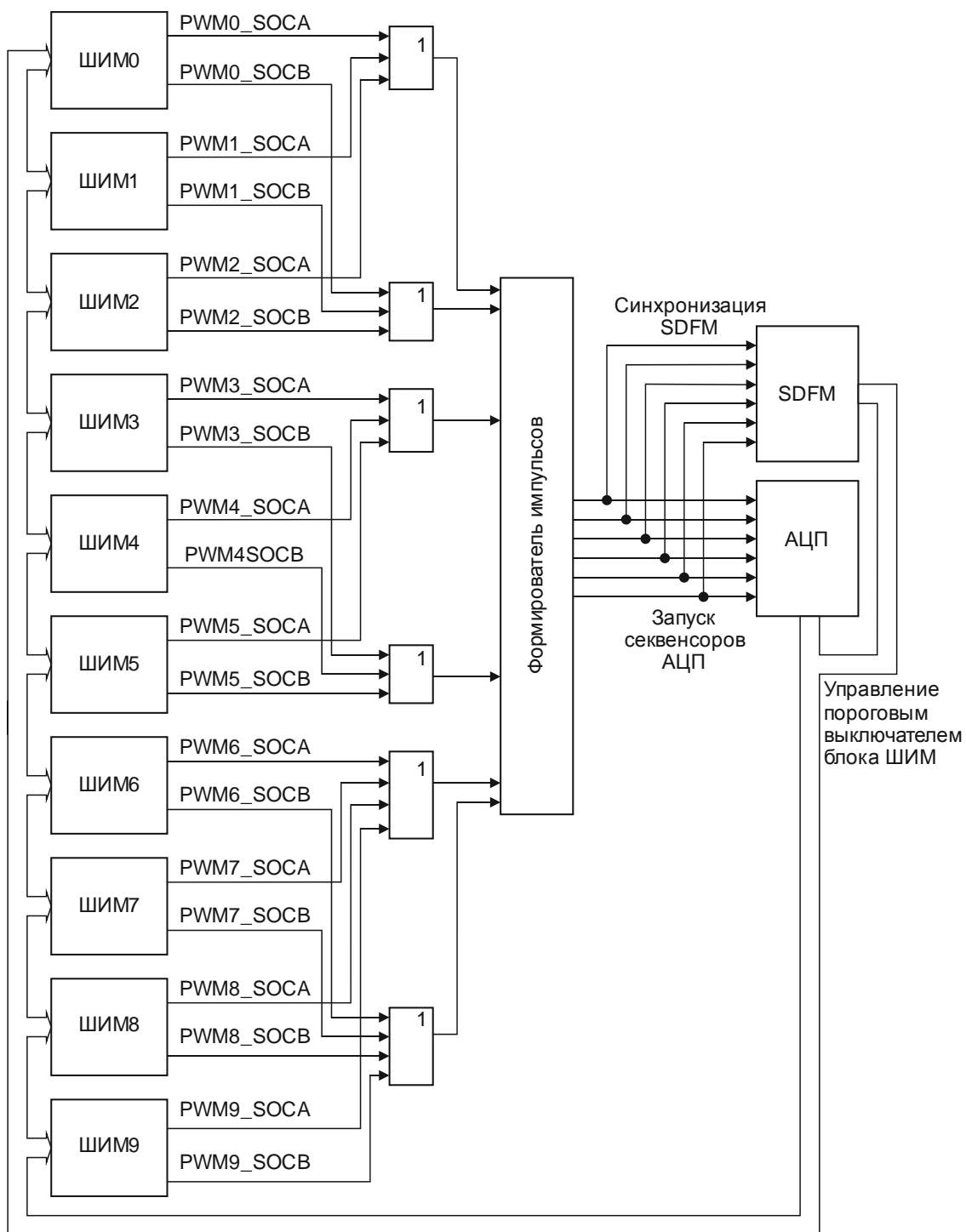


Рисунок 20.23 – Схема соединений между блоками ШИМ, АЦП и SDFM

## 20.10 ШИМ высокого разрешения

Микроконтроллер содержит шесть блоков ШИМ0-ШИМ5, поддерживающих функцию высокого разрешения.

Функция высокого разрешения блока ШИМ является дополнительной и имеет особенности:

- улучшенный контроль скважности выходного сигнала ШИМ канала А (сигнал PWMA);
- улучшенный контроль скважности выходного сигнала ШИМ канала В (сигнал PWMA);
- улучшенная точность переключения фронтов, с использованием увеличенной разрядности регистров CMPAHR и TBPNSHR (подключаются поля CMPAHR и TBPNSHR);
- логика калибровки для более точного определения значения реальной задержки.

Дополнительный регистр управления расширенными возможностями – HRCTL.

Функциональная схема функции высокого разрешения показана на рисунке 20.22.

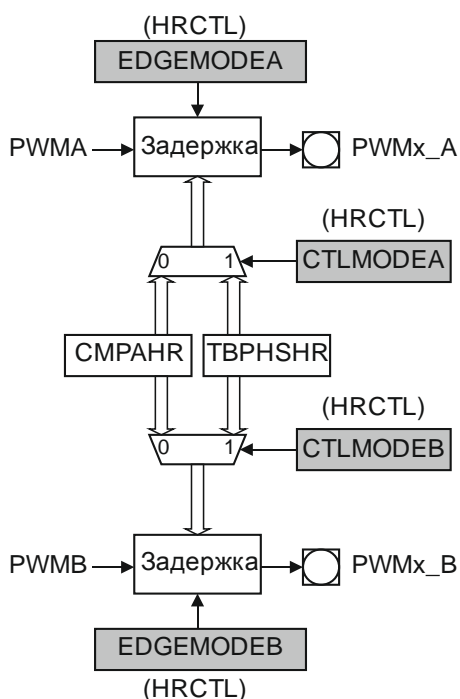


Рисунок 20.22 – Схема функции высокого разрешения

### Улучшение разрешающей способности

В блоке ШИМ высокого разрешения используется специальная линия задержки с шириной отсчета 120 пс, что увеличивает разрешающую способность. Общее время, добавляемое линией задержки, складывается из задержки выходных мультиплексоров ( $T_{MUX\_TYP}$ ) в 1.8нс и непосредственно задержки элементов ( $T_{ELEM\_TYP}$ ), по 120 пс за каждый элемент. Таким образом, при выбранной ненулевой задержке общее время в наносекундах рассчитывается по формуле:

$$T_{DEL} = 1,8 + 0,12 \times n, \quad (20.1)$$

где  $n$  – количество подключаемых элементов задержки (через поле CMPAHR или TBPNSHR).

### Конфигурация

Для каждого из выводов PWMA и PWMB возможен независимый выбор фронта, на который оказывает влияние линия задержки (поле EDGMODE). Можно установить влияние линии задержки только на передний фронт, на задний фронт или на любой из них. Также для каналов можно осуществить независимый выбор регистра, устанавливающего время задержки (поле STLMODE). Можно использовать регистр SMPAHR или TVPHSHR. Поле SMPAHR может использовать отложенную загрузку (теневой регистр), полностью аналогично тому, как это реализовано для регистра SMPA. Дополнительно полем HRLOAD регулируется событие, по которому произойдет теневая загрузка – при сброшенном HRLOAD загрузка SMPAHR из теневого регистра будет произведена по достижению счетчиком нуля, а при установленном HRLOAD загрузка будет произведена по достижению счетчиком периода.

### Ограничения диапазона скважности

Линия задержки не может работать в течение всего периода формируемого сигнала ШИМ, поэтому вводится ограничение на минимальную длительность импульса - три периода PCLK.

В случае если в соответствии с требуемой скважностью длительность импульса выходного сигнала ШИМ оказалась короче минимальной, то фронты будут выставлены с точностью, как и в обычном блоке ШИМ.

При работе на низких частотах ограничение по управлению скважностью сигнала ШИМ практически незаметно.

Суммарная задержка, генерируемая блоком ШИМ высокого разрешения, не может превышать двух периодов PCLK.

### Калибровка

Линия задержки состоит из 255 одинаковых, последовательно включенных элементов. Приблизительное время задержки одного элемента составляет 120 пс и определяется технологическими нормами, в соответствии с которыми выполнен микроконтроллер. Длительность отсчета также может незначительно меняться, в зависимости от текущего значения питания микроконтроллера и текущей температуры.

Для коррекции значения задержки в регистре HRCTL предусмотрены поля флагов DELAYCALA и DELAYCALB. Процесс калибровки заключается в настройке блока ШИМ и постепенном увеличении величины задержки с постоянным контролем флагов. Как только величина суммарной задержки превысит 1 период PCLK - DELAYCALA или DELAYCALB примут значение отличное от нуля. Далее необходимо обнулить эти поля и сделать инкремент задержки на величину в половину текущего значения. Затем продолжить инкрементировать величину задержки. Когда суммарная задержка достигнет 2 периодов, DELAYCALA или DELAYCALB примут значение 3h.

Пример: при частоте сигнала PCLK в 100 МГц ( $T_{PCLK}=10$  нс) соответствующие флаги появились при значении задержки в 68 и 148 тактов соответственно. Следовательно,

$$\begin{aligned} 1 * T_{PCLK} &= T_{MUX} + 68 * T_{ELEM}, \\ 2 * T_{PCLK} &= T_{MUX} + 148 * T_{ELEM}. \end{aligned}$$

Используя полученные равенства можно вычислить реальное время задержки мультиплексора  $T_{MUX}=1.5$  нс и реальное время задержки одного элемента  $T_{ELEM}=0,125$  нс

## 21 Модуль квадратурного декодера QEP

В микроконтроллере реализовано четыре модуля квадратурных декодеров.

Квадратурный декодер QEP преобразует цифровой сигнал с датчика положения вала, позволяя вычислять скорость, направление вращения, а также текущее положение вала.

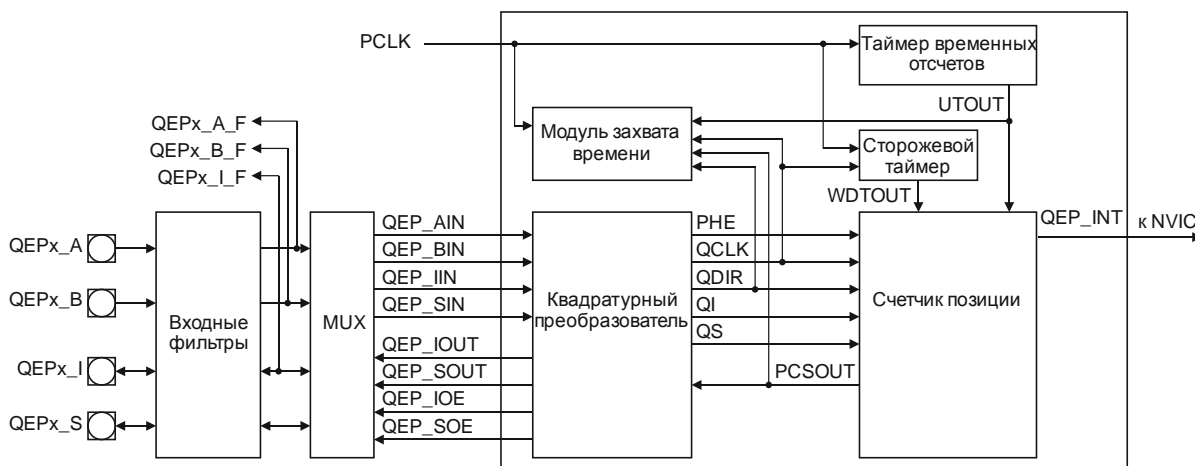


Рисунок 21.1 – Схема квадратурного декодера с мультиплексором входных/выходных сигналов

В состав квадратурного декодера QEP входят (см. рисунок 21.1):

- входные фильтры;
- настраиваемый обработчик сигналов входов;
- квадратурный преобразователь;
- счетчик позиции/блок управления;
- модуль захвата времени;
- таймер временных отсчетов;
- сторожевой таймер.

По умолчанию модуль захвата времени находится в сбросе и не тактируется. Разрешить тактирование и снять сброс можно, установив соответствующие биты в регистрах PCLKCFG и PRSTCFG блока RCU.

Входные фильтры дополняют фильтры портов ввода-вывода и могут быть настроены с помощью регистров QCQAUCTL и QCQUALSAMPLE блока SIU.

### 21.1 Обработчик сигналов входов

Квадратурный декодер QEP<sub>x</sub> (где x - номер декодера от 0 до 3) использует два квадратурных вывода микроконтроллера, работающих на вход. Также, имеются специальный индексный вывод и вывод стробирования, которые могут работать на вход и выход.

QEP<sub>x</sub>\_A и QEP<sub>x</sub>\_B – в квадратурном режиме – это два входа. Сигналы на входах сдвинуты по фазе на 90 градусов и по ним можно определить скорость и направление вращения ротора, см. рисунок 21.2. В режиме счета/направления сигналы на входах используются как тактовый и сигнал направления вращения ротора, по которым также можно вычислить скорость вращения.

QEP<sub>x</sub>\_I – индексный вывод. Сигнал на входе сигнализирует о полном обороте ротора. Позволяет сбрасывать счетчик позиции поворота ротора.

QEP<sub>x</sub>\_S – пользовательский вывод стробирования. Сигнал на входе может сбросить или защелкнуть счетчик позиции. Применяется при использовании концевых выключателей.

QEP<sub>x</sub>\_A\_F, QEP<sub>x</sub>\_B\_F, QEP<sub>x</sub>\_I\_F - прошедшие фильтрацию входные сигналы, которые могут быть заведены на входы блоков ECAP (регистр CAPMUX блока SIU).

Сигналы на входах могут быть проинвертированы. Инверсия включается установкой соответствующего бита в регистре QDECCTL.

Установкой бита SWAP можно включить обратный счет, т. е. программно подать сигнал с вывода QEPx\_A на вход QB квадратурного преобразователя, а сигнал с вывода QEPx\_B подать на вход QA (входы A и B меняются местами).

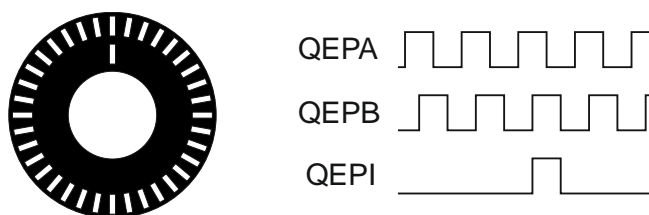


Рисунок 21.2 – Диаграмма входных сигналов

## 21.2 Квадратурный преобразователь

На рисунке 21.3 показана схема квадратурного преобразователя.

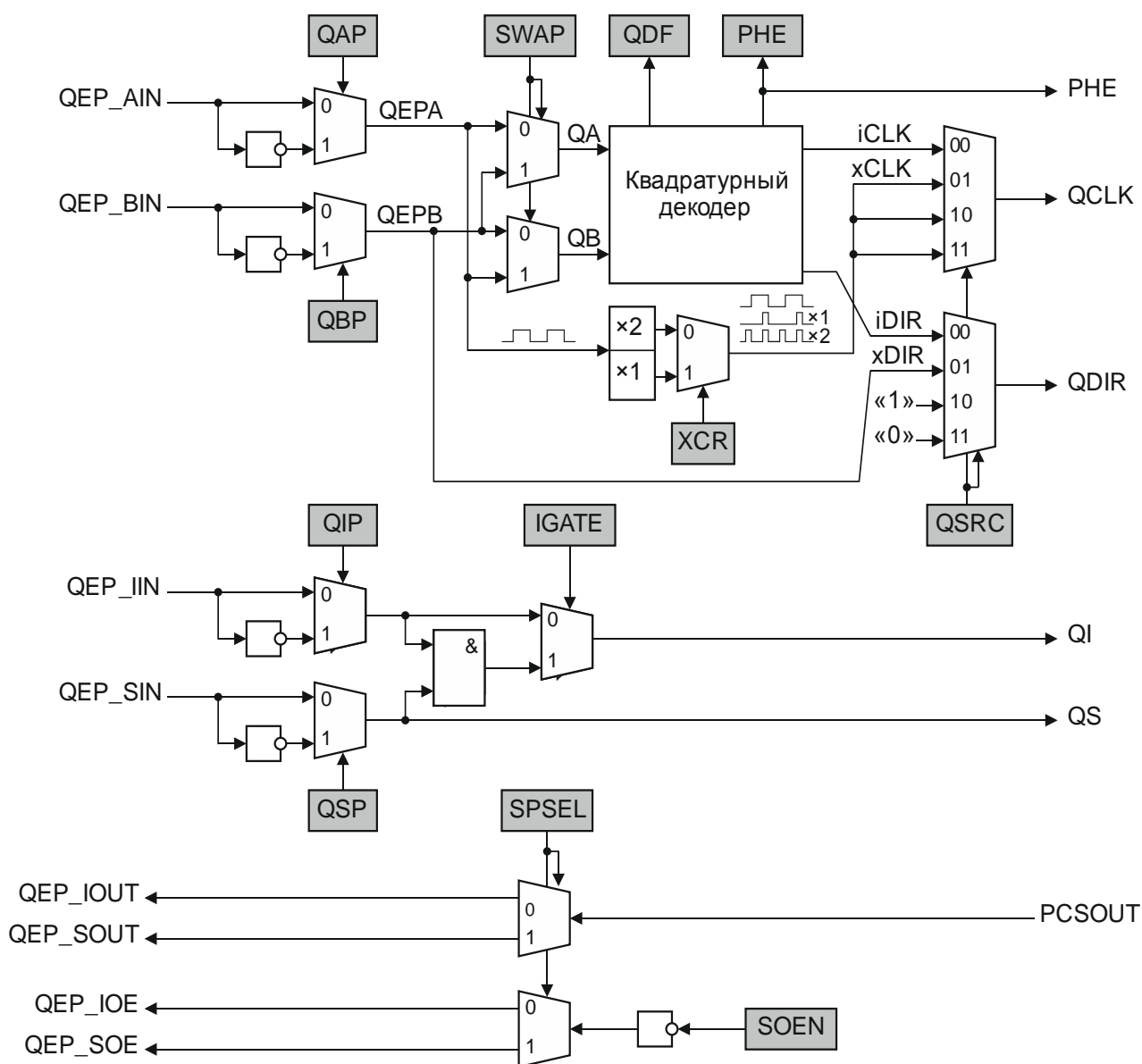


Рисунок 21.3 – Схема квадратурного преобразователя

## Режимы работы

Квадратурный преобразователь поддерживает четыре режима работы:

- режим квадратурного счета;
- режим счета/направления;
- режим счета вверх;
- режим счета вниз.

Выбор режима зависит от значения поля QSRC регистра QDECCTL.

## Режим квадратурного счета

Квадратурный преобразователь формирует сигнал направления вращения, тактовый сигнал и сигнал направления счета (вверх/вниз) для счетчика позиции.

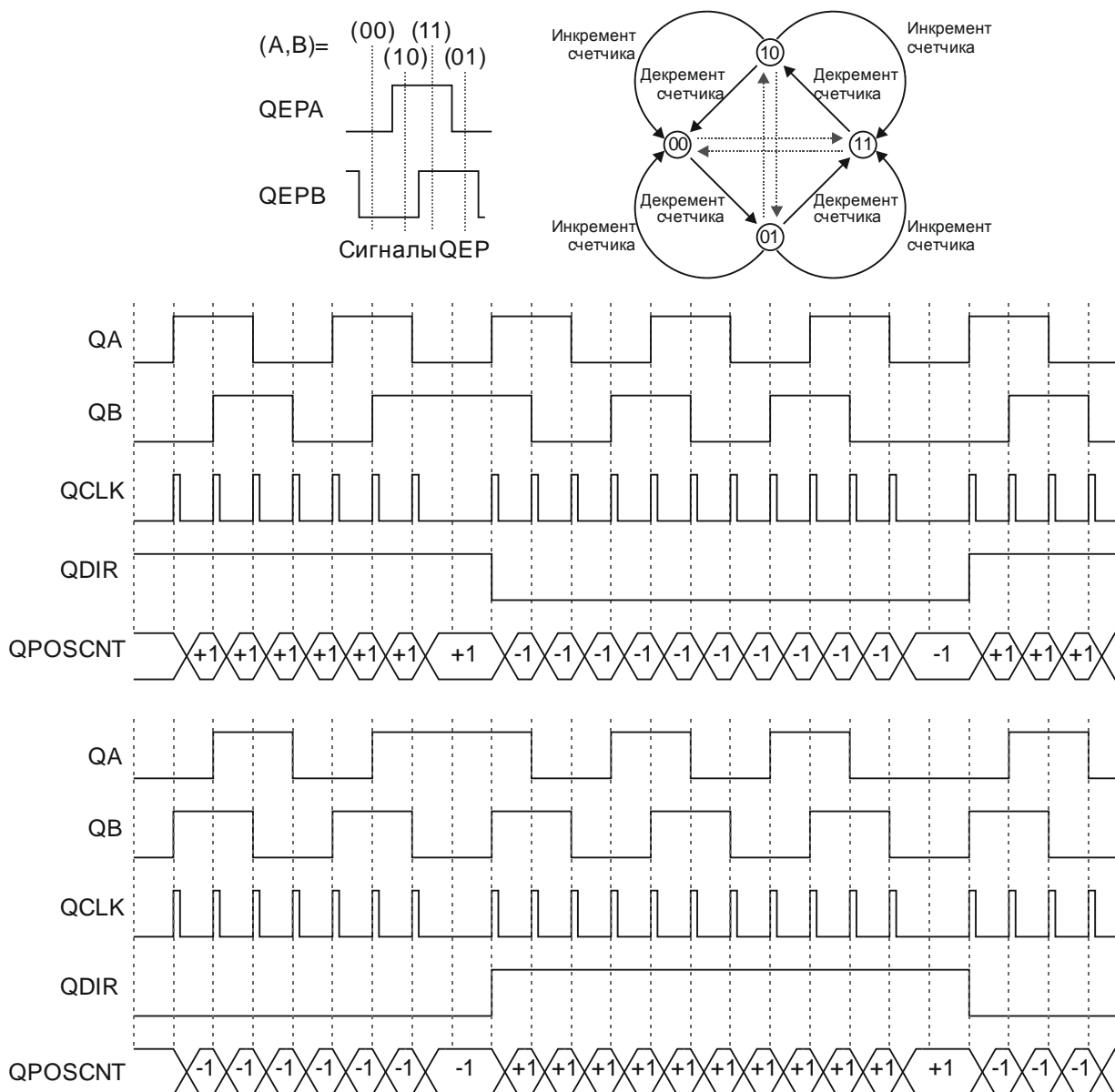


Рисунок 21.4 – Временная диаграмма и автомат состояний работы в квадратурном режиме счета

Направление вращения ротора определяется по порядку смены передних и задних фронтов на входах QEPx\_A и QEPx\_B. К примеру, если за передним фронтом сигнала на входе QEPx\_A следует передний фронт сигнала на входе QEPx\_B, см. рисунок 21.4, то направление вращения следует считать прямым, а счетчик позиции работает на увеличение. Если же за передним фронтом сигнала на входе QEPx\_B следует передний



фронт сигнала на входе QERx\_A, то направление вращения следует считать инверсным, а счетчик позиции работает на уменьшение. Если на обоих выводах зафиксировано одновременно два фронта, то такое состояние считается ошибочным.

Квадратурный преобразователь выдает четыре счетных импульса на один период входного сигнала, поскольку использует для счета передний и задний фронт сигналов.

#### **Режим счета/направления**

В этом режиме вывод QERx\_A работает как вход тактовых импульсов, а вывод QERx\_B – как вход задания направления счета. Счетчик позиции в этом режиме работает по каждому переднему фронту сигнала на входе QERx\_A.

#### **Режим вверх**

Режим используется для вычисления частоты следования импульсов на вывод QERx\_A. Фронт задается битом XCR регистра QDECCTL. Счетчик всегда работает на увеличение.

#### **Режим вниз**

Режим используется для вычисления частоты следования импульсов на выводе QERx\_A. Фронт задается битом XCR. Счетчик всегда работает на уменьшение.

### **21.3 Счетчик позиции**

Работа счетчика позиции контролируется посредством регистров QERCTL и QPOSCTL, которыми задается режим счета, сброса и хранения, а также логика для формирования внешнего сигнала синхронизации.

#### **Режимы сброса счетчика позиции**

Счетчик позиции может накапливать результат в течение многих оборотов вала, а может подсчитывать позицию только за один оборот, сбрасываясь каждый раз по событию прихода индексной метки. В зависимости от назначения могут использоваться следующие способы сброса счетчика позиции:

- по сигналу индексации;
- по переполнению;
- только по первому сигналу индексации;
- по таймеру временных отсчетов.

Режим задается полем PCRM регистра QERCTL.

Счетчик сбрасывается в ноль при его переполнении или при превышении значения регистра максимального значения QPOSMAX. Флаг прерывания, возникающего при переполнении счетчика, устанавливается в регистре QFLG.

#### **Режим сброса по сигналу индексации**

Режим сброса по сигналу индексации включен по умолчанию.

При получении сигнала с индексного вывода QERx\_I при счете вверх счетчик обнулится по следующему фронту сигнала тактирования QCLK. Если же сигнал индексации был получен при счете вниз, то в счетчик будет загружено значение QPOSMAX, см. рисунок 21.5.

При получении первого сигнала индексации, схема дожидается любого изменения на квадратурных входах и запоминает значение этого события – фронт, активный вывод QERx\_A или QERx\_B, а также направление вращения. Этот момент времени называется маркером индексации. При появлении этого события устанавливается бит FIMF регистра QEPSTS, а направление вращения сохраняется в бите FIDF. В дальнейшем, сброс счетчика будет производиться только в присутствии сигнала индексации и соответствия сохраненным значениям маркера индексации. Если направление вращения изменится, то

для формирования сброса сохраненное в маркере значение фронта (передний или задний фронт) меняется на обратное. Это сделано с целью привязки индексации к квадратурному сигналу QA/QB, а также для более точной обработки индексации, чтобы исключить влияние ширины импульса на индексный вывод. Сохраненные значения используются при сбросе по маркеру индексации (если поле IEL в регистре QEPCTL равно 11b).

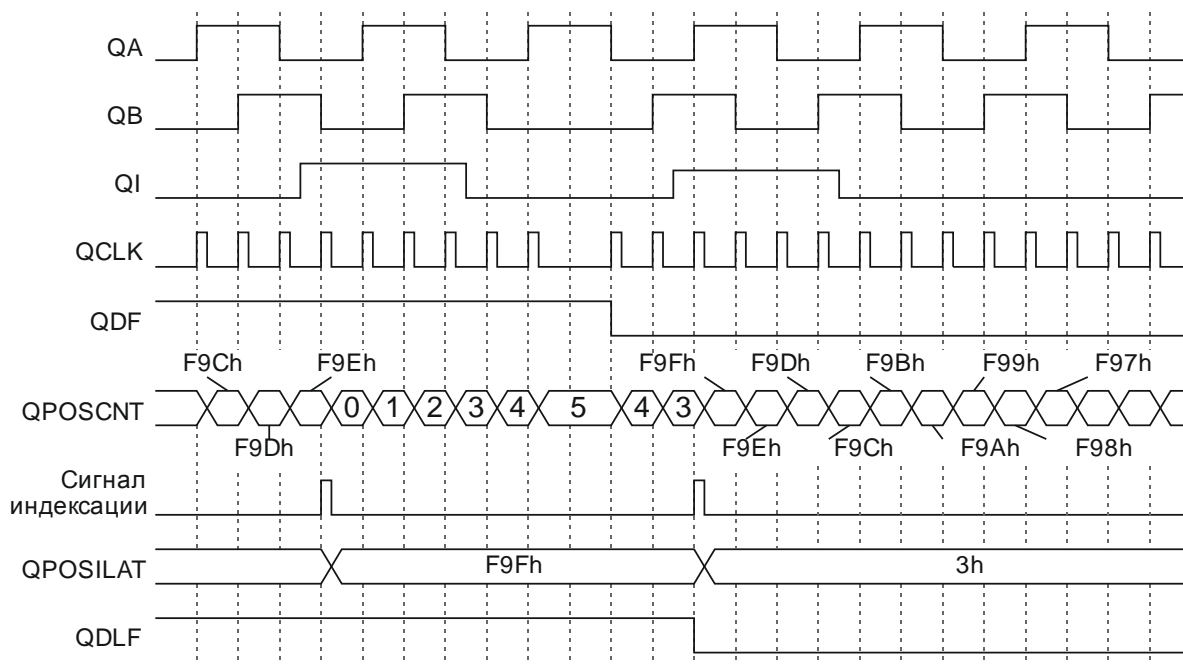


Рисунок 21.5 – Временные диаграммы сброса по сигналу индексации

По каждому сигналу индексации, включая маркер индексации, содержимое счетчика сохраняется в регистре QPOSILAT, а направление вращения – в бите QDLF регистра QEPSTS. Если при сохранении значение счетчика QPOSCNT не равно ни нулю, ни значению QPOSIMAX, то выставляется флаг ошибки счетчика позиции (бит PCEF в регистре QEPSTS) и флаг прерывания (бит PCE в регистре QFLG). Флаг ошибки счетчика позиции обновляется с каждым индексом, а флаг прерывания может быть сброшен только программно.

Поле настройки события индексации для сохранения счетчика позиции IEL (регистр QEPCTL) игнорируется. Также только в этом режиме могут устанавливаться флаг ошибки счетчика позиции PCEF и флаг соответствующего прерывания.

### Режим сброса по переполнению

Максимальное значение счетчика позиции задается регистром QPOSIMAX. Если счетчик считает вверх и достигнуто максимальное значение, то со следующим тактом синхросигнала счетчик обнулится. Если счетчик считает вниз и достигнуто значение нуля, то со следующим тактом синхросигнала в счетчик будет загружено значение QPOSIMAX. Сброс по событию индексации не производится.

Получение значений маркера индексации происходит аналогично тому, как это происходит в режиме сброса по сигналу индексации. Полученные значения могут использоваться при инициализации по маркеру индексации, если в поле IEL записано значение 11b в регистре QEPCTL. Временная диаграмма сброса показана на рисунке 21.6.

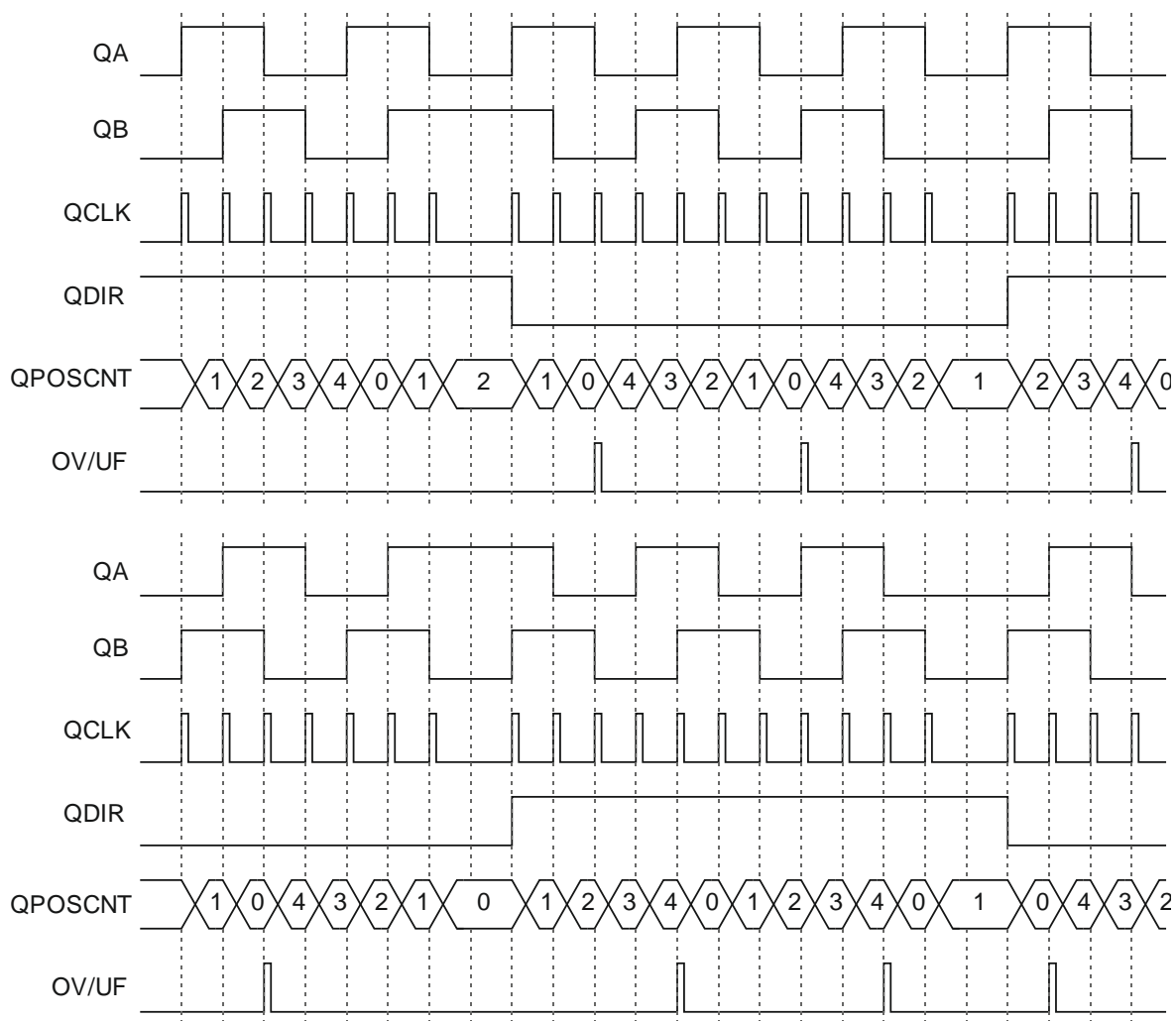


Рисунок 21.6 – Временные диаграммы сброса по сигналу переполнения

### Режим сброса по первому сигналу индексации

Если было получено событие индексации при счете вверх, то счетчик обнулится со следующим тактом синхросигнала. Если же событие индексации было зафиксировано при счете вниз, то со следующим тактом синхросигнала в счетчик будет загружено значение QPOSMAX. При последующем счете сброс может произойти только при достижении нуля или значения QPOSMAX (т.е. аналогично режиму сброса по переполнению), а дальнейшие возможные события получения сигнала на выводе индексации влиять на сброс не будут.

Получение значений маркера индексации происходит аналогично тому, как это происходит в режиме сброса по сигналу индексации. Полученные значения могут использоваться при инициализации по маркеру индексации, если в поле IEL записано значение 11b.

### Режим сброса по таймеру временных отсчетов

В этом режиме счетчик сбрасывается в ноль или загружается значением QPOSMAX, в зависимости от текущего режима счета (задается полем QSRC регистра QDECCTL), по событию срабатывания таймера временных отсчетов. В остальном режим аналогичен режиму сброса по переполнению.

Также возможно настроить сохранение значения счетчика QPOSCNT в регистр QPOSLAT перед сбросом, для этого необходимо включить модуль захвата, установив бит CEN в регистре QCAPCTL. Этот режим удобен для измерения частоты.

### **Сохранение счетчика позиции**

Внешние входы индексации и стробирования можно запрограммировать на формирование событий для сохранения значения счетчика позиции в регистры QPOSILAT и QPOSSLAT.

### **Сохранение по событию индексации**

В некоторых задачах не требуется сбрасывать счетчик позиции по каждому сигналу индексации, а вместо этого может потребоваться увеличить разрядность счетчика до 32 бит (режимы, задаваемые значениями PCRM равными 01b и 10b). В этом случае бит QDLF (направление вращения) в регистре QEPSTS будет перезаписываться по каждому сигналу индексации, а счетчик будет сохранять значение по следующим событиям индексации:

- по переднему фронту сигнала индексации при IEL = 01b;
- по заднему фронту сигнала индексации при IEL = 10b;
- по маркеру индексации при IEL = 11b.

Сохранение значения счетчика по маркеру индексации будет производиться только в присутствии сигнала индексации и по событию, эквивалентному сохраненному при первой индексации по маркеру. Если направление вращения изменится, то сохраненное в маркере значение типа фронта меняется на обратное значение. Это сделано с целью привязки индексации к квадратурному сигналу QA/QB, а также для более точной обработки индексации, чтобы исключить влияние ширины импульса на индексном выводе.

При сохранении значения счетчика в регистр QPOSILAT формируется флаг IEL прерывания индексации в регистре QFLG. В режиме сброса по сигналу индексации (PCRM = 00h) значение поля IEL в регистре QEPCTL игнорируется.

### **Сохранение по событию стробирования**

Значение счетчика сохраняется в регистр QPOSSLAT по каждому переднему фронту сигнала на входе QEP\_S, если сброшен бит SEL в регистре QEPCTL. Если же бит SEL установлен, то сохранение в регистре QPOSSLAT происходит по переднему фронту сигнала строба на входе QEP\_S при прямом направлении вращения и по заднему фронту для обратного вращения. При каждом сохранении счетчика в регистре QPOSSLAT устанавливается флаг прерывания SEL.

### **Инициализация счетчика позиции**

Счетчик событий может быть проинициализирован программно или по событиям:

- событие индексации;
- событие стробирования.

Входной сигнал индексации QEP1 может использоваться для инициализации счетчика по переднему и заднему фронту. Если поле IEI = 10b, то счетчик загружается значением регистра QPOSINIT по переднему фронту сигнала индексации. Аналогично, если IEI = 11b, то счетчик загружается значением QPOSINIT по заднему фронту сигнала индексации.

Если поле SEI = 10b, то счетчик загружается значением регистра QPOSINIT по переднему фронту сигнала стробирования на входе QEPx\_S. Если SEI = 11b, то счетчик загружается значением регистра QPOSINIT по переднему фронту сигнала стробирования, если идет счет вверх, и по заднему – если вниз.

Программно счетчик инициализируется при записи единицы в бит SW1 регистра QEPCTL. Бит не сбрасывается автоматически, но повторная запись единицы также приведет к инициализации счетчика.

## Компаратор счетчика позиции

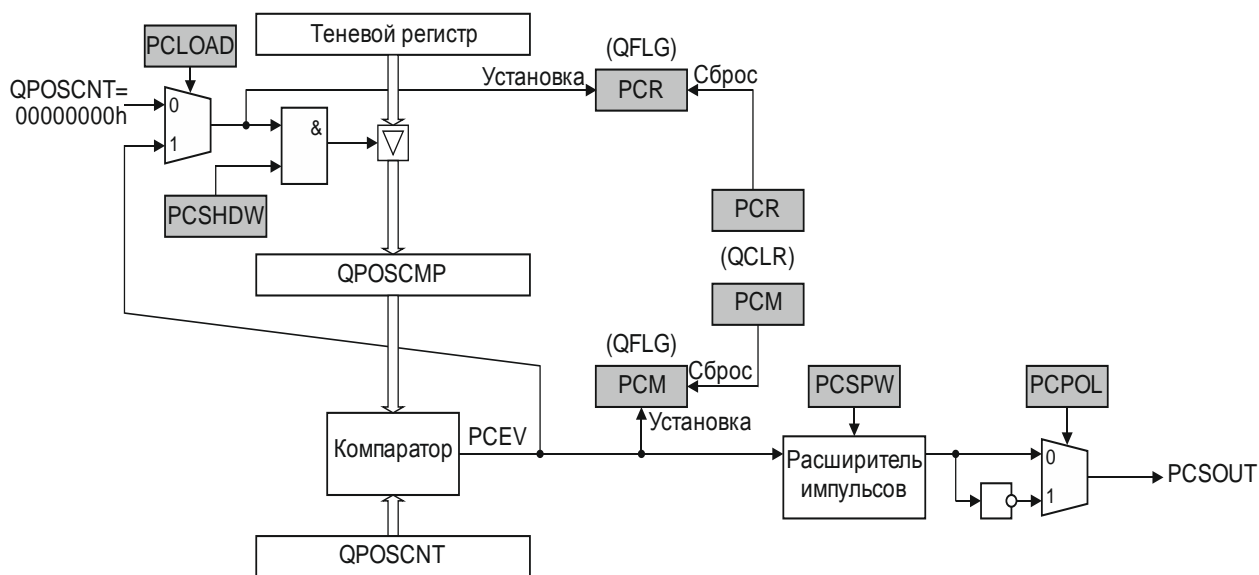


Рисунок 21.7 – Функциональная схема компаратора счетчика позиции

Компаратор, см. рисунок 21.7, сравнивает значение счетчика позиции с регистром QPOSCMP и при совпадении значений формирует прерывание, а также внешний синхросигнал, который может быть направлен на один из выводов: индексный вывод QEP\_I или вывод стробирования QEPx\_S. Бит SPSEL в регистре QDECCTL определяет, на какой именно вывод будет направлен сигнал синхронизации, а бит SOEN в регистре QDECCTL разрешает этому выводу работать как выход.

Регистр QPOSCMP может использовать режим отложенной загрузки, когда отложенное значение берется из теневого регистра, а если режим отложенной загрузки выключен, то запись в QPOSCMP производится сразу в активный регистр.

Отложенная загрузка происходит по следующим событиям:

- по совпадению результатов сравнения;
- по обнулению счетчика QPOSCNT.

Флаг успешного сравнения PCM устанавливается, когда выполняется условие  $QPOSCNT = QPOSCMP$ , при этом также формируется синхроимпульс требуемой длительности для извещения внешнего устройства (сигнал PCSOUT). Настраиваемая длительность синхроимпульса контролируется специальной схемой задержки.

Флаг PCR готовности компаратора к отложенной загрузке значения сравнения выставляется, когда выполняется условие для отложенной записи, заданное битом PCLOAD в регистре QPOSCTL. При этом генерация соответствующего прерывания и установка флага PCR будет осуществляться лишь при включении режима отложенной загрузки PCSHDW (регистр QPOSCTL).

## 21.4 Таймер временных отсчетов

Таймер, используемый для оповещения программного обеспечения о необходимости начать измерение скорости, представляет собой 32-разрядный таймер, работающий на частоте системного тактового сигнала. Таймер включается установкой бита UTE в регистре QEPCTL. Когда значение таймера достигает порога ( $QUTMR = QUPRD$ ), формируется прерывание и выставляется флаг UTO. Данный блок таймера может быть использован для вычисления скорости при высоком быстродействии, см. рисунок 21.8, а также для сохранения счетчика позиции, регистра таймера и регистра периода в регистрах QPOSLAT, QCTMRLAT и QCPRDLAT, соответственно. Режим сохранения определяется состоянием бита QCLM в регистре QEPCTL.

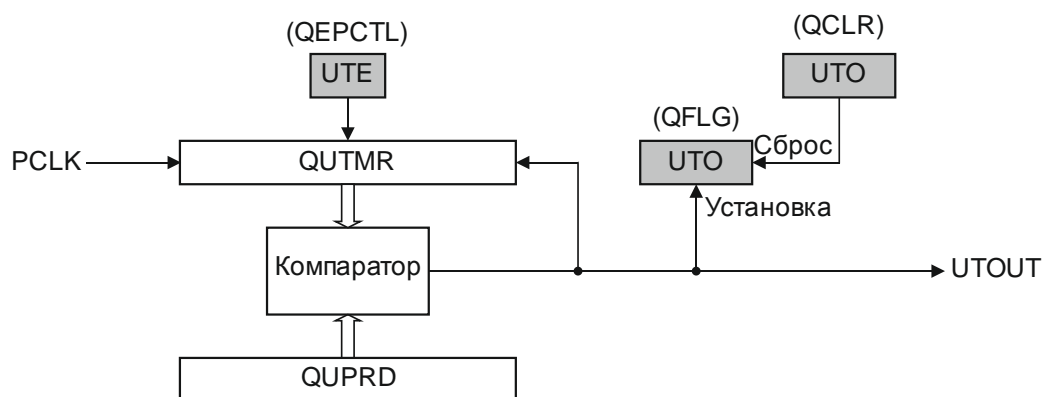


Рисунок 21.8 – Функциональная схема таймера временных отсчетов

## 21.5 Модуль захвата времени

Функциональная схема модуля захвата времени представлена на рисунке 21.9. Таймер использует тактовый сигнал и сигнал квадратурных событий с коэффициентами деления, программируемыми полями CCPS и UPPS в регистре QCAPCTL. Коэффициенты деления можно менять в процессе работы, но в этом случае может произойти событие захвата, содержащее неверные данные, которые следует игнорировать. Во избежание подобной ситуации перед изменением значений коэффициентов деления следует выключить модуль захвата времени (сбросить бит SEN), и снять все маски прерываний. После изменения коэффициентов проинициализировать таймер QCTMR (записать нулевое значение), сбросить все статусы, вновь разрешить прерывания и включить модуль захвата.

Также, существует возможность включить улучшенный режим теневой загрузки UPPS/CCPS. Если установить бит EPSLD в регистре QCAPCTL, то изменение UPPS/CCPS в процессе работы будет происходить следующим образом:

- при записи, значение UPPS/CCPS занесется в теневой регистр
- по ближайшему, после записи, событию QCLK: новое значение делителя примет силу, сбросится QCTMR, сбросится счетчик предделителя UPPS, сбросится UPEVNT.

Таким образом, в этом режиме – следующее событие (установка UPEVNT и захват QCTMR в QCPRD), которое будет следовать после изменения UPPS/CCPS, будет уже корректным. Без необходимости осуществлять пропуски данных или дополнительных программных манипуляций.

Если бит SELEVNT в регистре QCAPCTL сброшен, то по деленному квадратурному событию значение таймера QCTMR загружается в регистр периода QCPRD, после чего таймер сбрасывается, и устанавливается флаг UPEVNT в регистре QEPSTS, означающий обновление регистра QCPRD. Флаг сбрасывается программно записью единицы.

При установленном бите SELEVNT обновление регистра периода происходит по сигналу от выхода компаратора PCSOUT.

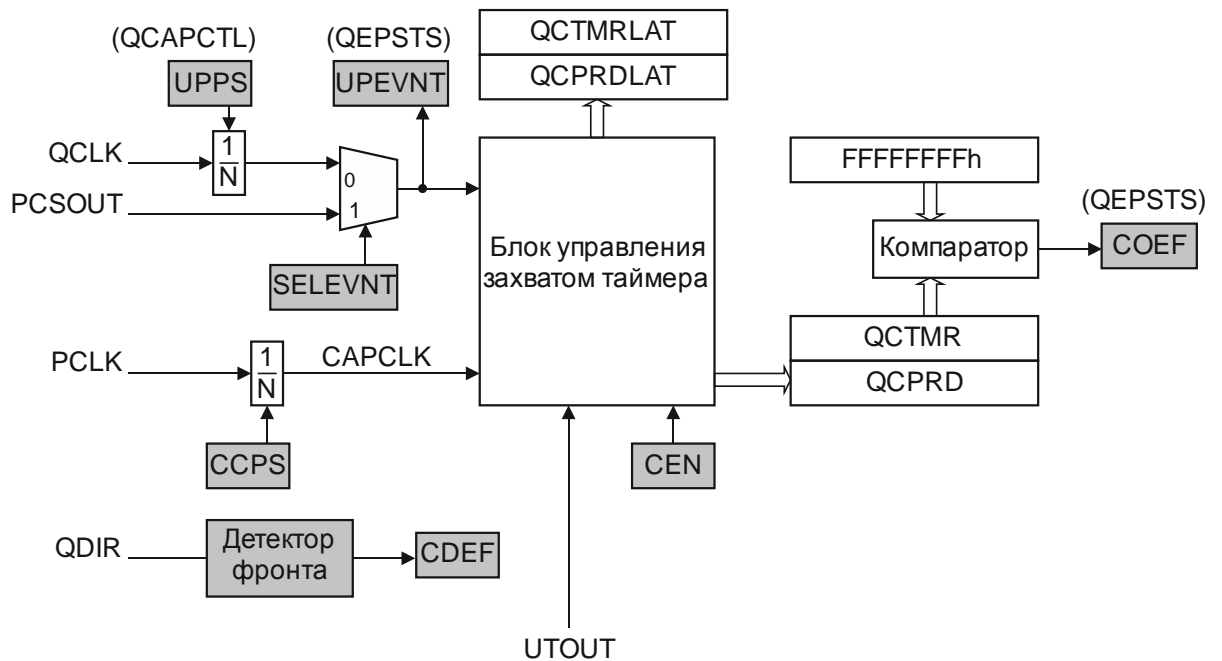


Рисунок 21.9 – Функциональная схема модуля захвата времени

Значение таймера можно использовать при измерениях скорости, если:

- его значение не превысило FFFF\_FFFFh;
- направление вращения за время измерения не изменилось.

Если между двумя событиями UPEVNT (т. е. во время измерения) таймер QCTMR переполнился, устанавливается флаг ошибки COEF в регистре QEPTS. Если между двумя событиями положения вала изменилось направление вращения, устанавливается флаг ошибки CDEF.

Значения таймера QCTMR и регистра периода QCPRD могут быть сконфигурированы для захвата в регистры QCTMRLAT и QCPRDLAT по событиям:

- прочитан регистр QPOSCNT;
- сработал сторожевой таймер.

Если бит QCLM сброшен, то при каждом чтении регистра счетчика позиции QPOSCNT регистры QCTMR и QCPRD загружаются в регистры QCTMRLAT и QCPRDLAT, соответственно.

Если бит QCLM установлен, то при каждом срабатывании таймера временных отсчетов счетчик позиции, регистр таймера и регистр периода захватываются в регистры QPOSLAT, QCTMRLAT и QCPRDLAT, соответственно.

Измерения на малых скоростях вращения (низкая частота квадратурного сигнала) производятся следующим образом – таймер QCTMR, тактирующийся от системного тактового сигнала с делителем CCPS, по событию UPEVNT, сохраняет свое значение времени в регистре QCPRD, одновременно сбрасывается и выставляет флаг UPEVNT в регистре QEPTS, чтобы сообщить программе об окончании измерения. Событие UPEVNT возникает каждые несколько тактов QCLK, в соответствии с запрограммированным коэффициентом деления UPPS в регистре QCAPCTL. Таким образом, зная количество квадратурных событий за измеренный отрезок времени, а также такой параметр, как количество квадратурных событий за полный оборот вала, можно вычислить скорость вращения.

Измерения на высоких скоростях, см. рисунок 21.10, могут производиться иначе. Таймер временных отсчетов формирует общую длительность измерения, счетчик позиции подсчитывает количество импульсов QCLK. Зная количество импульсов QCLK за один полный оборот, можно вычислить скорость вращения вала.

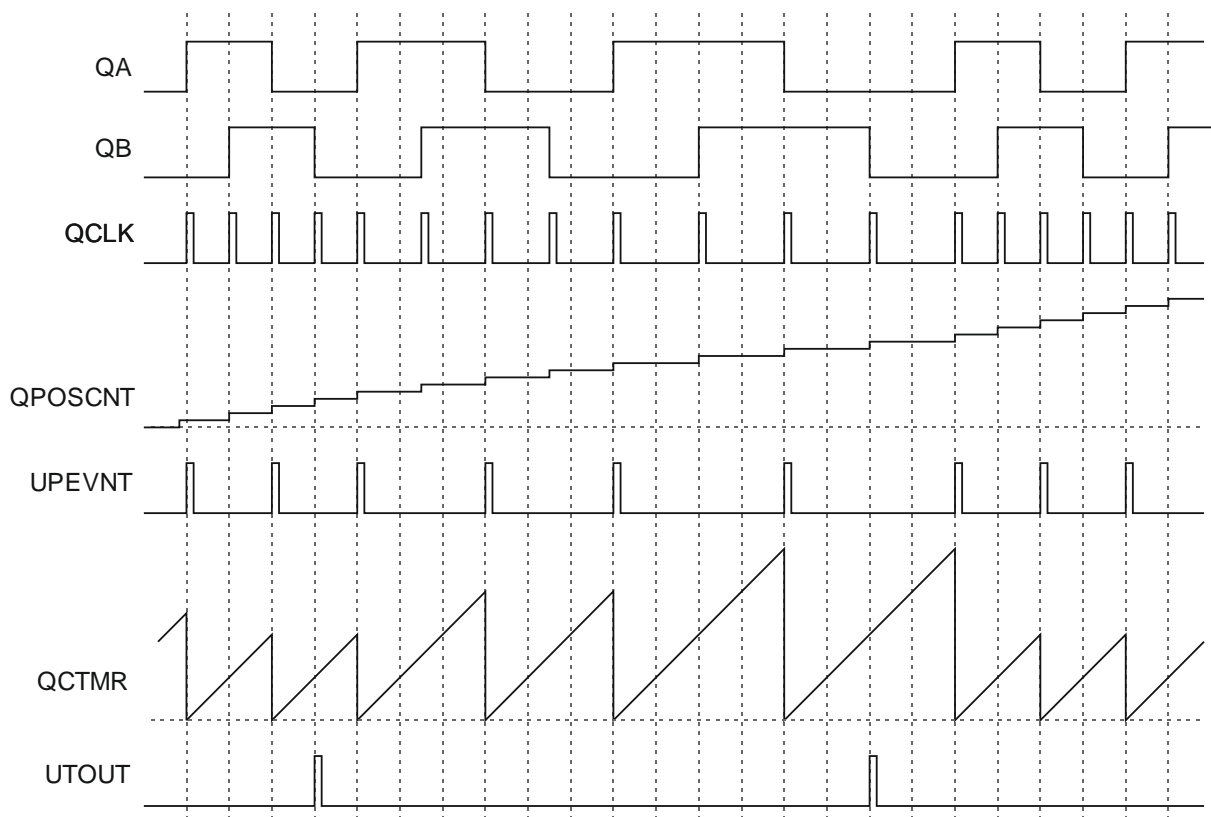


Рисунок 21.10 – Работа на высоких скоростях

Также существует и смешанный способ измерения скорости – по заданному значению счетчика позиции, с помощью компаратора счетчика позиции, можно сформировать событие UPEVNT (необходимо установить бит SELEVNT в регистре QCAPCTL), которое, так же как и при измерениях на малых скоростях, позволит получить значение таймера QCTMR. Для использования этого способа измерения скорости необходимо разрешить прерывание по событию PCSOUT компаратора и устанавливать в этом прерывании каждый раз порог сравнения компаратора QPOSCMP на заданное количество меток вперед по сравнению с текущей позицией счетчика QPOSCNT (в зависимости от направления вращения). Тогда, устанавливая QPOSCMP дальше от QPOSCNT с увеличением скорости вращения, можно поддерживать оптимальное захватываемое время, обеспечивающее максимальную точность измерения времени для всех диапазонов вращения. Этот способ измерения наиболее сложен, но и наиболее универсален.

Дополнительно, модуль захвата времени способен генерировать запросы DMA. Чтобы разрешить генерацию необходимо установить бит DMAEN в регистре DMAREQ. Затем, каждый раз, когда флаг UPEVNT (регистра EPSTS) переходит из 0 в 1, будет генерироваться запрос. Если UPEVNT будет оставаться несброшенным, то запросы генерироваться не будут. При этом, когда происходит чтение QCPRD при DMAEN = 1, флаг UPEVNT сбрасывается автоматически.

## 21.6 Сторожевой таймер блока QEP

Блок квадратурного декодера QEP содержит 32-битный сторожевой таймер, который тактируется системным тактовым сигналом, деленным на 64, и сбрасывается любым квадратурным событием (перепад на выводе QEPx\_A/QEPx\_B). Если ни одного квадратурного события не было зафиксировано до события QWDTMR = QWDPRD, сторожевой таймер формирует флаг прерывания WTO в регистре QFLG. Регистр QWDPRD содержит значение срабатывания сторожевого таймера. Функциональная схема сторожевого таймера представлена на рисунке 21.11.



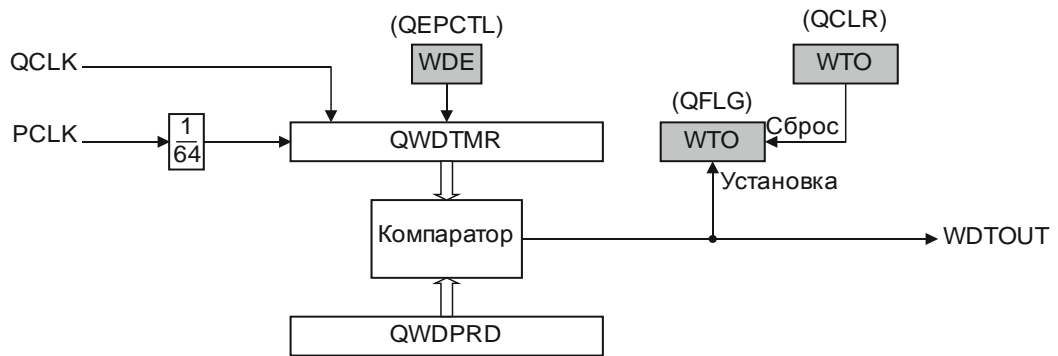


Рисунок 21.11 – Функциональная схема сторожевого таймера

## 21.7 Прерывания

Блок квадратурного декодера QEP содержит 11 источников прерываний, см. рисунок 21.12. Система прерываний состоит из регистра маски прерываний QEINT, регистра флагов прерываний QFLG, а также схемы формирования внешнего прерывания QEP\_INT по наличию активных флагов. Прерывание INT также может быть маскировано в контроллере прерывания NVIC. Сброс флагов прерываний осуществляется через регистр QCLR. Сброс флага активности прерывания INT осуществляется записью в регистр INTCLR. Также, прерывание можно сформировать программной записью в регистр QFRC, но для этого необходимо предварительно включить счетчик позиции, установив бит QPEN в регистре QEPCTL.

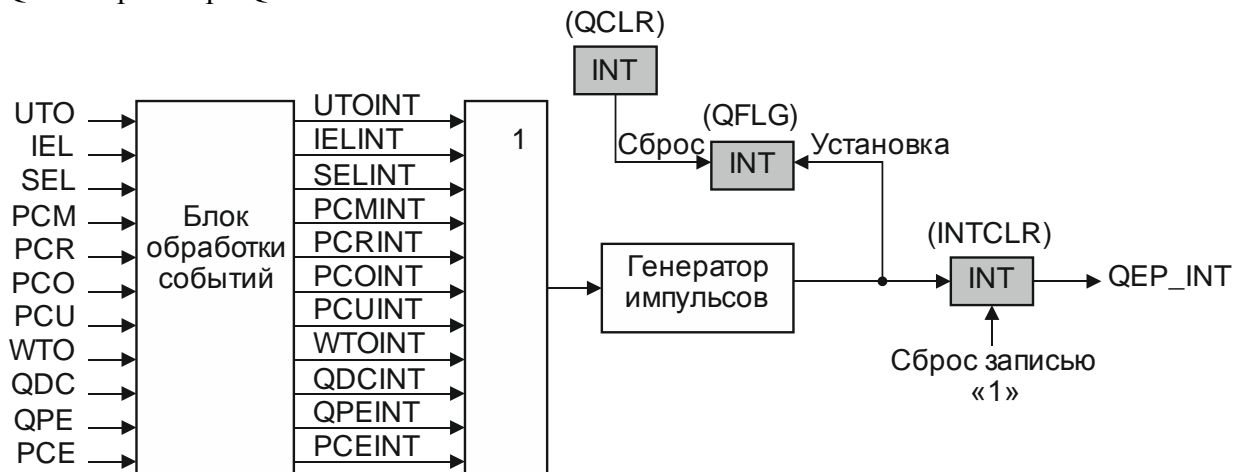


Рисунок 21.12 – Схема системы прерываний

На рисунке 21.13 показана схема формирования прерывания внутри блока обработки событий для события UTO (срабатывание таймера временных отсчетов). Схемы формирования прерываний для остальных событий идентичны.

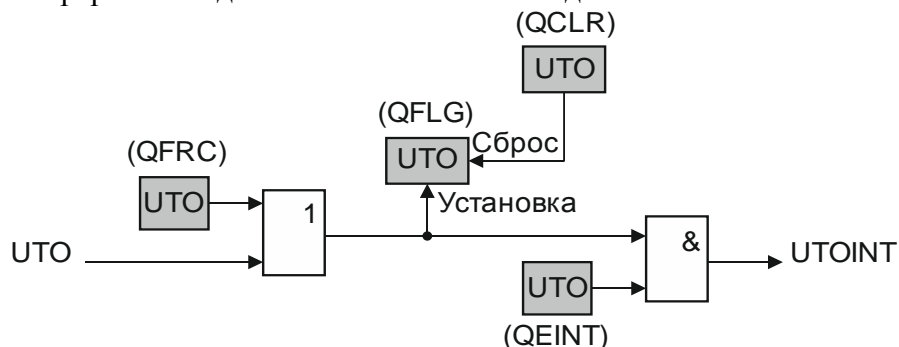


Рисунок 21.13 – Схема формирования прерывания UTOINT

## 22 Сигма-дельта демодулятор SDFM

Сигма-дельта демодулятор (SDFM) – это четырехканальный цифровой фильтр, спроектированный специально для решения задач управления электроприводом, таких как измерение токов и определение позиции с помощью вращающихся трансформаторов (резольверов).

Каждый из каналов может принимать независимый дельта-сигма модулированный поток бит. Битовые потоки обрабатываются индивидуально настраиваемыми цифровыми фильтрами и также могут быть заведены на специальные компараторы, с помощью которых можно отслеживать события превышения и занижения тока, а также моменты перехода тока через ноль.

Основные возможности:

- 8 внешних выводов, состоящих из 4 пар тактового сигнала и данных (SDFM<sub>x</sub>\_CLK и SDFM<sub>x</sub>\_DATA, где x равен 0 – 3);
- поддержка 4 режимов тактирования модулятора;
- наличие 4 основных независимых фильтров данных;
- наличие 4 компараторов - дополнительных фильтров с функцией сравнения данных с пороговыми значениями, позволяющих отслеживать события превышения и занижения тока, а также моменты перехода тока через ноль;
- выбор типа фильтра (Sinc1, Sinc2, Sincfast, Sinc3, Sinc4, Sinc5);
- конфигурируемое значение децимации;
- синхронизация работы фильтров с блоками ШИМ;
- фильтры данных имеют встроенное FIFO в размером 8×32 бит;
- генерация запросов DMA по настраиваемому уровню заполнения FIFO;
- наличие прерываний, как от фильтров данных, так и от компараторов.

### 22.1 Структура демодулятора

Сигма-дельта демодулятор имеет структуру, показанную на рисунке 22.1.

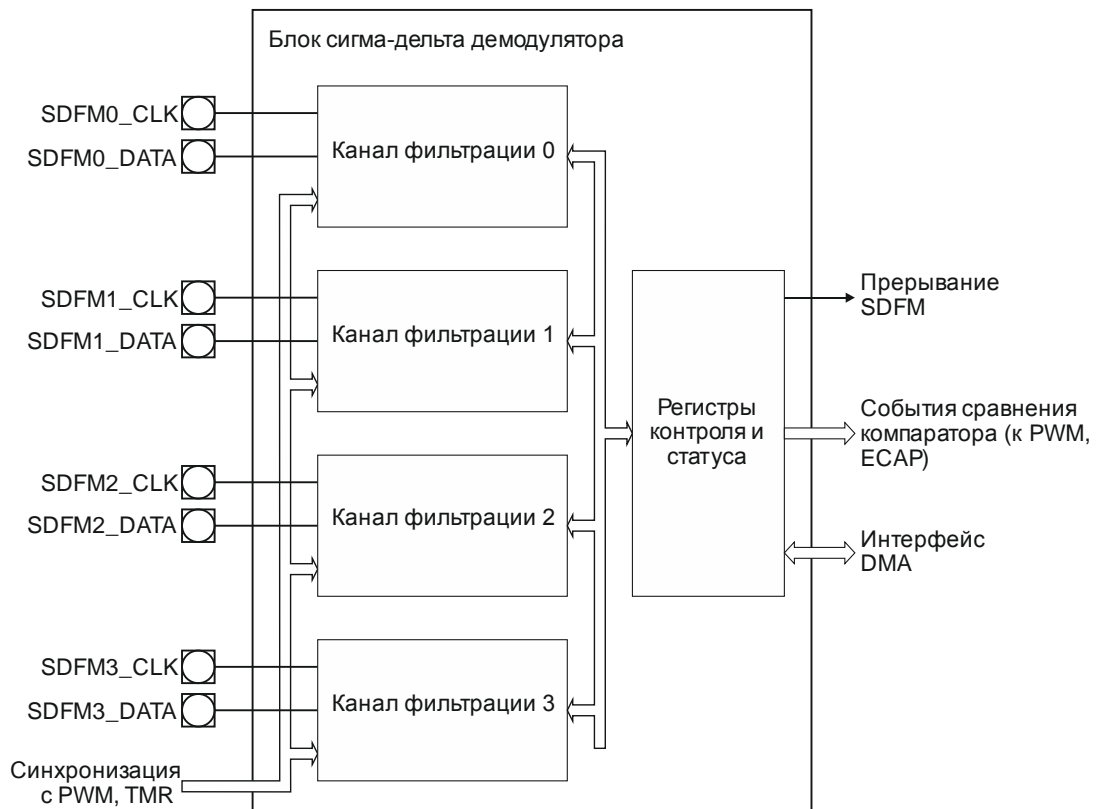


Рисунок 22.1 - Структурная схема сигма-дельта демодулятора

Главным образом, демодулятор состоит из четырех каналов фильтрации, которые могут конфигурироваться независимо друг от друга. Каналы, соответственно, включают в себя:

- входной декодер;
- фильтр данных (основной фильтр);
- компаратор (дополнительный фильтр с функцией сравнения).

Функциональная схема канала фильтрации демодулятора показана на рисунке 22.2. Как видно из схемы, фильтр данных и компаратор используют один и тот же битовый поток, однако, конфигурируются и обрабатывают его независимо.

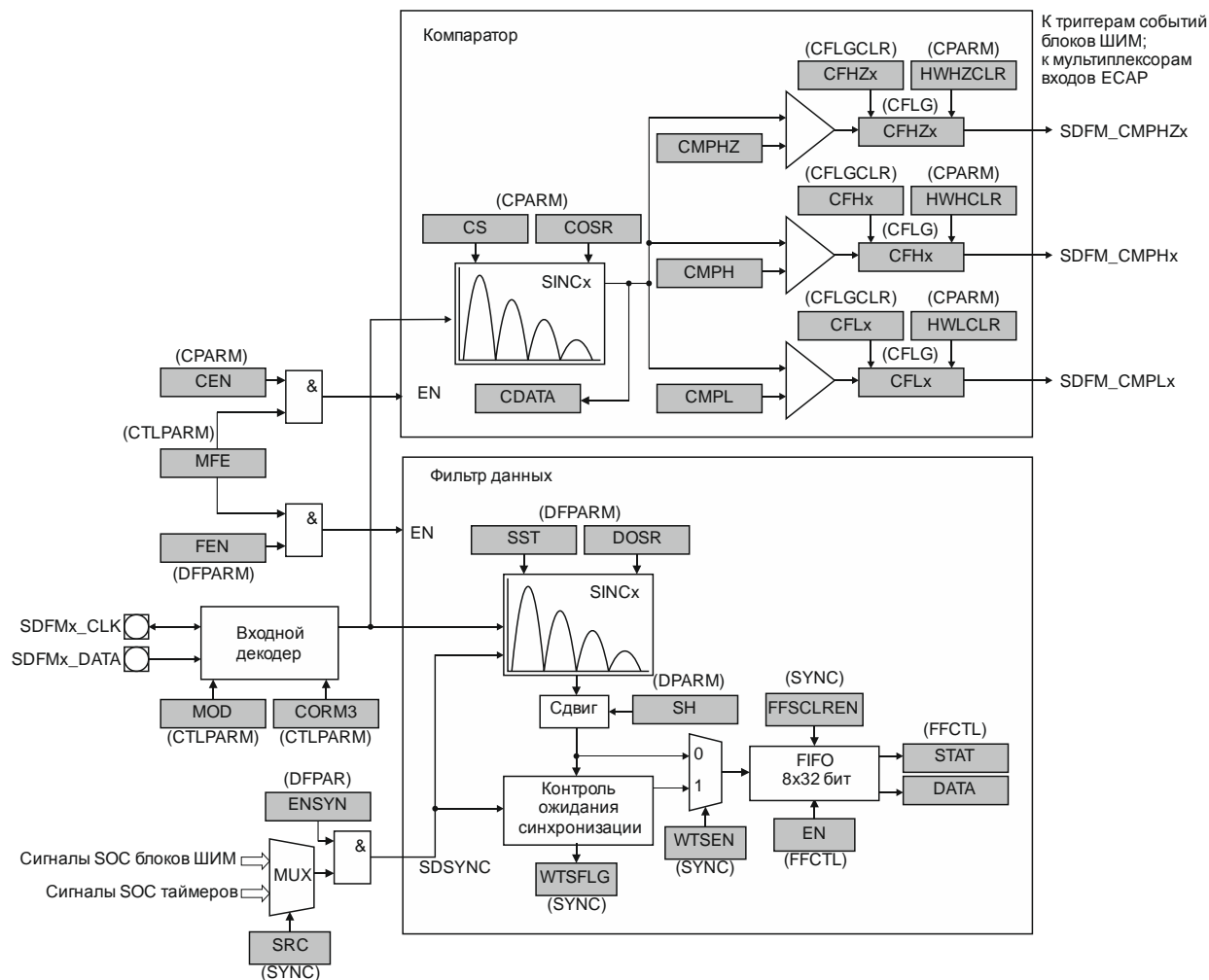


Рисунок 22.2 – Функциональная схема канала x (x от 0 до 3) демодулятора

## 22.2 Входной декодер

Входной декодер осуществляет первичную обработку поступающего сигма-дельта модулированного битового потока. Модулятор может применять различные виды как синхронной, так и асинхронной передачи данных – выбор режима декодирования осуществляется с помощью поля MOD регистра CTLPARM. Поддерживаемые режимы тактирования модулятора представлены в таблице 22.1.

Таблица 22.1 – Поддерживаемые режимы тактирования модулятора

Режим	Описание
Режим 0 (MOD=00b)	Частота тактового сигнала модулятора SDFMx_CLK совпадает со скоростью выставления данных. Данные SDFMx_DATA захватываются по каждому положительному фронту декодером.
Режим 1 (MOD=01b)	Частота тактового сигнала модулятора SDFMx_CLK в два раза меньше скорости выставления данных. Данные SDFMx_DATA захватываются декодером по каждому перепаду тактового сигнала.
Режим 2 (MOD=10b)	Тактовый сигнал модулятора SDFMx_CLK неактивен, для передачи данных используется манчестерское кодирование.
Режим 3 (MOD=11b)	Декодер является источником тактового сигнала SDFMx_CLK. По первому положительному фронту модулятор выставляет данные на SDFMx_DATA, по следующему происходит их захват декодером.

Частота для режима 3 генерируется из тактового сигнала АНВ с помощью делителя CORM3 регистра STLPARM. Делитель принимает значения от 0 до 255, и результирующий коэффициент деления определяется по формуле  $2 \times (CORM3 + 1)$ .

Зависимость формы сигналов на выводах SDFMx\_CLK и SDFMx\_DATA показана на рисунке 22.3.

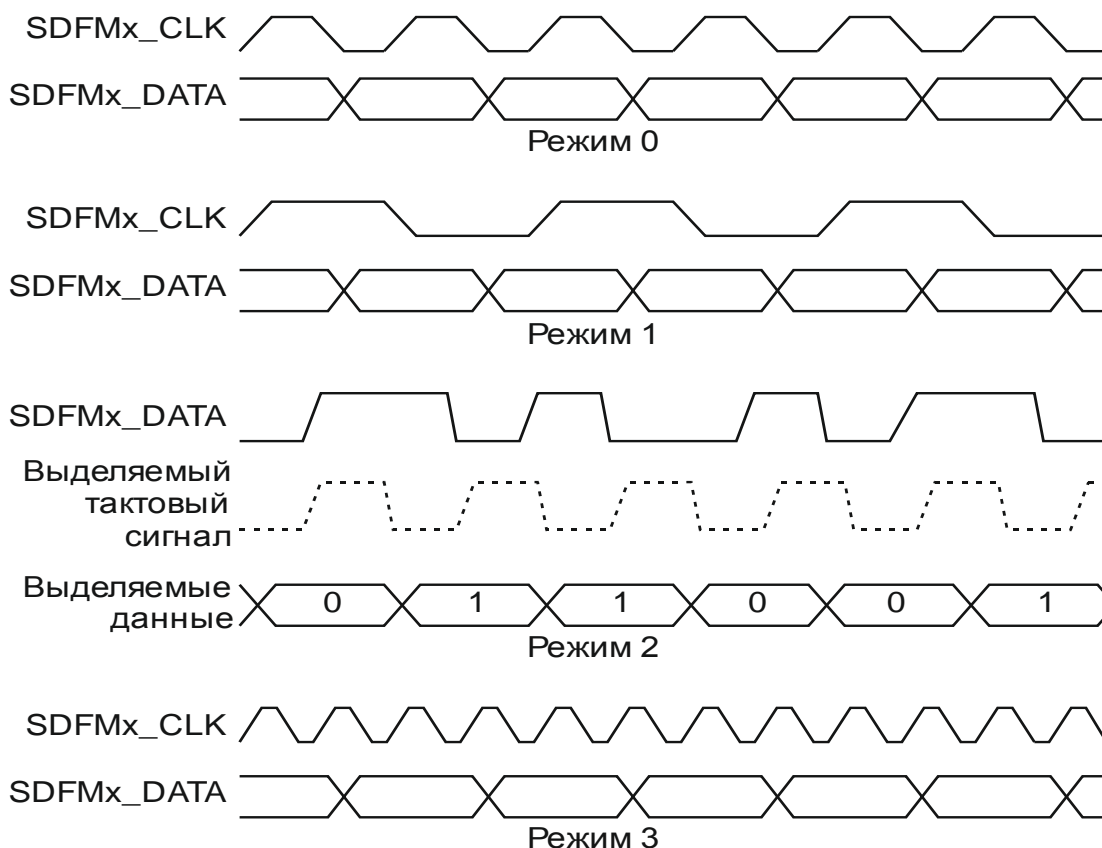


Рисунок 22.3 - Форма входных сигналов в зависимости от выбранного режима тактирования модулятора

### 22.3 Фильтр данных

Фильтр данных - это конфигурируемый Sinc-фильтр, который поддерживает режимы работы: Sinc1, Sinc2, Sinc3, Sinc4, Sinc5 and SincFast. По умолчанию, фильтр выключен и чтобы его активировать, сначала необходимо разрешить работу всех фильтров глобально - установить бит MFE в регистре MFILEN, а затем записать единицу в бит FEN

соответствующего регистра DFPARM. После установки FEN фильтр начнет выдавать данные, которые будут сохраняться в буфере DATA.

Выбор режима фильтрации осуществляется с помощью поля SST регистра DFPARM.

Коэффициент децимации, соответствующий полю DOSR регистра DFPARM, может принимать значения от 1 до 4096 (в поле пишется значение на единицу меньше).

### Формат представления и диапазоны результатов

Фильтр интерпретирует низкий уровень входных данных как «-1», а высокий как «1». Поэтому данные на выходе фильтра могут быть как положительными, так и отрицательными. Формат выходных данных – 32-битное знаковое целое в дополнительном коде. В таблице 22.2 показаны возможные диапазоны данных фильтра в зависимости от режима работы и некоторых коэффициентов децимации DOSR.

Таблица 22.2 — Диапазоны результатов фильтра данных в зависимости от конфигурации

DOSR	Sinc1	Sinc2	Sinc3	Sinc4	Sinc5	Sincfast
x	x	$x^2$	$x^3$	$x^4$	$x^5$	$2x^2$
3	[-4; 4]	[-16; 16]	[-64; 64]	[-256; 256]	[-1024; 1024]	[-32; 32]
7	[-8; 8]	[-64; 64]	[-512; 512]	[-4096; 4096]	[-32768; 32768]	[-128; 128]
15	[-16; 16]	[-256; 256]	[-4096; 4096]	[-65536; 65536]	[-1048576; 1048576]	[-512; 512]
31	[-32; 32]	[-1024; 1024]	[-32768; 32768]	[-1048576; 1048576]	[-33554432; 33554432]	[-2048; 2048]

Целое знаковое 32-битное слово может принимать значения от -2147483648 до 2147483647, что является ограничением для использования больших значений DOSR с Sinc-фильтрами высокого порядка. Максимально допустимое значение DOSR для каждого из типов фильтров показано в таблице 22.3.

Таблица 22.3 – Максимально допустимое DOSR для различных типов фильтров

Фильтр	DOSR	Диапазон данных
Sinc1	4095	[-4096; 4096]
Sinc2	4095	[-16777216; 16777216]
Sinc3	1289	[-2146689000; 2146689000]
Sinc4	214	[-2136750625; 2136750625]
Sinc5	72	[-2073071593; 2073071593]
Sincfast	4095	[-33554432; 33554432]

На рисунке 22.4 продемонстрирован пример АЧХ Sinc1, Sinc3 и Sinc5 фильтров при DOSR=31 и частоте модуляции 10 МГц.

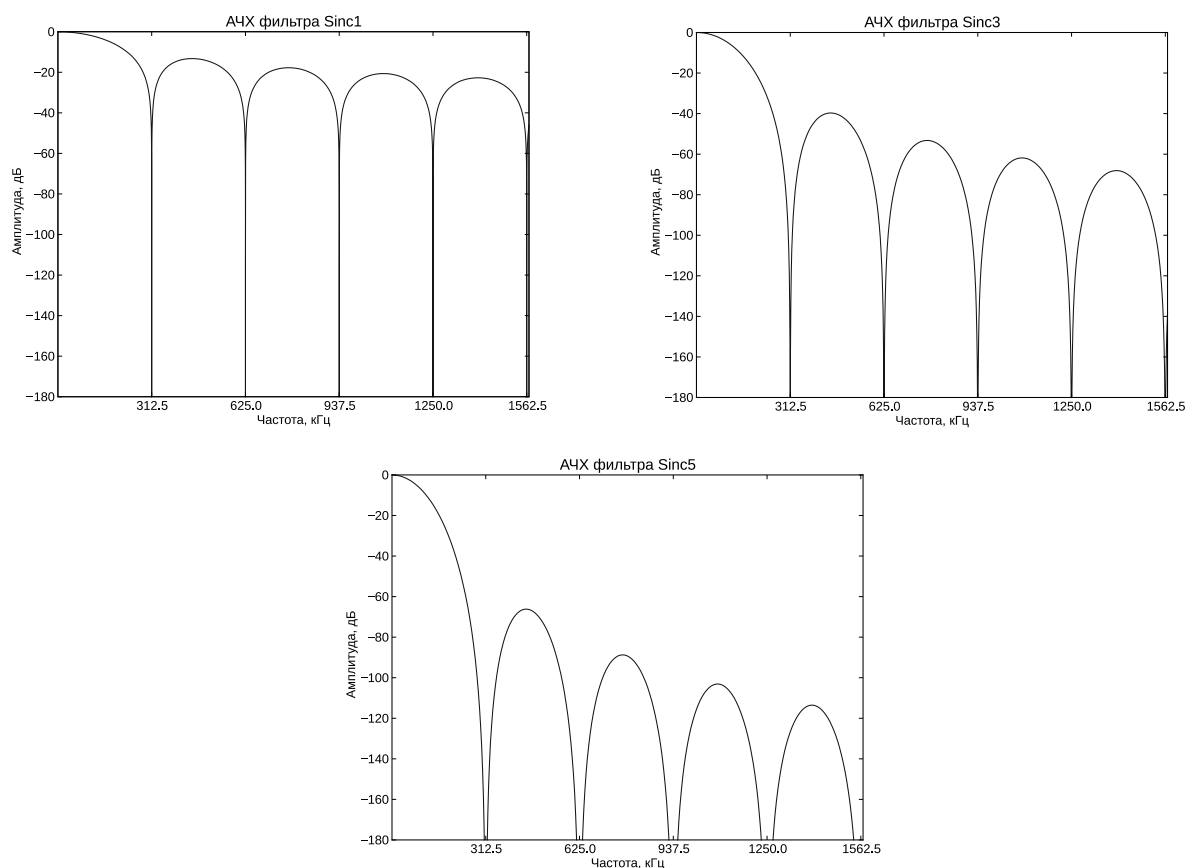


Рисунок 22.4 – АЧХ Sinc1, Sinc2, Sinc3 фильтров при DOSR=31 и частоте модуляции 10 МГц

В фильтре данных предусмотрен механизм аппаратного сдвига результатов, для отбрасывания необходимого количества младших бит. Величина сдвига вправо может быть равна от 1 до 23 разрядов и задается с помощью поля SH регистра DPARM. По умолчанию поле равно нулю - сдвиг отключен.

Стоит отметить, что из-за особенностей архитектуры цифровых Sinc-фильтров, первые несколько результатов, получаемых сразу после настройки и включения фильтра или события синхронизации, будут некорректны. Конкретное количество для каждого из типов фильтров показано в таблице 22.4.

Таблица 22.4 – Количество некорректных результатов, получаемых сразу после настройки и включения или синхронизации фильтра

Фильтр	Количество
Sinc1	Отсутствуют
Sinc2	1
Sinc3	2
Sinc4	3
Sinc5	4
Sincfast	2

### Буфер результатов FIFO

В состав каждого фильтра данных входит буфер результатов, организованный по принципу FIFO и состоящий из восьми 32-битных ячеек. Чтение FIFO осуществляется через регистр DATA.

По умолчанию, буфер отключен и, соответственно, в DATA хранится только один последний результат. Активировать FIFO можно путем установки бита EN в регистре

FFCTL. Количество результатов, попавших в FIFO, можно отследить с помощью поля STAT регистра FFCTL. Существует возможность генерации прерывания по уровню заполнения FIFO, заданному полем IL регистра FFCTL. Каждый раз, когда значение  $STAT \geq IL$  будет генерироваться прерывание.

Способы очистки буфера FIFO:

- сброс разрешения работы фильтра данных (бит MFE регистра MFILEN или бит FEN регистра DFPARM);
- автоматическая очистка по синхронизации SDSYNC, если установлен бит FFSCLEN регистра SYNC и включена синхронизация ( $WTSEN = 1$ ).

### **Внешняя синхронизация**

Каждый из фильтров данных может быть синхронизирован с блоком ШИМ или таймером посредством импульса на входе SDSYNC. Сигналы PWM\_SOCA, PWM\_SOCB и TMR\_ADCSOC подключаются к SDSYNC через мультиплексор, управляемый полем SRC регистра SYNC.

Событие синхронизации приводит к сбросу внутреннего счётчика децимации (его максимальным значением является DOSR регистра DFPARM). По умолчанию, синхронизация отключена, и может быть активирована установкой бита ENSYN регистра DFPARM.

Дополнительно, может быть включена функция ожидания синхронизации. Эта функция позволяет начать сохранять результаты фильтрации в FIFO только после прихода события синхронизации. По умолчанию функция выключена, её активация происходит с помощью установки бита WTSEN в регистре SYNC.

Когда функция ожидания синхронизации включена, то FIFO игнорирует все поступающие с фильтра результаты до тех пор, пока не случится событие синхронизации. По приходу события выставляется флаг WTSFLG в регистре SYNC, и данные начинают сохраняться в FIFO до тех пор, пока флаг установлен.

Флаг WTSFLG может очищаться двумя способами:

- автоматически, по установке соответствующего флага прерывания FDRx в регистре IFLG. Управляется автоматический сброс полем WTSCLEN регистра SYNC, активирован по умолчанию;
- вручную, путем установки бита WTSFLGCLR.

В режиме ожидания синхронизации пока флаг WTSFLG сброшен, данные в буфер не поступают.

## **22.4 Компаратор**

Основной задачей дополнительного фильтра с функцией сравнения (компаратора) является слежение за попаданием результатов фильтрации в заданный диапазон и обеспечение быстрой реакции ШИМ в случае выхода из него.

Компаратор - это конфигурируемый Sinc-фильтр, который поддерживает режимы работы: Sinc1, Sinc2, Sinc3, Sinc4, Sinc5 and SincFast. По умолчанию компаратор выключен и чтобы его активировать, сначала необходимо разрешить работу всех фильтров глобально – установить бит MFE в регистре MFILEN, а затем записать единицу в бит SEN соответствующего регистра SPARM. После установки SEN фильтр начнет обрабатывать входной битовый поток данных, проводить сравнение результата с установленными порогами, а последний результат сохранять в регистре CDATA.

Выбор режима фильтрации осуществляется с помощью поля CS регистра SPARM.

Коэффициент децимации, соответствующий полю COSR регистра SPARM, может принимать значения от 1 до 4096 (в поле пишется значение на единицу меньше).

### Формат представления и диапазоны результатов

Компаратора интерпретирует низкий уровень входных данных как «0», а высокий как «1». Поэтому данные на выходе фильтра – 32-битное беззнаковое целое.

В таблице 22.5 показаны возможные диапазоны данных фильтра в зависимости от режима работы и некоторых коэффициентов децимации COSR.

Таблица 22.5 – Диапазоны результатов компаратора в зависимости от конфигурации

DOSR	Sinc1	Sinc2	Sinc3	Sinc4	Sinc5	Sincfast
x	x	x <sup>2</sup>	x <sup>3</sup>	x <sup>4</sup>	x <sup>5</sup>	2x <sup>2</sup>
3	[0; 4]	[0; 16]	[0; 64]	[-256; 256]	[0; 1024]	[0; 32]
7	[0; 8]	[0; 64]	[0; 512]	[0; 4096]	[0; 32768]	[0; 128]
15	[0; 16]	[0; 256]	[0; 4096]	[0; 65536]	[0; 1048576]	[0; 512]
31	[0; 32]	[0; 1024]	[0; 32768]	[0; 1048576]	[0; 33554432]	[0; 2048]

Целое беззнаковое 32-битное слово может принимать значения от 0 до 4294967295, что является ограничением для использования больших значений COSR с Sinc-фильтрами высокого порядка. Максимально допустимое значение COSR для каждого из типов фильтров показано в таблице 22.6.

Таблица 22.6 – Максимально допустимое COSR для различных типов фильтров

Фильтр	DOSR	Диапазон данных
Sinc1	4095	[0; 4096]
Sinc2	4095	[0; 16777216]
Sinc3	1625	[0; 4291015625]
Sinc4	255	[0; 4294967295]
Sinc5	84	[0; 4182119424]
Sincfast	4095	[0; 33554432]

### Сравнение по условию «Результат ≥ СМРН»

Выходное значение фильтра может проходить сравнение с тремя пороговыми уровнями, которые соответствуют трём типам событий.

Данные больше либо равны верхнему порогу СМРН – это событие может быть использовано для контроля выхода за верхнюю границу диапазона. При положительном сравнении установится соответствующий флаг CFHx в регистре CFLG. Состояние флага CFHx при этом передается на триггеры событий блоков ШИМ.

Флаг может сбрасываться двумя способами:

- вручную, с помощью установки бита CFHx в регистре CFLGCLR;
- автоматически, если установлен бит HWHCLR регистра СPАРМ и произошло отрицательное сравнение.

### Сравнение по условию «результат ≥ СМРНЗ»

Данные больше либо равны порогу нуля СМРНЗ – это событие может быть использовано для контроля частоты превышения порога, например, порога нуля. При положительном сравнении установится соответствующий флаг CFHZx в регистре CFLG. Состояние флага CFHZx при этом передается на триггеры событий блоков ШИМ.

Флаг может сбрасываться двумя способами. Вручную, с помощью установки бита CFHZx в регистре CFLGCLR. И автоматически, если установлен бит HWHZCLR регистра СPАРМ и произошло отрицательное сравнение.



### Сравнение по условию «Результат ≤ CMPL»

Данные меньше либо равны нижнему порогу CMPL – это событие может быть использовано для контроля выхода за нижнюю границу диапазона. При положительном сравнении установится соответствующий флаг CFLx в регистре CFLG. Состояние флага CFLx при этом передается на триггеры событий блоков ШИМ.

Флаг может сбрасываться двумя способами. Вручную, с помощью установки бита CFLx в регистре CFLGCLR. И автоматически, если установлен бит HWHCLR регистра CPARM и произошло отрицательное сравнение.

### 22.5 Скорость работы и задержка Sinc-фильтра

В фильтре данных и компараторе применяются одинаковые типы Sinc-фильтров, скорость работы которых можно оценить с помощью выражения:

$$R_F = \frac{R_M}{OSR + 1},$$

где  $R_F$  – скорость выдачи результатов фильтром, сэмплов/с;

$R_M$  – скорость выдачи данных модулятором, Гц;

OSR – величина децимации (поле DOSR регистра DFPARM для фильтра данных или поле COSR регистра CPARM для компаратора).

Величина задержки сигнала, проходящего через фильтр, определяется по формуле:

$$L_F = \frac{O_F}{R_F},$$

где  $L_F$  – величина задержки, с;

$R_F$  – скорость выдачи результатов фильтром, сэмплов/с;

$O_F$  – порядок фильтра.

Соответствия между типом фильтра и его порядком приведены в таблице 22.7.

Таблица 22.7 – Соответствие между типом фильтра и его порядком

Фильтр	Порядок
Sinc1	1
Sinc2	2
Sinc3	3
Sinc4	4
Sinc5	5
Sincfast	3

Пример расчета представлен ниже.

Исходные данные:

- тип фильтра Sinc3 (порядок  $O_F = 3$ );
- скорость модулятора  $R_M = 10$  МГц;
- величина децимации  $OSR = 255$ .

Скорость выдачи результатов фильтром:

$$R_F = \frac{10000000}{255 + 1} = 39063 \text{ сэмплов/с.}$$

Задержка фильтра:

$$L_F = \frac{3}{39063} = 76,8 \text{ мкс.}$$

## 22.6 Генерация прерываний

Общая схема генерации прерываний показана на рисунке 22.5.

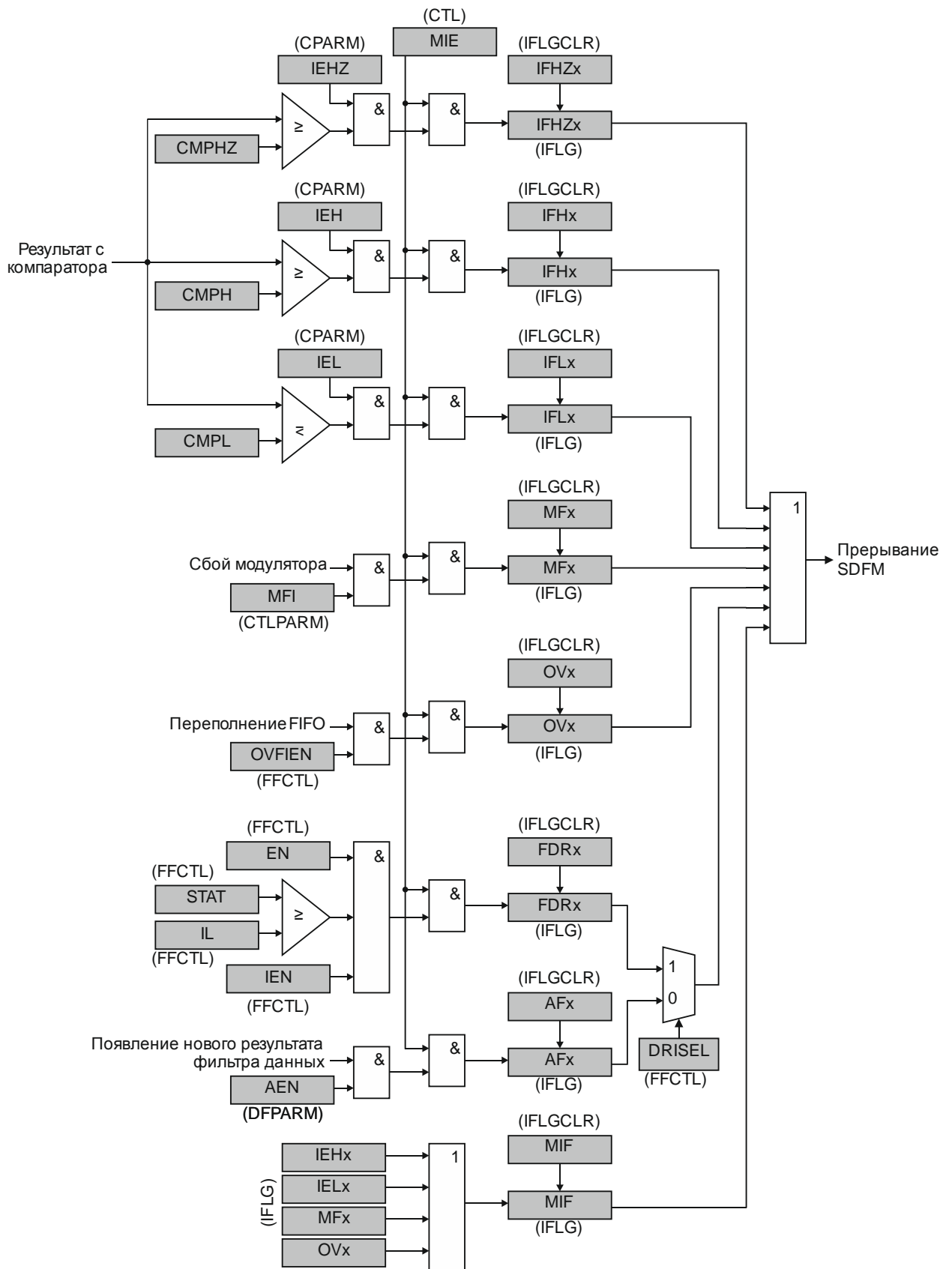


Рисунок 22.5 – Схема генерации прерываний демодулятора

### **Прерывание фильтра данных по уровню заполнения FIFO**

Буфер FIFO каждого фильтра данных может быть настроен на генерацию прерывания по определенному уровню заполнения (поле IL регистра FFCTL).

Процедура настройки генерации такого прерывания:

- включить FIFO, установив бит EN в регистре FFCTL;
- включить прерывание FIFO, установив бит IEN в регистре FFCTL;
- настроить уровень заполнения с помощью поля IL регистра FFCTL;
- выбрать в качестве источника прерывания фильтра уровень заполнения FIFO, установив бит DRISEL в регистре FFCTL;
- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

Теперь, когда значение поля  $STAT \geq IL$  будет вызвано прерывание SDFM, и установится соответствующий флаг прерывания FDRx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит FDRx регистра IFLGCLR.

### **Прерывание фильтра данных по переполнению FIFO**

Буфер FIFO каждого фильтра данных может быть настроен на генерацию прерывания по переполнению.

Процедура настройки генерации такого прерывания:

- включить FIFO, установив бит EN в регистре FFCTL;
- включить прерывание FIFO по переполнению, установив бит OVFIEN в регистре FFCTL;
- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг OVx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит OVx регистра IFLGCLR.

Дополнительно, при генерации этого прерывания также установится флаг MIF регистра IFLG, который может быть сброшен записью единицы в бит MIFx регистра IFLGCLR.

### **Прерывание фильтра данных по появлению новых результатов**

Каждый фильтр данных может быть настроен на генерацию прерывания по появлению новых результатов на выходе.

Процедура настройки генерации такого прерывания:

- включить прерывание по появлению новых результатов, установив бит AEN в регистре DFPARM;
- выбрать в качестве источника прерывания фильтра появление новых результатов, сбросив бит DRISEL в регистре FFCTL;
- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг AFx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит AFx регистра IFLGCLR.

### **Прерывание компаратора по условию «Результат $\geq$ СМРН»**

Компаратор может быть настроен на генерацию прерывания, если данные больше либо равны верхнему порогу СМРН.

Процедура настройки генерации такого прерывания (после настройки компаратора):

- установить бит IEN регистра SPARM;
- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг IFHx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит IFHx регистра IFLGCLR. Дополнительно, при генерации этого прерывания также установится флаг MIF регистра IFLG, который может быть сброшен записью единицы в бит MIFx регистра IFLGCLR.

#### **Прерывание компаратора по условию «Результат $\geq$ CMPHZ»**

Компаратор может быть настроен на генерацию прерывания, если данные больше либо равны порогу нуля CMPHZ.

Процедура настройки генерации такого прерывания (после настройки компаратора):

- установить бит IENZ регистра SPARM;

- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг IFHZx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит IFHx регистра IFLGCLR.

#### **Прерывание компаратора по условию «Результат $\leq$ CMPL»**

Компаратор может быть настроен на генерацию прерывания, если данные меньше либо равны нижнему порогу CMPL.

Процедура настройки генерации такого прерывания (после настройки компаратора):

- установить бит IEL регистра SPARM;

- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг IFLx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит IFLx регистра IFLGCLR. Дополнительно, при генерации этого прерывания также установится флаг MIF регистра IFLG, который может быть сброшен записью единицы в бит MIFx регистра IFLGCLR.

#### **Прерывание по сбою модулятора**

Если тактовый сигнал на выводе SDFMx\_CLK отсутствует в течение 64 тактов тактовой частоты АНВ, то такое событие считается сбоем, который может использоваться для вызова прерывания.

Процедура настройки генерации такого прерывания:

- установить бит MFIE в регистре CTLPARM;

- разрешить глобальную генерацию прерываний блока установкой бита MIE регистра CTL.

При генерации прерывания установится соответствующий флаг MFx в регистре IFLG. Сброс флага можно осуществить через запись единицы в бит MFx регистра IFLGCLR. Дополнительно, при генерации этого прерывания также установится флаг MIF регистра IFLG, который может быть сброшен записью единицы в бит MIFx регистра IFLGCLR.

## **22.7 Генерация запросов DMA**

Для разрешения использования DMA фильтром данных, необходимо установить бит EN в регистре DMACTL. Поле LVL того же регистра задает уровень заполнения буфера FIFO фильтра, по достижении которого будет запущен DMA. Перенос данных будет выполняться, пока не будет передано число результатов, соответствующее состоянию поля LVL.

## 23 Приемопередатчик UART

В состав микроконтроллера два идентичных упрощенных асинхронных приемопередатчика UART0, UART1.

В состав приемопередатчика входят два буфера, два сдвиговых регистра и настраиваемый делитель для обеспечения необходимой скорости передачи. Структурная схема приемопередатчика показана на рисунке 23.1.

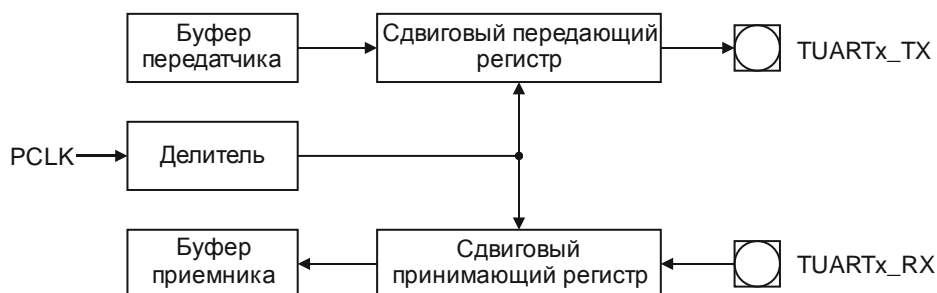


Рисунок 23.1 – Структурная схема приемопередатчика

### 23.1 Функционирование приемопередатчика

Скорость передачи настраивается с помощью делителя, управляемого регистром BAUDDIV. Минимально допустимое значение деления равняется 16.

Приемопередатчик функционирует в следующем режиме:

- передача и прием слов данных шириной 8 бит;
- отсутствие бита четности;
- один стоповый бит.

Включается передатчик установкой бита TXEN регистра CTRL. Запись в регистр DATA помещает данные в буфер передатчика. Разделение передающей части на буфер и сдвигающий регистр позволяет записывать новые данные во время передачи текущих. При заполненном буфере выставляется флаг TXF регистра STATE. При попытке записать новые данные в заполненный буфер установится флаг TO регистра STATE. Сбросить флаг переполнения можно записью единицы.

Включается приемник установкой бита RXEN регистра CTRL. Чтение регистра DATA возвращает данные из буфера приемника. Разделение приемной части на буфер и сдвигающий регистр позволяет читать уже принятые данные во время приёма текущих. При заполненном буфере выставляется флаг RXF регистра STATE. При попытке получить новые данные в заполненный буфер установится флаг RO регистра STATE. Сбросить флаг переполнения можно записью единицы.

### 23.2 Прерывания

Существует возможность формирование прерывания по любому из четырёх источников:

- по приему;
- по передаче;
- по переполнению приемного буфера;
- по переполнению передающего буфера.

Включается прерывание с помощью установки соответствующего бита регистра CTRL. При возникновении прерывания устанавливается соответствующий источнику флаг в регистре INTSTAT. Сбрасываются флаги прерываний записью единицы.

## 24 Приемопередатчик UART

В состав микроконтроллера входят четыре идентичных универсальных асинхронных приемопередатчика UART0-UART3.

В состав приемопередатчика входят два буфера типа FIFO. Буфер приемника имеет разрядность 12, буфер передатчика – разрядность восемь. Каждый буфер может хранить до 32 байт данных, и каждый буфер может быть сконфигурирован (программно) как 32-байтный или как однобайтный.

В состав приемопередатчика также включен кодек последовательного интерфейса инфракрасной (ИК) передачи данных в соответствии с протоколом IrDASIR.

Приемопередатчик обеспечивает:

- независимое маскирование прерываний от буфера передатчика, буфера приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки;

- возможность деления тактовой частоты в диапазоне от 1 до 65 535 (допускается использование нецелых коэффициентов деления, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц);

- поддержку прямого доступа к памяти;

- поддержку функции управления модемом (сигналы CTS, DCD, DSR, RTS, DTR и RI).

Приемопередатчик реализует:

- передачу данных длиной от 5 до 8 бит со скоростью до 921 600 бит/с;

- контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение либо не передается);

- формирование одного или двух стоповых бит;

- обнаружение ложных стартовых битов;

- формирование и обнаружение сигнала разрыва линии.

Кодек ИК передачи данных IrDASIR обеспечивает:

- программный выбор обмена данными по линиям асинхронного приемопередатчика либо кода ИК связи IrDASIR;

- поддержку функционирования с информационной скоростью до 115 200 бит/с;

- поддержку длительности бит для нормального режима и 3/16 для режима пониженного энергопотребления;

- программируемое деление опорной частоты для получения заданной длительности бит в режиме пониженного энергопотребления.

### Функциональные возможности

Режим работы приемопередатчика и скорость обмена данными контролируются регистром LCRH и регистрами делителя IBRD и FBRD.

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 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 приемника. В случае переполнения буфера приемника также устанавливается соответствующий бит, а буфер становится недоступным для записи.

## 24.1 Функционирование блока UART

На рисунке 24.1 показана упрощенная функциональная схема приемопередатчика.

Генератор тактового сигнала приемопередатчика формирует синхросигнал последовательного обмена данными, который представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK, и частотой в 16 раз превышающей частоту передачи данных.

Буфер передатчика предназначен для хранения данных, полученных от ЦП, до тех пор, пока они не будут переданы внешнему устройству.

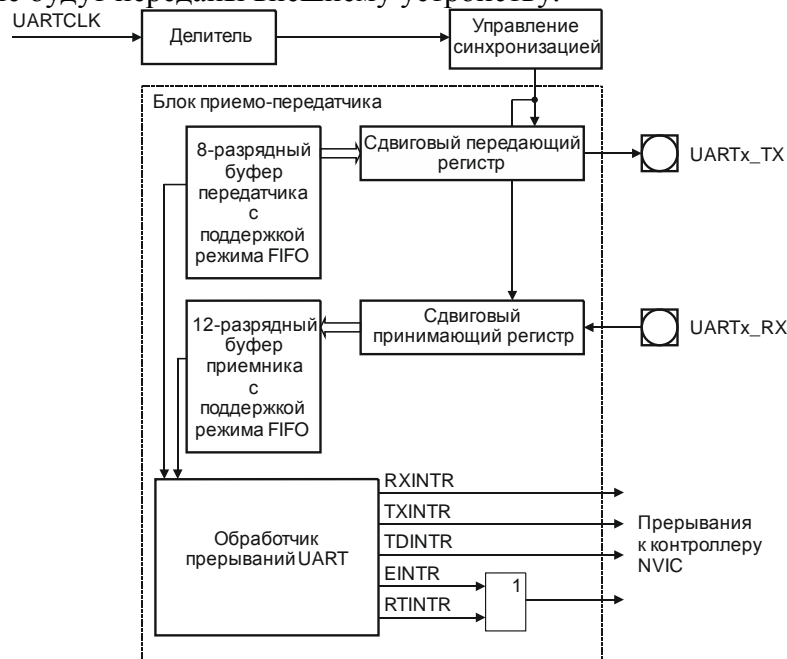


Рисунок 24.1 – Функциональная схема приемопередатчика

Буфер приемника предназначен для хранения данных и кодов ошибки (принятых от внешнего устройства) до тех пор, пока они не будут прочитаны ЦП.

Обработчик прерываний генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения независимых прерываний по схеме ИЛИ. Сигнал прерывания передается на контроллер NVIC.

### Сброс модуля

Приемопередающая логика модуля, управляющие регистры и FIFO по умолчанию находятся в сбросе. Снять сброс можно путём установки бита RSTDIS в соответствующем регистре UARTCFGx (i – номер модуля 0 или 1) блока RCU.

### Синхронизация

Существует ограничение на соотношение между частотами тактовых сигналов PCLK и UARTCLK:

$$f_{\text{UARTCLK}} \leq 5/3 \times f_{\text{PCLK}} \quad (24.1)$$

Например, для достижения максимальной скорости передачи данных 921 600 бод (при  $f_{\text{UARTCLK}} = 921\,600 \times 16 = 14,7456$  МГц) частота  $f_{\text{PCLK}}$  должна быть не менее 8,84736 МГц.

Для точной настройки частоты передачи данных используются два делителя – один управляется регистром UARTCFGx блока RCU, второй находится внутри модуля UART.

Коэффициент деления второго делителя имеет целую и дробную части, которые задаются регистрами IBRD и FBRD. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными на стандартных скоростях, используя в качестве источника тактовый сигнал с произвольной частотой более 3,6864 МГц.

Коэффициент деления  $K$  частоты сигнала UARTCLK рассчитывается по формуле

$$K = f_{\text{UARTCLK}} / (16 \times \text{baudrate}), \quad (24.2)$$

где  $f_{\text{UARTCLK}}$  – частота сигнала синхронизации блока UART – UARTCLK, Гц;

baudrate – скорость передачи, бод.

Получившееся дробное десятичное число следует разделить на две части – целую и дробную.

Целая часть после преобразования в двоичный формат записывается в регистр IBRD.

Дробная часть умножается на 64 и округляется до ближайшего целого числа. Полученное число преобразовывается в двоичный формат и записывается в регистр FBRD.

Для примера, пусть требуемая скорость передачи данных 230 400 бит/с и частота тактового сигнала UARTCLK равна 4 МГц. Тогда:

$$K = (4 \times 10^6) / (16 \times 230\,400) = 1,085.$$

Получившееся число разбивается на две части – 1 и 0,085.

В регистр IBRD записывается значение 0001h.

Значение  $(0,085 \times 64)$  округляется и преобразовывается в 05h для записи в регистр FBRD.

Таким образом, реальные значения коэффициента деления частоты и скорости передачи будут следующими:

$$K = 1 + 5/64 = 1,078,$$

$$\text{baudrate} = (4 \times 10^6) / (16 \times 1,078) = 231\,911 \text{ бит/с.}$$

Ошибка установки скорости:

$$\Delta = ((231\,911 - 230\,400) / 230\,400) \times 100 \% = 0,656 \%.$$

Максимальная ошибка установки скорости передачи данных:

$$\Delta = (1/64) \times 100 \% = 1,56 \%.$$

Такая ошибка возникает в случае  $K = 1$ , при этом разница накапливается в течение 64 тактовых интервалов.

Содержимое регистров LCRH, IBRD и FBRD обновляется при записи в регистр LCRH. Таким образом, для того, чтобы новые параметры коэффициента деления вступили в силу, после их записи в регистры IBRD и FBRD, необходимо осуществить запись в регистр LCRH и только в такой последовательности.

Примечание – Изменение содержимого регистров IBRD, FBRD и LCRH допускается только во время, когда приемопередатчик запрещен и не осуществляется передача/прием байта.

### **Передача и прием данных**

Данные для передачи заносятся в буфер передатчика посредством записи в регистр DR. После записи хотя бы одного байта в буфер передатчика устанавливается флаг BUSY



в регистре FR. Это состояние флага сохраняется, пока буфер передатчика не пуст (даже если работа приемопередатчика запрещена). Далее, если работа приемопередатчика разрешена (установлены биты UARTEN и TXE регистра CR), начинается передача информационного кадра с параметрами, указанными в регистре управления линией LCRH. Передача данных продолжается до опустошения буфера передатчика (до окончания передачи всех байт). По окончании передачи сбрасывается флаг BUSY.

При приеме байта данных (установлены биты UARTEN и RXE регистра CR) для каждого бита производятся три выборки уровня, и решение о значении бита принимается по мажоритарному принципу.

В случае если приемник находился в неактивном состоянии (постоянный высокий уровень сигнала на линии UARTx\_RX), и произошла смена уровня входного сигнала с высокого на низкий (стартовый бит), включается счетчик, тактируемый внутренним сигналом, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов.

Стартовый бит считается достоверным в случае, если сигнал на линии UARTx\_RX сохраняет низкий логический уровень в течение восьми периодов внутреннего синхросигнала с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

После обнаружения достоверного стартового бита очередной бит данных фиксируется каждые 16 отсчетов тактового сигнала. Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

По окончании приема байта производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UARTx\_RX). После этого байт заносится в буфер приемника вместе с тремя битами признаков ошибки, см. рисунок 24.2, и битом переполнения буфера.

В 12-разрядной ячейке буфера байт данных располагается в области младших восьми бит, три бита признаков ошибки – в битах с 8 по 10.

Флаг переполнения буфера приемника выставляется в том случае, если к моменту, когда очередной кадр данных полностью принят, буфер уже заполнен. В этом случае принятый кадр остается в сдвиговом принимающем регистре и, в случае приема следующего кадра данных, будет потерян.

Как только в буфере приемника освобождается место для записи, кадр данных, находящийся в сдвиговом регистре, переписывается в буфер, а флаг переполнения сбрасывается.

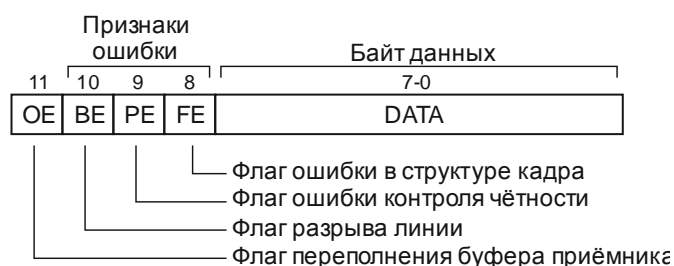


Рисунок 24.2 – 12-разрядная ячейка принимающего буфера

Данные из буфера приемника можно прочитать посредством регистра DR. Состояние признаков ошибки и флага переполнения определяется чтением регистра RSR и относится к последнему байту, считанному из регистра DR, в связи с этим регистр DR всегда должен считываться первым.

Все флаги сбрасываются одновременно записью любого значения в регистр RSR или после сброса устройства.

## Примечания

1 Необходимо запрещать работу приемопередатчика перед перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема, то перед остановкой он завершает выполняемую операцию.

2 Целостность данных в буферах передатчика и приемника не гарантируется, если установился флаг BRK (разрыв линии), или если программное обеспечение произвело остановку приемопередатчика после его повторного перевода в разрешенное состояние.

## 24.2 Функционирование ИК передатчика

В случае активизации ИК передатчика (бит SIREN регистра CR), передача информации осуществляется не с помощью вывода UARTx\_TX и, а посредством UARTx\_SIR.

В этом случае устройство переводит линию UARTx\_TX в пассивное состояние (высокий уровень) и перестает реагировать на изменение состояния модема, а также сигнала на линии UARTx\_RX.

Кодер преобразует поток данных с выхода асинхронного передатчика, сформированный по закону модуляции без возврата к нулю (NRZ). Спецификация физического уровня протокола IrDASIR подразумевает использование модуляции с возвратом к нулю и инверсией (RZI), в соответствии с которой передача логического нуля соответствует излучению одного светового ИК импульса. Сформированный выходной поток импульсов подается на усилитель и, далее, на ИК светодиод.

Длительность импульса в режиме IrDA составляет три периода внутреннего тактового генератора с частотой Baud16, то есть, 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением (активируется установкой бита SIRLP регистра CR) ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115 200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1,8432 МГц, в свою очередь, формируемого путем деления тактовой частоты модуля UART. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр ILPR.

Выход кодера имеет активное низкое состояние. При передаче логической единицы выход кодера остается в низком состоянии, при передаче логического нуля – формируется импульс, при этом выход кратковременно переводится в высокое состояние.

Как в нормальном режиме, так и в режиме пониженного энергопотребления использование нецелых значений коэффициента деления скорости передачи данных увеличивает джиттер фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами будут нерегулярными – период сигнала в разное время будет содержать различное количество периодов тактового сигнала модуля UART. Можно показать, что в наихудшем случае величина джиттера в потоке ИК импульсов может достигать трех периодов синхросигнала UARTCLK.

### 24.3 Режим модема

Приемопередатчик может использоваться как оконечное устройство или как оборудование передачи данных. Сигналы модема в режиме оконечного устройства и их назначение представлено в таблице 24.1.

Таблица 24.1 – Назначение сигналов в режиме модема

Сигнал	Назначение в зависимости от режима работы		Режим работы вывода
	Оконечное устройство	Оборудование передачи данных	
UARTx_RTS	Готов к передаче данных	Запрос передачи данных	Выход
UARTx_CTS	Запрос передачи данных	Готов к передаче данных	Вход
UARTx_DTR	Приемник данных готов	Источник данных готов	Выход
UARTx_DSR	Источник данных готов	Приемник данных готов	Вход
UARTx_DCD	Обнаружен информационный сигнал	–	Вход
UARTx_RI	Индикатор вызова	–	Вход

#### Аппаратное управление потоком данных

Программно активируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью выводов UARTx\_RTS и UARTx\_CTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена рисунке 15.3.

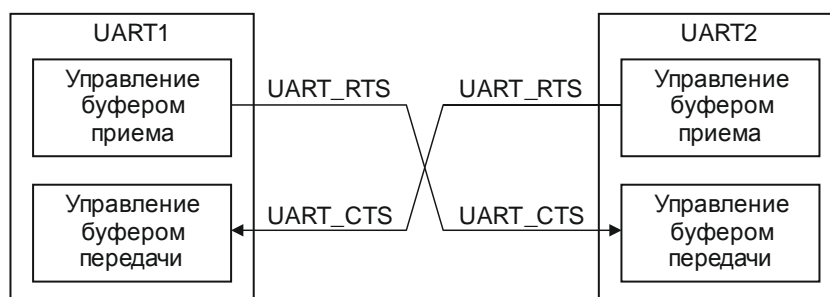


Рисунок 24.3 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, вывод UARTx\_RTS переводится в активное состояние только после того, как в буфере приемника появляется заданное количество свободных ячеек.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода вывода UARTx\_CTS в активное состояние.

Режим аппаратного управления потоком данных задается путем задания битов RTSEN и CTSEN в регистре управления CR.

Примечание – В случае если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEN для проверки состояния линии RTS.

Логика управления потоком данных по RTS использует данные о превышении уровня заполнения буфера приемника. Сигнал на выводе UARTx\_RTS переводится в активное состояние только после того, как в буфере приемника появляется заданное

количество свободных ячеек. После достижения порогового уровня заполнения буфера приемника сигнал RTS снимается (переводится в пассивное состояние), указывая, таким образом, на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего кадра.

Обратно в активное состояние сигнал RTS переводится после считывания данных из буфера приемника в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае если управление потоком данных по RTS запрещено, но при этом работа приемопередатчика разрешена, прием будет осуществляться до полного заполнения буфера приемника либо до завершения передачи данных.

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния вывода UARTx\_CTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе вывода UARTx\_CTS в неактивное состояние модуль завершает выдачу текущего передаваемого кадра, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено, и при этом работа приемопередатчика UART разрешена, данные будут выдаваться до опустошения буфера передатчика.

#### **24.4 Интерфейс прямого доступа к памяти**

Приемопередатчик оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMACR. Интерфейс DMA включает в себя шесть сигналов.

RXDMASREQ (для приема) – запрос передачи отдельного символа, инициируемый приемопередатчиком. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер приемника содержит, по меньшей мере, один символ.

RXDMABREQ (для приема) – запрос блочного обмена данными, инициируемый приемопередатчиком. Сигнал переходит в активное состояние в случае, если заполнение буфера приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера посредством полей регистра IFLS.

RXDMACLR (для приема) – сброс запроса на DMA, инициируемый приемопередатчиком. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

TXDMASREQ (для передачи) – запрос передачи отдельного символа, инициируемый приемопередатчиком. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер передатчика содержит, по меньшей мере, одну свободную ячейку.

TXDMABREQ (для передачи) – запрос блочного обмена данными, инициируемый приемопередатчиком. Сигнал переводится в активное состояние в случае, если заполнение буфера передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера посредством полей регистра IFLS.

TXDMACLR (для передачи) – сброс запроса на DMA, инициируемый 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 взаимно исключающими, 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 установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов, поскольку для них блок 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 DMATXDMAE или RXDMAE в регистре управления DMACR.

В случае запрета буферов устройство способно передавать и принимать только одиночные символы, и, как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления RXDMASREQ и TXDMASREQ.

Когда буферы 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.

В таблице 24.1 указаны значения параметров срабатывания запросов блочного обмена RXDMABREQ и TXDMABREQ в зависимости от порога заполнения буфера.

Таблица 24.1 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Количество незаполненных ячеек буфера передатчика	Количество заполненных ячеек буфера приемника
1/8	28	4
1/4	24	8
1/2	16	16
3/4	8	24
7/8	4	28

В регистре управления DMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки EINTR. При этом соответствующие линии запроса DMA – RXDMASREQ и RXDMABREQ переводятся в неактивное состояние (маскируются) до сброса EINTR. На линии запроса DMA, обслуживающие передатчик, состояние EINTR не влияет.

## 24.5 Прерывания

В модуле предусмотрено семь маскируемых источников прерываний.

Сигналы запросов на прерывания:

- RXINTR – от приемного FIFO;
- TXINTR – от передающего FIFO;
- RTINTR – по таймауту приемника;
- TDINTR – по окончанию передачи в линии;

- MSINTR – по состоянию модема;
- EINTR – по ошибке.
- UARTINTR – логическое ИЛИ сигналов запросов на прерывания.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IMSC.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания). Сброс прерывания осуществляется программно путём установки соответствующего бита в регистре ICR.

### **Сигнал RXINTR**

Запрос возникает в случае обнаружения одного из событий:

- буфер приемника в режиме FIFO и его заполнение достигло заданного порогового значения;
- буфер приемника имеет одну ячейку (режим FIFO запрещен) и принят один кадр данных.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока из буфера не будет прочитан, как минимум, один байт или выполнен программный сброс прерывания (регистр ICR).

### **Сигнал TXINTR**

Запрос возникает в случае обнаружения одного из событий:

- буфер передатчика в режиме FIFO и его опустошение достигло заданного порогового значения;
- буфер передатчика имеет одну ячейку (режим FIFO запрещен) и пуст.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока в буфер не будет записан, как минимум, один байт или выполнен программный сброс прерывания.

Запись данных в буфер передатчика допускается как перед разрешением работы приемопередатчика и прерываний, так и после разрешения.

Примечание – Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если работа приемопередатчика и прерывания от него разрешена до осуществления записи данных в буфер передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера.

### **Сигнал TDINTR**

Запрос возникает в случае окончания передачи в линии.

### **Сигнал MSINTR**

Прерывание возникает в случае изменения любой из линий состояний модема (UARTx\_CTS, UARTx\_DCD, UARTx\_DSR, UARTx\_RI). Прерывание сбрасывается программно.

### **Сигнал RTINTR**

Запрос возникает в случае, если буфер приемника не пуст и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание сбрасывается после считывания данных из буфера приемника до его опустошения или программно.

### **Сигнал EINTR**

- Запрос возникает в случае ошибки при приеме данных. Оно может быть вызвано:
- ошибкой в структуре кадра;

- ошибкой контроля четности;
- разрывом линии;
- переполнением буфера приемника.

### **Сигнал UARTINTR**

Логическое ИЛИ вышеперечисленных сигналов.

Таким образом, в контроллер NVIC поступает 7 линий прерываний от каждого блока UART.

## **24.6 Программирование**

Для программирования рекомендуется следующая последовательность действий:

- запретить работу приемопередатчика;
- дождаться окончания приема и/или передачи текущего байта данных;
- сбросить буфер передатчика посредством сброса бита FEN регистра LCRH;
- изменить настройки регистра CR;
- разрешить работу приемопередатчика.

## 25 Контроллер интерфейса CAN 2.0b

### 25.1 Протокол CAN

Последовательный интерфейс CAN (Controller Area Network) – интерфейс связи, эффективно поддерживающий распределенное управление в масштабе реального времени с высокой помехозащищенностью. Протокол связи определен в спецификации CAN 2.0B.

Протокол CAN оптимизирован для систем, в которых должно передаваться относительно небольшое количество информации (по сравнению с Ethernet или USB) к любому или всем узлам сети. Множественный доступ с опросом состояния шины позволяет каждому узлу получить доступ к шине с учетом приоритетов. Неадресная структура сообщений позволяет организовать многоабонентскую доставку данных с сокращением трафика шины. Быстрая устойчивая передача информации с системой контроля ошибок позволяет отключать неисправные узлы от шины, что гарантирует доставку критических по времени сообщений.

Область применения протокола CAN: от высокоскоростных сетей связи до электропроводов в автомобиле. Высокая скорость передачи данных (до 1 Мбит/с), хорошая помехозащищенность протокола, защита от неисправности узлов – делают шину CAN подходящей для промышленных приложений управления типа Device Net.

CAN имеет асинхронную последовательную структуру шины с одним логическим сегментом сети. CAN сеть может состоять из двух или более узлов с возможностью подключения/отключения узлов от шины без перенастройки других устройств, см. рисунок 18.1.

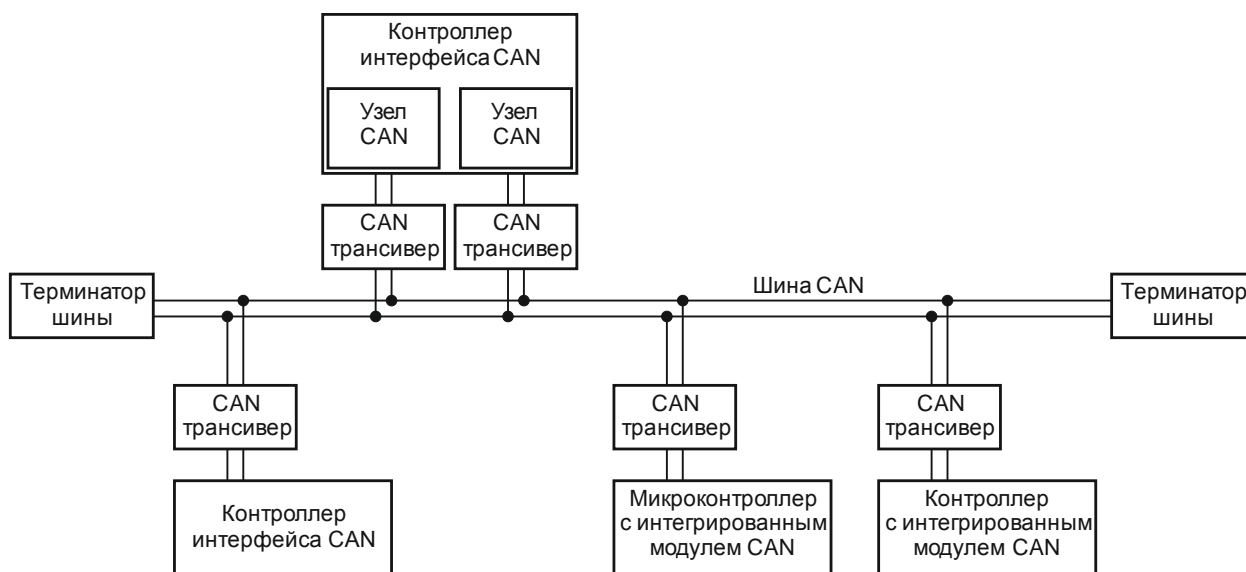


Рисунок 25.1 – Общая структура CAN сети

Логика шины работает по механизму монтажного И, в котором рецессивный бит соответствует логической единице, а доминантный – логическому нулю. Пока ни один узел не формирует доминантный бит, шина находится в рецессивном состоянии. Появление на шине доминантного бита (выставленного одним или несколькими узлами) создает доминантное состояние шины. Отсюда следует, что при выборе среды передачи данных необходимо точно определить, какое состояние будет доминантным, а какое – рецессивным. Одним из наиболее распространенных и дешевых вариантов линии связи является витая пара. Линии шины тогда называются CANH и CANL и могут быть подключены непосредственно к устройствам. Не существует никакого дополнительного стандарта на среду передачи данных.



При использовании в качестве линии связи витой пары с нагрузочными резисторами на концах можно получить максимальную скорость передачи данных 1 Мбит/с при длине линии до 40 м. Для линий связи протяженностью более 40 м необходимо снизить скорость передачи данных (для линии 1 000 м скорость шины должна быть не более 40 Кбит/с). Из-за дифференциального характера линии связи шина CAN малочувствительна к электромагнитным помехам. Экранирование шины значительно снизит воздействие внешнего электромагнитного поля, что особенно важно для высокоскоростных режимов работы.

Двоичная информация кодируется. Доминантным является низкий уровень, рецессивным – высокий уровень. Для гарантированной синхронизации данных всеми узлами шины используется принцип «бит-стаффинга». Это означает, что при последовательной передаче пяти бит одинаковой полярности передатчик вставляет один дополнительный бит противоположной полярности перед передачей остальных битов. Приемник также проверяет полярность и удаляет дополнительные биты.

В CAN протоколе при передаче данных приемные узлы не адресуются, а указывается идентификатор передатчика. С помощью идентификатора указывается содержание сообщения (например, применительно автомобиля – обороты, температура двигателя и т. д.) и степень приоритета сообщения. Более высокий приоритет у идентификатора, имеющего меньшее бинарное значение.

При коллективном доступе к шине используется неразрушающий арбитраж с опросом состояния шины. Перед началом передачи данных узел проверяет состояние шины (отсутствие активности на шине). При начале передачи сообщения узел становится управляющим шины, все остальные узлы переходят в режим приема. После приема сообщения (подтвержденного каждым узлом) каждый узел проверяет идентификатор в сообщении и сохраняет сообщение, если это требуется. В противном случае, сообщение сбрасывается. Если два или более узлов начинают передачу данных одновременно, поразрядный арбитраж позволяет избежать конфликта на шине. Каждый узел выдает на шину свой идентификатор (старший бит формируется первым) и контролирует ее состояние. Если узел посылает «1», а считывает «0», значит арбитраж потерян, и узел переключается в режим приема. Это происходит тогда, когда идентификатор конкурирующего узла имеет меньшее бинарное значение. Таким образом, узел с высоким приоритетом выигрывает арбитраж без необходимости повторять сообщение. Все остальные узлы будут пытаться передать сообщение после освобождения шины. Данный механизм не позволяет передавать сообщения одновременно разными узлами. Для этого программно должно быть обеспечено, чтобы узлы, передающие данные, не имели одинаковых идентификаторов. Оригинальная спецификация в версии CAN 2.0b (так называемая расширенная версия CAN) определяет возможность идентификатора иметь длину 11 или 29 бит.

Протокол CAN предусматривает следующие типы сообщений:

- сообщение данных (стандартное и расширенное);
- удаленный запрос данных;
- сообщение об ошибке;
- сообщение о перезагрузке.

#### **Стандартное сообщение данных**

Формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 25.2.

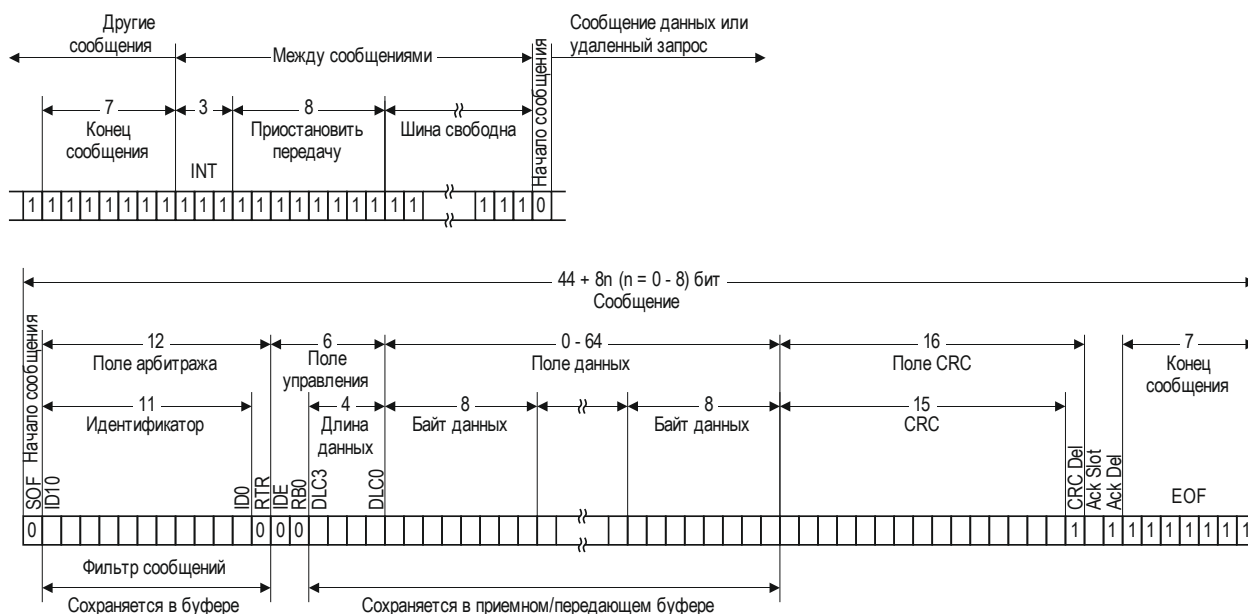


Рисунок 25.2 – Стандартное сообщение данных

Стандартное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;

- поле арбитража (12 бит), включающее поле ID идентификатора (11 бит), и бит RTR передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);

- поле управления (6 бит), включающее бит IDE, – указатель расширенного идентификатора (IDE = «0» соответствует стандартному идентификатору, IDE = «1» соответствует расширенному идентификатору), бит RBO – резервный доминантный бит и поле DLC – числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (допустимые значения от 0 до 8, другие значения использоваться не могут);

- поле данных (от 0 до 64 бит), содержащее целое число байт данных;

- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит), используемое для обнаружения возможных ошибок передачи данных, и бит CRC Del рецессивный разделитель CRC;

- поле подтверждения (2 бита), включающее бит ACK Slot подтверждения передачи (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом), и бит ACK Del рецессивный разделитель подтверждения;

- поле EOF конца сообщения (7 бит).

Между передачами двух любых сообщений шина должна оставаться в рецессивном состоянии как минимум в течение времени появления 3 бит (поле INT простоя). Если после появления трех рецессивных бит (поле INT) ни один узел не начал передачу, шина переходит в состояние бездействия IDLE и находится в рецессивном состоянии до появления доминантного бита сообщения.

### Расширенное сообщение данных

Расширенное сообщение данных формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 25.3.

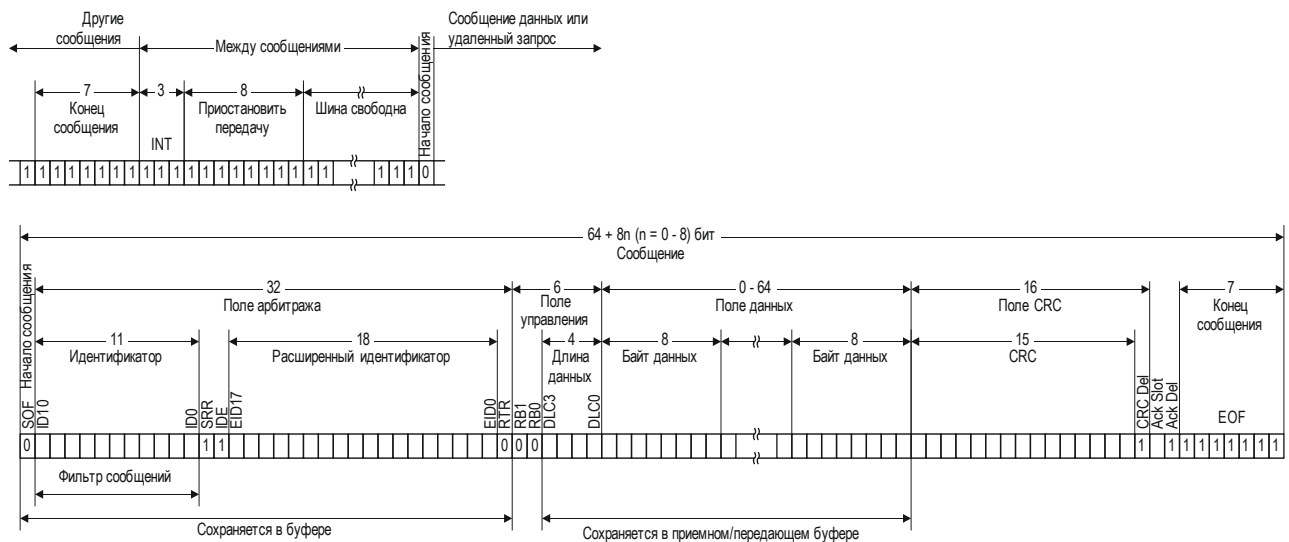


Рисунок 25.3 – Расширенное сообщение данных

Расширенное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;

- поле арбитража (38 бит), включающее поле стандартного идентификатора (11 бит), бит SRR заместитель удаленного запроса, бит IDE указатель расширенного идентификатора (рецессивный, что соответствует расширенному идентификатору) и поле расширенного идентификатора (18 бит);

- бит RTR передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);

- поле управления (6 бит), включающее бит RB0 – резервный доминантный бит, бит RB1 резервный доминантный бит и поле DLC числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (допустимые значения – от 0 до 8, другие значения использоваться не могут);

- поле данных (от 0 до 64 бит), содержащее целое число байт данных;

- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит), используемое для обнаружения возможных ошибок передачи данных и бит CRC Del рецессивный разделитель CRC;

- поле подтверждения (2 бита), включающее бит ACK Slot подтверждения передачи (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом) и бит ACK Del рецессивный разделитель подтверждения;

- поле EOF конца сообщения (7 бит).

## Удаленный запрос данных

Формируется, когда узлу требуются данные другого узла. Узел назначения посылает удаленный запрос с идентификатором источника. Соответствующий узел источника (распознавший свой идентификатор) посылает стандартное или расширенное сообщение в ответ на запрос.

Удаленный запрос данных существует в стандартном и расширенном вариантах. На рисунке 25.4 представлен вариант удаленного запроса со стандартным идентификатором.

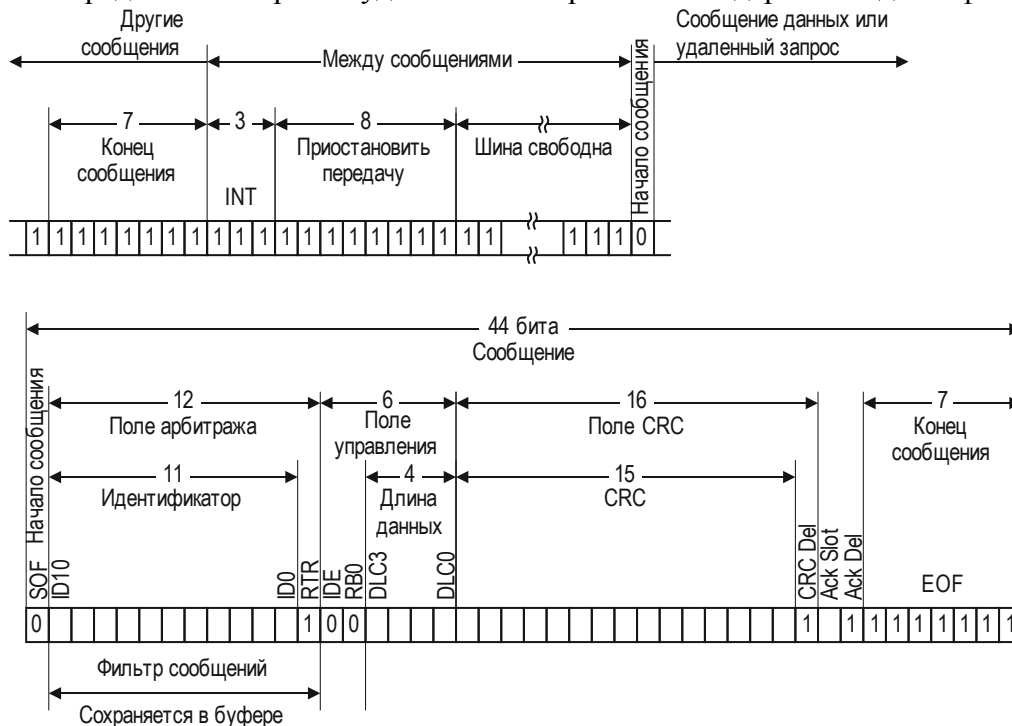


Рисунок 25.4 – Удаленный запрос данных (стандартный формат)

Имеются только два отличия содержимого удаленного запроса от сообщения данных:

- бит RTR в удаленном запросе передается в рецессивном состоянии;
- поле данных отсутствует (в сообщении не передается никаких данных, значение в поле DLC любое в пределах от 0 до 8).

В самом маловероятном случае, когда одновременно формируется удаленный запрос, и устройство пытается передать данные с одинаковыми идентификаторами, арбитраж будет выигран устройством, передающим данные, из-за доминантного состояния бита RTR.

Узел, который посылал запрос, получает данные немедленно.

## Сообщение об ошибке

Сообщение об ошибке формируется любым узлом, который обнаруживает ошибку на шине. Формат сообщения показан на рисунке 25.5.

Сообщение об ошибке состоит из двух полей: поле разделителя ошибки и поле флага ошибки. Возможны два типа поля флага ошибки, в зависимости от вида ошибки узла, обнаружившего ее.

Если ошибку обнаружил активный узел (как в примере на рисунке 25.5), тогда он прерывает передачу текущего сообщения, формируя флаг активной ошибки. Флаг активной ошибки состоит из шести последовательных доминантных бит, которые нарушают правила «бит-стаффинга» (правила заполнения и передачи бит на шине). Остальные узлы также обнаруживают ошибку и начинают формировать сообщение об ошибке. Таким образом, поле флага ошибки может содержать от 6 до 12 доминантных бит

(сформированных одним узлом или более). Поле флага ошибки дополняется разделителем ошибки, состоящим из восьми рецессивных бит и позволяющим перезапустить связь с шиной после обнаружения ошибки. После перехода шины в нормальное состояние узлы возобновляют передачу данных, остановленный узел повторяет передачу сообщения, переданного до этого с ошибкой.

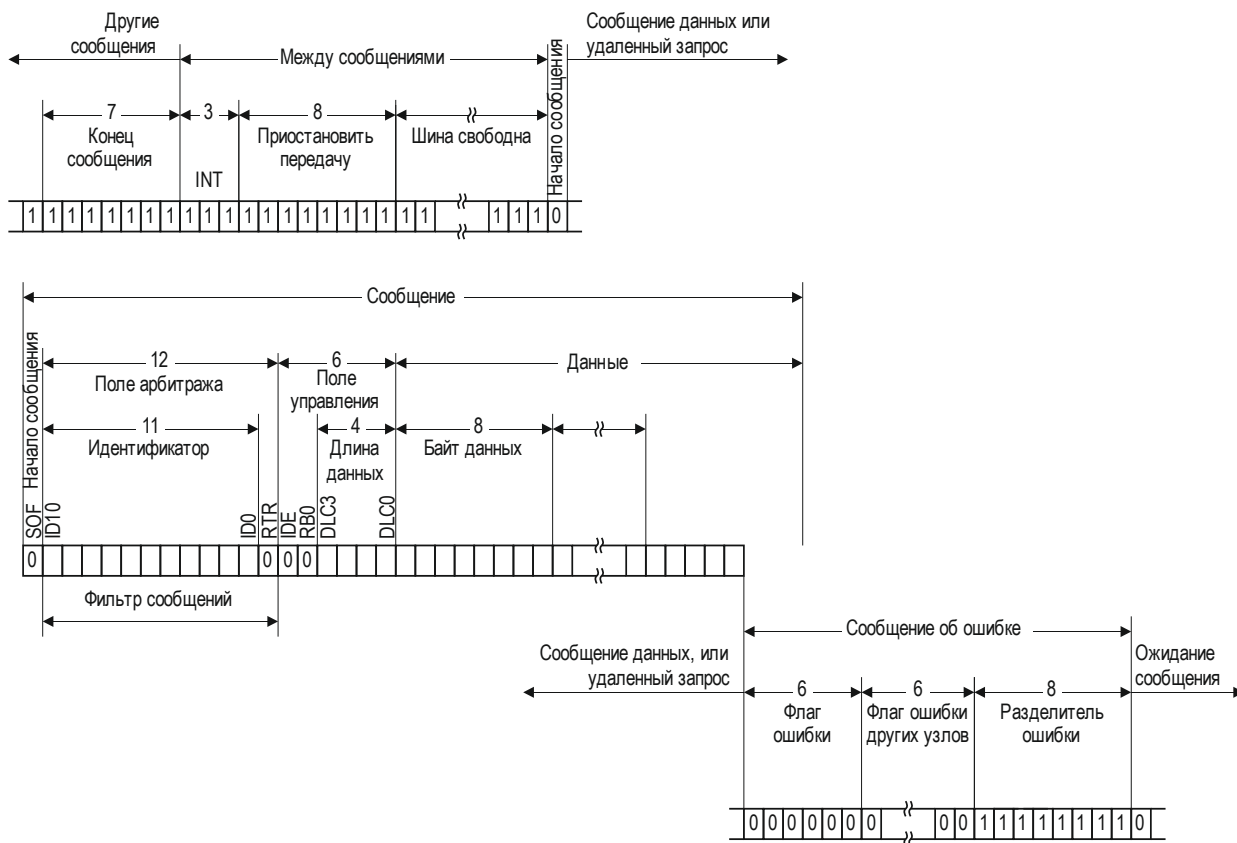


Рисунок 25.5 – Сообщение об ошибке

Если ошибку обнаружил пассивный узел, тогда он формирует флаг пассивной ошибки, состоящий из шести последовательных рецессивных бит, и затем разделитель ошибки. Таким образом, сообщение о пассивной ошибке состоит из 14 рецессивных бит. Это не нарушает правила «бит-стаффинга» на шине и не оказывает влияния на передачи других узлов. Исключение составляет узел, который передает данные узлу, обнаружившему ошибку. В этом случае правила «бит-стаффинга» нарушаются, и передача данных прекращается. После передачи пассивной ошибки узел должен ожидать шесть последовательных рецессивных бит для восстановления связи с шиной.

### Сообщение о перезагрузке

Формат сообщения о перезагрузке аналогичен формату сообщения об ошибке, но может быть сформирован, только когда шина простаивает.

Сообщение о перезагрузке показано на рисунке 25.6.

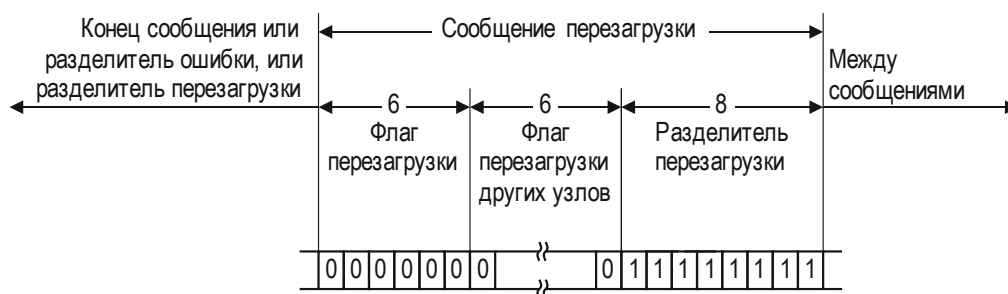


Рисунок 25.6 – Сообщение о перезагрузке

Разделитель перезагрузки состоит из восьми последовательных рецессивных бит.

Узел может сформировать сообщение о перезагрузке в двух случаях:

- между сообщениями обнаружен доминантный бит, что является ненормальным во время простоя шины;

- для задержки передачи нового сообщения.

Узел может последовательно сформировать не более двух сообщений перезагрузки.

Флаг перезагрузки состоит из шести последовательных доминантных бит. Другие узлы обнаруживают перезагрузку и начинают формировать ее самостоятельно. Поэтому на шине во время выполнения перезагрузки может быть до 12 доминантных бит.

## 25.2 Структура и функционирование контроллера CAN

В состав контроллера CAN входят два идентичных независимых узла CAN0 и CAN1, ОЗУ для хранения сообщений, которое является общим для узлов, и система управления. Контроллер CAN имеет следующие функциональные особенности:

- соответствие ISO 11898;
- функционирование согласно спецификации CAN 2.0b (активная версия);
- отдельные управляющие регистры для каждого из двух узлов;
- программируемая скорость передачи информации до 1 Мбит/с;
- гибкий и полный контроль передачи сообщений и обработки ошибок.

Контроллер CAN реализует 16 линий прерываний и 256 объектов сообщений для хранения сообщений и их параметров в ОЗУ. Каждый объект сообщения может быть привязан к любому из узлов, сконфигурирован для передачи или приема как стандартных, так и расширенных сообщений и удаленных запросов. Каждый объект имеет индивидуальную маску для фильтрации принимаемых сообщений. Объекты сообщений могут объединяться в классы, с разными уровнями приоритета, могут объединяться для построения структур FIFO произвольных размеров (до 256 объектов в одной структуре). Кроме того, реализована возможность попарного соединения объектов для формирования шлюзов для автоматической передачи сообщений между узлами. Параллельно с вышеуказанными свойствами объекты сообщений могут организовываться в списки с постоянно доступной реорганизацией (совместимость с TwinCan-устройствами, которые не имеют списков).

Структура контроллера CAN приведена на рисунке 25.7.

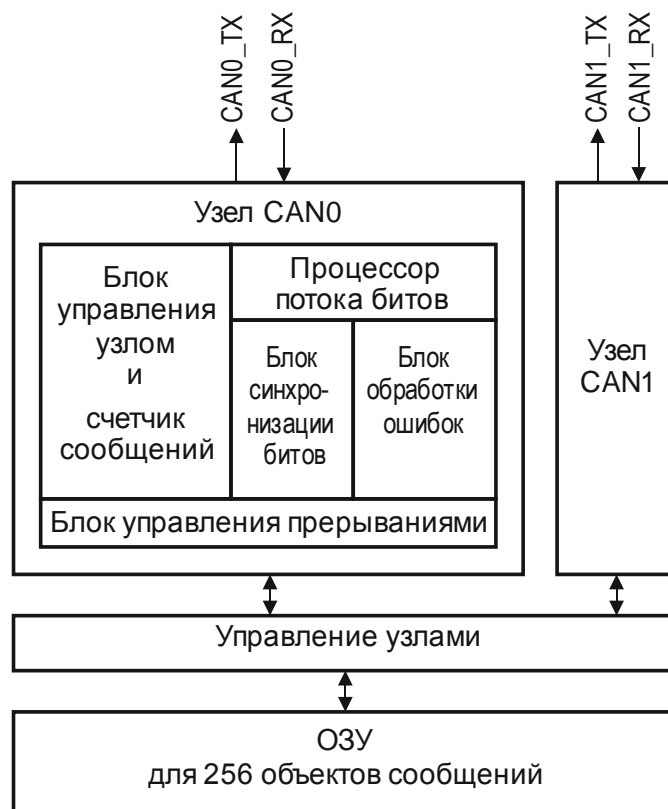


Рисунок 25.7 – Общая структура контроллера CAN

### Синхронизация

Тактирующим сигналом контроллера CAN является сигнал FIN (HLCK), приходящий с генератора тактовых сигналов. На основе этого сигнала посредством программируемого дробного делителя частоты формируется внутренний синхросигнал FCAN (FOUT), синхронизирующий работу контроллера и являющийся базовым синхросигналом для передачи/приема сообщений по внешней шине CAN.

### Включение контроллера CAN

По умолчанию, после сброса микроконтроллера контроллер CAN выключен. На это также указывает состояние флага DISS регистра CLC. Когда контроллер выключен, этот флаг установлен.

Для включения контроллера CAN следует записать ноль в бит DISR регистра CLC. После этого флаг DISS сбросится. Рекомендуется проверять состояние флага DISS, перед началом программирования регистров контроллера, которые не доступны в выключенном состоянии.

### Выключение контроллера CAN

Программно можно перевести контроллер CAN в режим выключения установкой бита DISR. Контроллер завершает все текущие операции, после чего устанавливает флаг DISS и отключает внутреннее тактирование, в связи с чем, все регистры становятся недоступными для обращения.

### Простой шины

Между передачами сообщений шина CAN находится в рецессивном состоянии. Для выполнения условий простой шины необходимо, чтобы было получено, как минимум, три рецессивных бита после завершения передачи/приема очередного сообщения.

### Анализ работы контроллера CAN

Для анализа работы контроллера CAN доступны два режима – общего анализа и внутренней петли.

Режим общего анализа включается установкой бита CALM регистра NCR узла и позволяет осуществлять независимый мониторинг работы узла, не затрагивая шину CAN. В этом режиме сообщения данных и удаленные запросы отслеживаются без участия узла в операциях на шине. Выходы узла находятся в рецессивном состоянии. Узел может получать сообщения данных, сообщения удаленных запросов и сообщения об ошибках, но работа узла на передачу запрещена. Полученные сообщения данных/удаленных запросов остаются без подтверждения (бит подтверждения остается в рецессивном состоянии), но принимаются и сохраняются (при совпадении идентификаторов) в соответствующих объектах сообщений. В ответ на входящие сообщения не выдается подтверждение, и не генерируются сообщения об ошибках. На удаленные запросы не выдаются сообщения данных, а сами сообщения данных не могут быть переданы установкой бита запроса передачи TXRQ регистра состояния объекта сообщения MOSTAT. Прерывания после приема генерируются (если это разрешено) для всех принятых сообщений, не содержащих ошибок.

Режим внутренней петли включается установкой бита LBM регистра NPCR и позволяет проводить внутреннее тестирование контроллера CAN, а также отладку управляющей программы без доступа к внешней шине CAN. Внутренняя петля состоит из внутренней шины CAN (внутри контроллера CAN) и переключателя выбора шины для каждого узла, см. рисунок 25.8. С помощью переключателя каждый узел CAN может быть подключен либо к внутренней шине (режим внутренней петли), либо к внешней шине (нормальный режим работы). Если выбран режим внутренней петли, то на внешнем передающем выводе узла CAN поддерживается рецессивный уровень сигнала, а состояние принимающего вывода игнорируется.

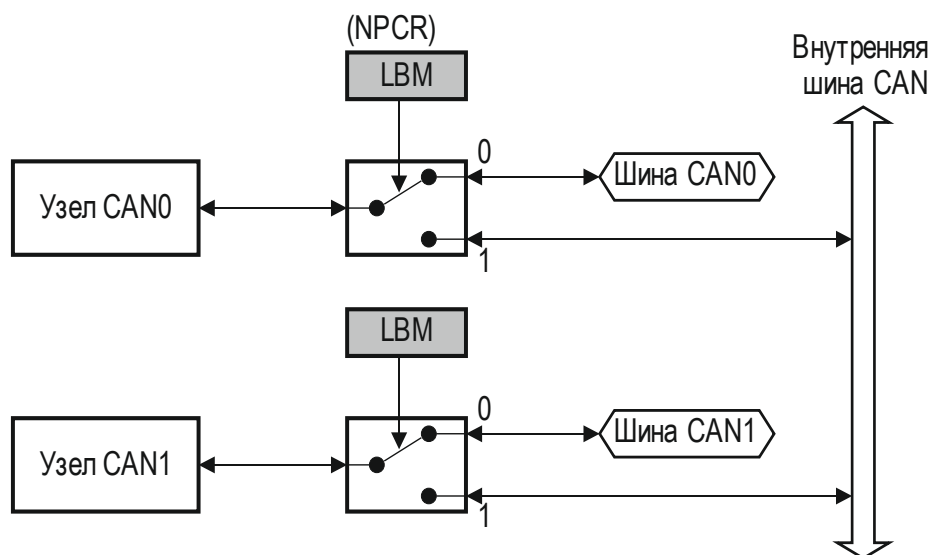


Рисунок 25.8 – Режим внутренней петли

Если оба узла CAN функционируют в режиме внутренней петли, они взаимодействуют друг с другом посредством внутренней шины CAN, не оказывая влияние на работу других модулей, функционирующих в нормальном режиме.

### Дробный делитель

Дробный делитель позволяет генерировать тактовый сигнал FOUT из входного FIN (HCLK) путем программирования делителя посредством регистра FDR.



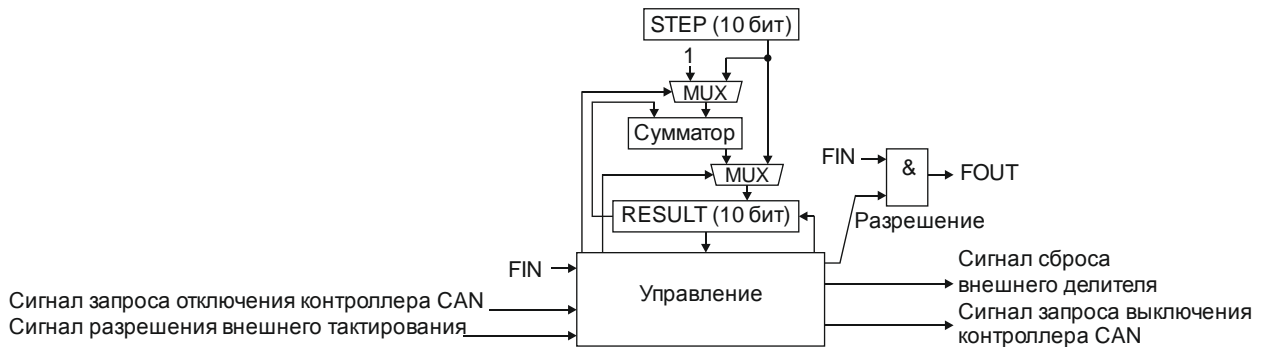


Рисунок 25.9 – Схема дробного делителя

Задаваемое значение частоты сигнала  $F_{IN}$  зависит от длительности передачи одного бита информации и должно быть  $n$ -кратно ей. Поскольку длительность передачи бита определяется количеством квантов времени  $Nt_q$ , (см. далее), то для расчета частоты сигнал  $F_{IN}$  в МГц следует пользоваться формулой:

$$f_{FIN} = n \times Nt_q, \quad (25.1)$$

где  $Nt_q$  – количество квантов времени  $t_q$ ;

$n$  – целое число, начиная с 1 (для задания кратности).

Дробный делитель делит  $F_{IN}$  путем умножения на величину  $1/val$  или величину  $1024/val$  для любого  $val$  от 0 до 1023, получая на выходе тактовый сигнал  $F_{OUT}$  ( $F_{CAN}$ ).

На рисунке 25.9 показана блок-схема дробного делителя. Логика дробного делителя работает по-разному, в зависимости от режима, задаваемого полем  $DM$ .

В режиме нормального деления ( $DM = 01b$ ) делитель работает как перегружаемый счетчик с шагом инкрементирования, равным единице. Состояние счетчика доступно посредством поля  $RESULT$ . Каждый раз, при переполнении (т. е. когда  $RESULT = 3FFh$ ), формируется импульс сигнала  $F_{OUT}$ , после чего в счетчик загружается значение из поля  $STEP$ .

Частота сигнала  $F_{OUT}$  определяется по формуле

$$f_{FOUT} = f_{FIN} \times 1/(1024 - STEPd), \quad (25.2)$$

где  $STEPd$  – значение поля  $STEP$  в десятичном формате.

Отсюда следует, что для получения сигнала  $F_{OUT}$  с частотой равной частоте сигнала  $F_{IN}$  значение  $STEP$  должно быть равно  $3FFh$ . На рисунке 25.10 показано формирование сигнала  $F_{OUT}$  при значении  $STEP = 3FDh$  ( $1021d$ ).

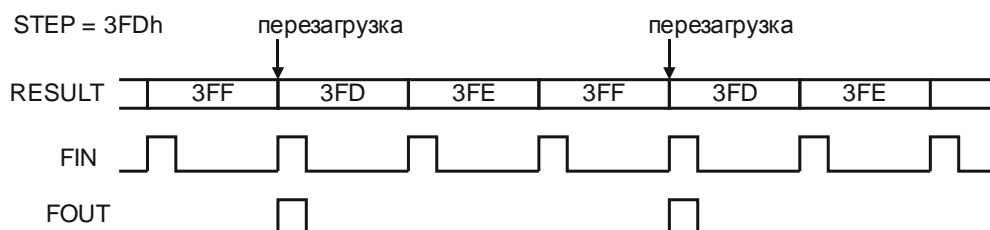


Рисунок 25.10 – Формирование сигнала  $F_{OUT}$  в нормальном режиме

В режиме дробного деления ( $DM = 10b$ ) делитель работает как перезагружаемый счетчик, но шаг инкрементирования в этом случае равен значению поля  $STEP$ . Если результат инкрементирования значения  $RESULT$  на величину  $STEP$  превышает  $3FFh$ , возникает переполнение счетчика, формируется импульс сигнала  $F_{OUT}$ , после чего в счетчик загружается значение, на которое результат инкрементирования превысил  $3FFh$ .

Частота выходного сигнала FOUT определяется по формуле

$$F_{OUT} = F_{IN} \times STEPd/1024d . \quad (25.3)$$

В целом, режим дробного деления позволяет программировать частоту сигнала FOUT с более высокой точностью, чем нормальный режим, но сигнал может иметь джиттер периода, не превышающий одного периода FIN, в связи с чем, не рекомендуется использовать режим дробного деления при высоких скоростях передач.

На рисунке 25.11 показано формирование сигнала FOUT при значении STEP = 234h (564d).  $F_{OUT} = F_{IN} \times 564/1024 = 0,55 \times F_{IN}$ .

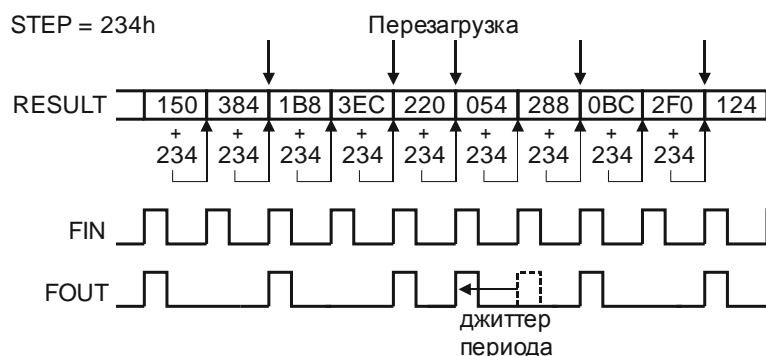


Рисунок 25.11 – Формирование сигнала FOUT в режиме дробного деления

Процесс выключения делителя начинается одновременно с возникновением запроса выключения контроллера CAN.

### Контроллер сообщений

Контроллер сообщений управляет обменом сообщениями между CAN узлами и памятью сообщений и выполняет следующие функции:

- фильтрация входящих сообщений для определения корректного объекта сообщения для сохранения полученных данных;
- определение объекта сообщения, содержимое которого будет передано в первую очередь (для каждого узла индивидуально);
- передача содержимого объекта сообщения к CAN узлу с параллельной вставкой в сообщение битов управления и состояния;
- осуществление буферизации FIFO и функционирования шлюза;
- объединение битов уведомления ждущих обработки сообщений.

### Управление прерываниями блока CAN

На рисунке 25.12 показана структура формирования запроса на прерывание.

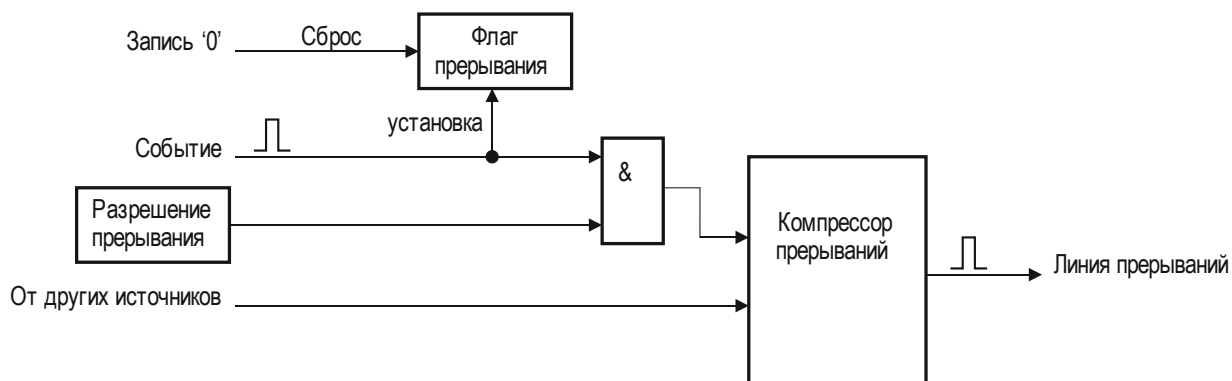


Рисунок 25.12 – Структура формирования запроса на прерывание

Событие, по которому должен быть сгенерирован запрос на прерывание, устанавливает флаг прерывания и (если разрешено) формирует запрос на прерывание на одной из 16 линий прерываний. Импульс запроса на прерывание генерируется независимо от состояния флага прерывания. Флаг прерывания может быть сброшен программно, записью нуля. Если к одной линии прерываний подключены несколько источников прерываний, то появление импульса от любого источника сформирует запрос на прерывание. Логика управления прерываниями использует схему компрессии прерываний.

Источниками прерываний являются:

- CAN узлы (восемь источников – по четыре для каждого узла);
- объекты сообщений (512 источников – по два для каждого объекта);
- программное прерывание (источник – регистр MITR).

Каждый аппаратный источник прерывания управляется четырьмя битами указателя прерываний, который определяет для него одну из 16 линий прерываний, что позволяет коммутировать на одну линию несколько источников прерываний. На рисунке 25.13 представлена схема коммутации линий прерываний.

Когда объект сообщения `Msg_x` генерирует запрос на прерывание по окончании приема или передачи сообщения, запрос передается на линию прерываний, выбранную в битовом поле `RXINP` или `TXINP` регистра `MOIPR` объекта сообщения `Msg_x`. Если количество объектов сообщений больше, чем количество линий прерываний, то на одну линию могут приходиться несколько запросов прерываний. Для разрешения конфликтов на линиях прерываний в контроллере CAN предусмотрен механизм распределения приоритетов для объектов сообщений.

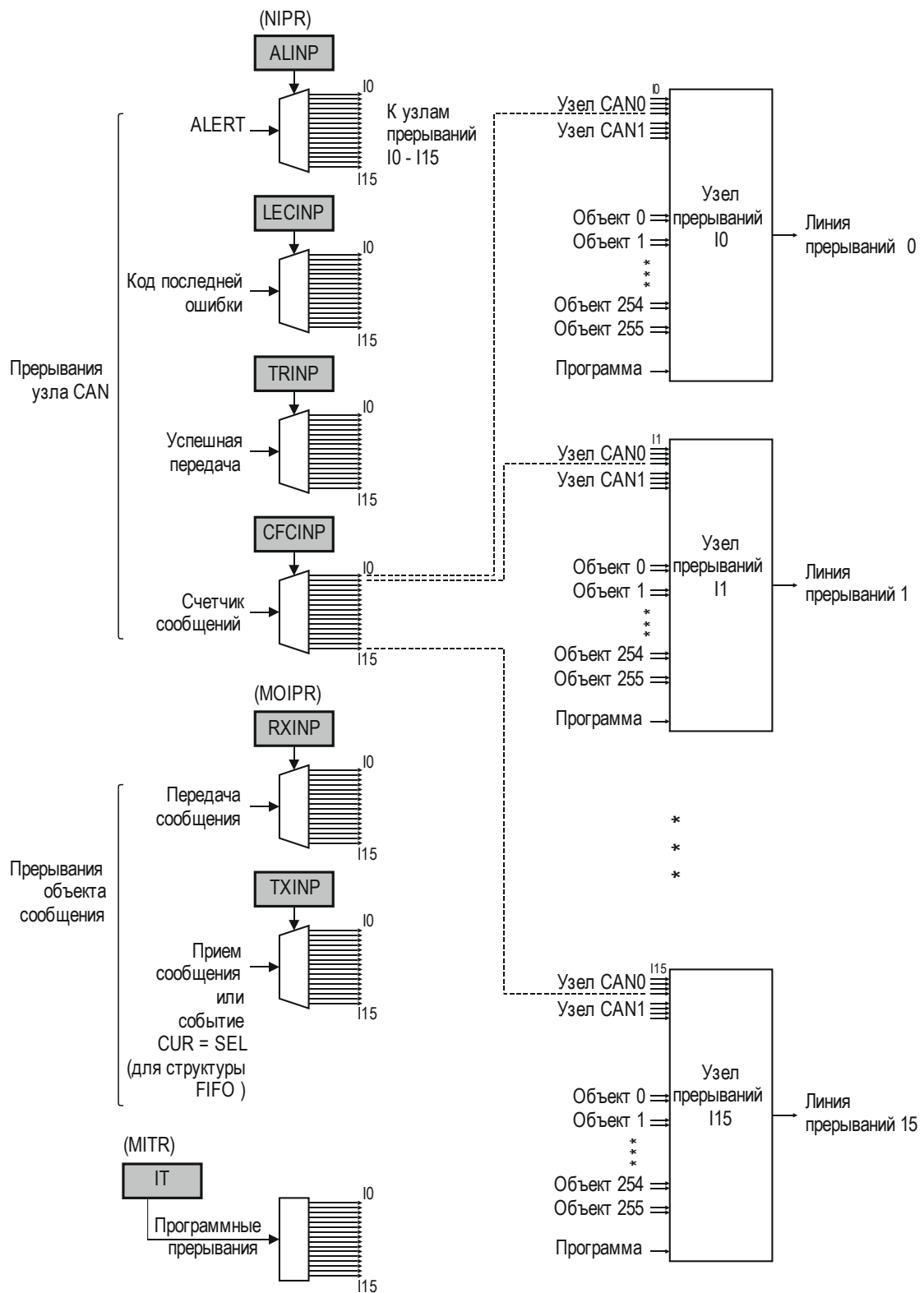


Рисунок 25.13 – Схема коммутации линий прерываний

### 25.3 Узел CAN

Каждый узел CAN имеет свою собственную логику управления и выдачи информации о состоянии и может быть сконфигурирован и работать независимо от другого узла.

Режим конфигурации включается установкой бита CCE регистра NCR. Режим конфигурации позволяет изменять параметры синхронизации битов и состояния счетчиков ошибок.

Конфигурация прерываний задается битами TRIE, ALIE и LECIE:

- бит TRIE управляет разрешением прерывания после передачи сообщения;
- бит ALIE управляет разрешением прерываний по ошибке;
- бит LECIE управляет разрешением прерывания по коду последней ошибки.

Регистр NSR отражает текущее состояние, содержит информацию о передачах и ошибках узла CAN.

#### Блок управления узлом CAN

Блок управления узлом координирует работу:

- разрешает/запрещает действия узла на шине;
- разрешает/запрещает и генерирует различные события, касающиеся работы узла (ошибка на шине, успешное завершение передачи сообщения), которые приводят к формированию запросов на прерывания;
- управляет счетчиком сообщений.

#### Блок синхронизации битов CAN

Согласно стандарту ISO 11898 время передачи одного бита разделено на сегменты, которые, в свою очередь, составлены из целочисленных отрезков времени, называемых квантами времени  $t_q$ , см. рисунок 25.14. Квант времени – фиксированная единица времени, получаемая из частоты синхронизации и делителя контроллера CAN.

Сегмент синхронизации  $T_{sync}$  позволяет синхронизировать начало обмена данными между передатчиком и приемником. Длительность сегмента всегда равна одному кванту времени.

Сегмент распространения  $T_{prop}$  используется для компенсации физического времени запаздывания сигнала в пределах сети. Длительность сегмента рассчитывается с учетом времени прохождения сигнала от передатчика к приемнику и обратно, входной задержки компаратора и задержки выхода драйвера и может составлять от 1 до 8 квантов времени.

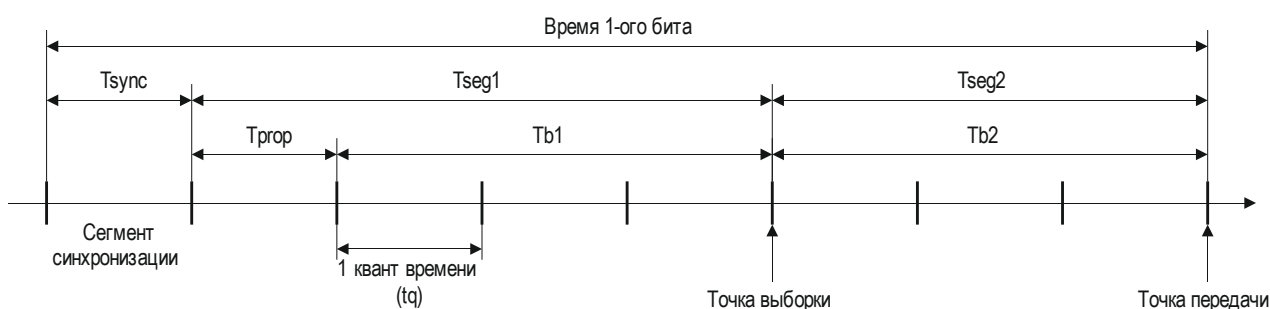


Рисунок 25.14 – Структура одного бита

Сегменты буфера фазы 1 и буфера фазы 2 –  $T_{b1}$  и  $T_{b2}$ , расположенные до и после точки выборки, используются для компенсации смещения фазы тактовых частот источника и приемника, обнаруживаемой после появления сегмента синхронизации, а также для оптимального расположения точки выборки полученного бита.

Точка выборки – момент, когда читается состояние шины CAN для определения принятого бита. Как правило, длительность временного интервала от начала бита до точки выборки составляет (60 – 70) % времени бита, в зависимости от системных параметров.

Сегмент распространения и сегмент буфера фазы 1 вместе составляют сегмент параметра 1 ( $T_{seg1}$ ), который определяется битовым полем TSEG1 регистра синхронизации битов NBTR (может быть записан, только если установлен бит CCE

регистра NCR). Согласно стандарту ISO, минимальная длительность сегмента параметра 1 должна составлять три кванта времени.

Сегмент параметра 2 (Tseg2) определяется битовым полем TSEG2 регистра NBTR и охватывает сегмент буфера фазы 2. Минимальная длительность сегмента параметра 2 составляет два кванта времени.

Согласно стандарту ISO, минимальная длительность одного бита, получающаяся сложением сегментов Tsync, Tseg1 и Tseg2, не должна быть менее 8 квантов времени.

Максимальная длительность бита – 25 квантов времени.

Примечание – Минимальное номинальное время передачи одного бита составляет 1 мкс, что соответствует скорости передачи 1 Мбит/с.

Формулы вычисления значений сегментов и времени одного бита Tbit:

- при DIV8 = 0 значение кванта времени

$$tq = (BRP + 1)/F_{OUT}; \quad (25.4)$$

- при DIV8 = 1 значение кванта времени

$$tq = 8 \times (BRP + 1)/F_{OUT}; \quad (25.5)$$

- Tsync = 1 × tq;

- Tseg1 = (TSEG1 + 1) × tq ≥ 3tq;

- Tseg2 = (TSEG2 + 1) × tq ≥ 2tq;

- Tbit = Tsync + Tseg1 + Tseg2 ≥ 8tq.

Чтобы компенсировать смещение фазы между частотами генераторов различных узлов шины CAN, каждое устройство должно синхронизироваться по фронту смены уровня сигнала на шине CAN от рецессивного к доминантному. Как только фронт обнаруживается, логика синхронизации сравнивает его текущее положение с ожидаемым и выполняет настройку значений параметров Tseg1 и Tseg2.

Контроллер CAN использует два механизма синхронизации – аппаратный и ресинхронизацию (синхронизация с восстановлением тактовых интервалов).

Аппаратная синхронизация выполняется по каждому фронту смены уровня сигнала на шине CAN от рецессивного к доминантному. При аппаратной синхронизации временные интервалы сегментов, из которых складываются времена битов, не изменяются в течение всего сообщения.

Ресинхронизация выполняется автоматическим удлинением сегмента Tseg1 или укорачиванием сегмента Tseg2. Максимальное значение изменения сегментов колеблется в пределах от 1 до 4 квантов времени. Синхронизация выполняется только при появлении фронта смены уровня сигнала на шине CAN от рецессивного к доминантному. Фиксированное значение максимального числа последовательных бит одинаковой полярности гарантирует своевременное восстановление синхронизации. Смещение фазы фронта смены уровня сигнала на шине отслеживается относительно сегмента синхронизации и измеряется в квантах времени.

Если величина фазового смещения меньше или равна запрограммированному значению ширины перехода ресинхронизации T<sub>SJW</sub>, выполняется аппаратная синхронизация.

Если величина смещения фазы больше, чем T<sub>SJW</sub>, а фазовое смещение положительно, то удлиняется сегмент Tseg1, в случае отрицательного фазового смещения укорачивается сегмент Tseg2.

Значение T<sub>SJW</sub> определяется полем SJW регистра NBTRx по формуле

$$T_{SJW} = (SJW + 1) \times tq. \quad (25.6)$$

Помимо прочего, должны соблюдаться следующие правила:

$$T_{seg1} \geq T_{sjw} + T_{prop} \quad (25.7)$$

$$\text{и } T_{seg2} \geq T_{sjw}. \quad (25.8)$$

Соотношения между максимальным отклонением частоты сигнала FOUT и сегментами буферов фаз и шириной перехода ресинхронизации следующие:

$$- \Delta F_{OUT} \leq T/2 \times (13 \times T_{bit} - T_{b2}); \quad (25.9)$$

$$- \Delta F_{OUT} \leq T_{sjw} / 20 \times T_{bit}, \quad (25.10)$$

где  $T$  – меньшее из  $T_{b1}$  и  $T_{b2}$ .

В итоге:

-  $T_{sync}$  составляет 1 квант времени;

-  $T_{prop}$  – от 1 до 8 квантов времени;

-  $T_{b1}$  – от 1 до 8 квантов времени;

-  $T_{b2}$  – выбирается равным двум квантам времени или равным сегменту  $T_{b1}$ , если его значение более двух квантов времени;

-  $T_{sjw}$  может составлять максимально 4 кванта времени, однако, в типовых приложениях достаточно 1.

Корректные значения параметров синхронизации битов должны быть записаны в регистр NBTR (доступен, если установлен бит CCE) до окончания инициализации (до сброса бита INIT регистра NCR), т. е. до начала работы CAN узла.

### **Процессор потока битов**

Процессор потока битов формирует (на основе содержимого объектов сообщений) сообщения данных и удаленные запросы непосредственно перед отправкой на шину CAN. Процессор потока управляет генератором CRC (генератор контрольной суммы) и добавляет контрольную сумму к сообщению. После вставки битов начала SOF и конца EOF сообщения, процессор потока начинает передачу сообщения по правилам арбитража шины CAN. В течение всего времени передачи сообщения процессор потока битов ведет мониторинг шины. Если обнаруживается несовпадение текущего (определяемого мониторингом) и ожидаемого (выдаваемого CAN узлом) уровня напряжения на шине CAN, генерируется ошибка и соответствующий ей запрос на прерывание. Код возникшей ошибки отражается в битовом поле LEC регистра NSR.

Корректность получаемых данных проверяется и подтверждается или не подтверждается кодом CRC. В случае отсутствия подтверждения возникает ошибка, генерируется запрос на прерывание и код ошибки выставляется в регистре NSR. Кроме этого, на шину выдается сообщение об ошибке.

После получения сообщения, не содержащего ошибок, и разбиения его на идентификатор и пакет данных, полученная информация записывается в буфер блока обработки сообщений, формируется соответствующее прерывание, и обновляются регистры состояния.

### **Блок обработки ошибок**

Блок обработки ошибок предназначен для выявления ошибок в работе устройств узла. В составе блока есть два счетчика: счетчик ошибок приема (поле REC в регистре NECNT) и счетчик ошибок передачи (поле TEC). Инкрементированием и декрементированием счетчиков управляет процессор потока битов.

Если процессор потока битов сам выявляет ошибку в процессе передачи, то счетчик TEC инкрементируется на 8. Инкрементирование на 1 происходит, если об ошибке сообщено внешним CAN-устройством путем генерирования сообщения об ошибке. Направление передачи с ошибочным сообщением и узел, сообщивший об ошибке передачи, указывают на соответствующие узлы CAN в регистрах NECNT, что используется для анализа ошибки.

В зависимости от значений счетчиков ошибок узел CAN может находиться в одном из трех состояний:

- активной ошибки;
- пассивной ошибки;
- отключен от шины.

Узел находится в состоянии активной ошибки, если значение каждого из счетчиков ошибок меньше 128. Узел в состоянии активной ошибки присоединен к шине CAN и посылает флаг активной ошибки при обнаружении ошибок.

Узел находится в состоянии пассивной ошибки, если значение хотя бы одного из счетчиков ошибок больше или равно 128. Узел подключен к шине, но при обнаружении ошибок посылает флаг пассивной ошибки. После передачи узел в состоянии пассивной ошибки будет ждать инициализации дальнейшей передачи.

Узел находится в состоянии отключения от шины, если значение счетчика ошибок TEC больше или равно 256. О том, что CAN узел находится в состоянии отключения от шины CAN, сигнализирует флаг BOFF регистра NSR. Узел в состоянии отключения от шины CAN не может работать с шиной CAN (выходные передатчики отключены).

Флаг EWRN регистра NSR устанавливается, когда хотя бы один из счетчиков достиг или превысил лимит ошибок, определенный в битовом поле EWRNLVL регистра NECNT. Как только значения обоих счетчиков перестанут превышать лимит ошибок, флаг EWRN сбросится.

### **Счетчик сообщений**

Счетчик сообщений может использоваться для получения информации о завершении передачи/приема сообщения соответствующего узла CAN. Подсчет сообщений осуществляется 16 разрядным счетчиком, который управляется регистром NFCR. Битовые поля CFMOD и CFSEL определяют режим работы и событие для инкрементирования счетчика.

Каждый узел CAN имеет в своем составе 16-разрядный счетчик сообщений/синхросчетчик, который подсчитывает количество принятых и переданных сообщений. Битовое поле CFSEL определяет один из трех режимов работы счетчика.

В режиме подсчета сообщений после успешной передачи и/или приема сообщения, содержимое счетчика копируется в битовое поле CFCVAL регистра MOIPR объекта сообщения Msg\_x, участвующего в пересылке данных. После чего счетчик сообщений инкрементируется.

### **Прерывания узла CAN**

Коммутация линий запросов прерываний показана на рисунке 25.15.

Узел может генерировать запросы на прерывания в случае:

- успешной передачи/приема сообщения;
- обнаружения кода последней ошибки;
- переполнения счетчика сообщений;
- состояния ALERT (состояние, возникающее, когда хотя бы один из счетчиков ошибок узла достиг значения своего лимита, изменяется состояние «отключен от шины», возникает ошибка длины списка или ошибка списка объектов).

После каждой успешной передачи или успешного приема сообщения генерируется (если разрешено соответствующими битами TXOK и RXOK) прерывание. Битовое поле TRINP регистра NIPR задает одну (из 16) линию прерывания.

Прерывание узла при возникновении кода последней ошибки формируется (если разрешено битом LECIE), если после модификации поля LEC его значение больше нуля. Битовое поле LECINP задает линию прерывания.



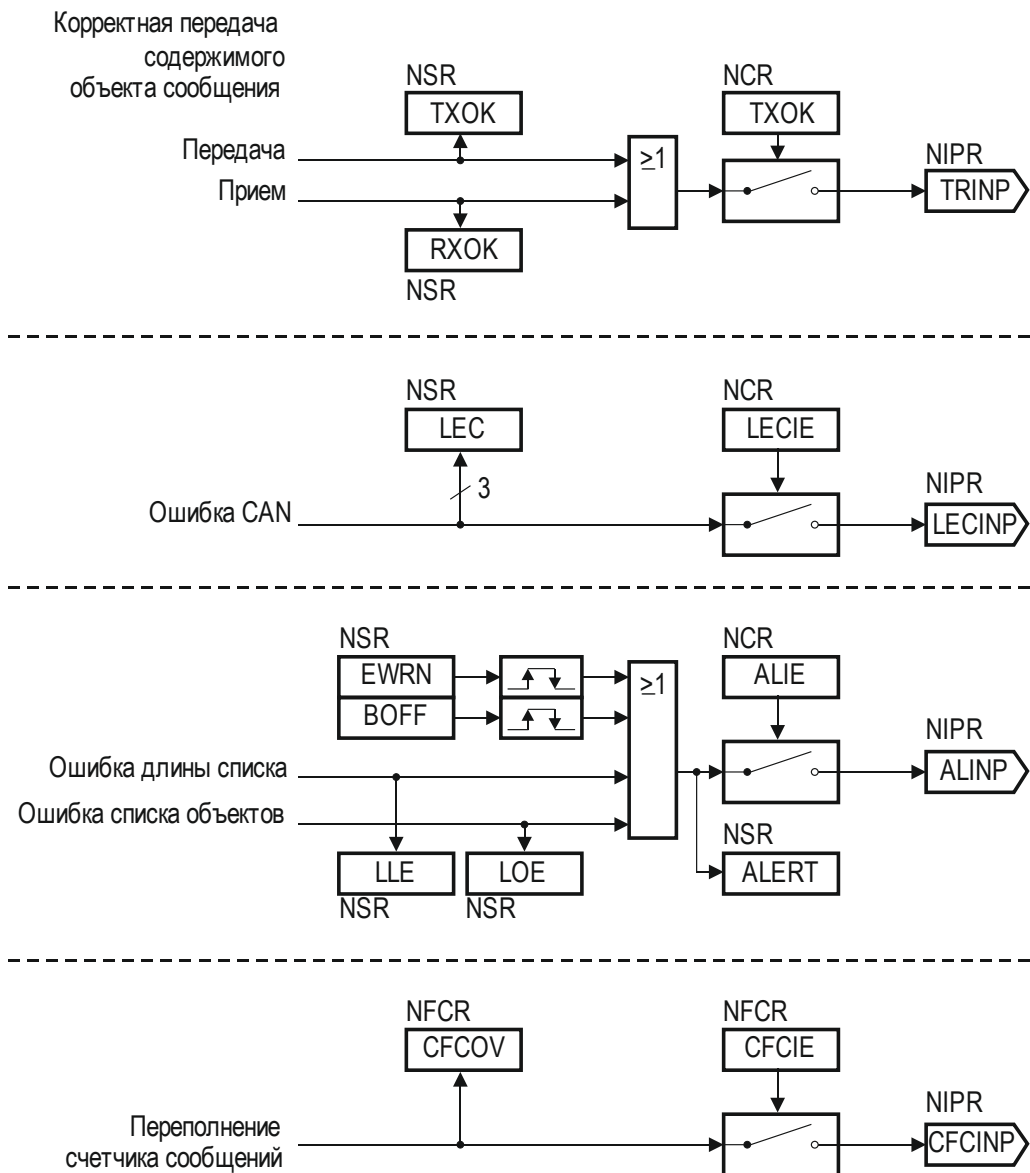


Рисунок 25.15 – Прерывания CAN узла

Прерывание узла при переполнении счетчика сообщений генерируется, если оно разрешено битом CFCIE регистра NFCR. Битовое поле CFCINP задает линию прерывания.

Прерывание ALERT может быть сформировано (если разрешено битом ALERT) любым из следующих событий:

- изменение состояния бита BOFF;
- изменение состояния бита EWRN;
- ошибка длины списка, которая также выставляет бит LLE;
- ошибка элемента списка, которая также выставляет бит LOE;
- бит INIT выставлен аппаратно.

Битовое поле ALINP задает линию прерывания.

В дополнение к аппаратным прерываниям есть возможность программного генерирования прерываний с использованием регистра прерываний MITR. Запись единицы в n-й разряд битового поля IT генерирует сигнал запроса прерывания на соответствующей ему n-ой линии прерываний (одной из 16). Установка нескольких битов приводит к параллельному генерированию запросов прерываний на соответствующих установленным битам линиях прерываний.

## 25.4 Объекты сообщений

### Регистры управления и состояния объектов сообщений

В состав каждого объекта сообщения входят девять 32-разрядных регистров:

- управления и состояния – МОСТР (только запись) и МОСТАТ (только чтение), доступные по одному адресу;
- арбитража – МОАР;
- данных – МОДАТАН и МОДАТАЛ;
- маски – МОАМР;
- указателя прерываний – МОИПР;
- указателя FIFO/шлюза – МОФГПР;
- управления функционированием – МОФСР.

Расположение регистров представлено на рисунке 25.16, где для примера взят пятый объект сообщения.

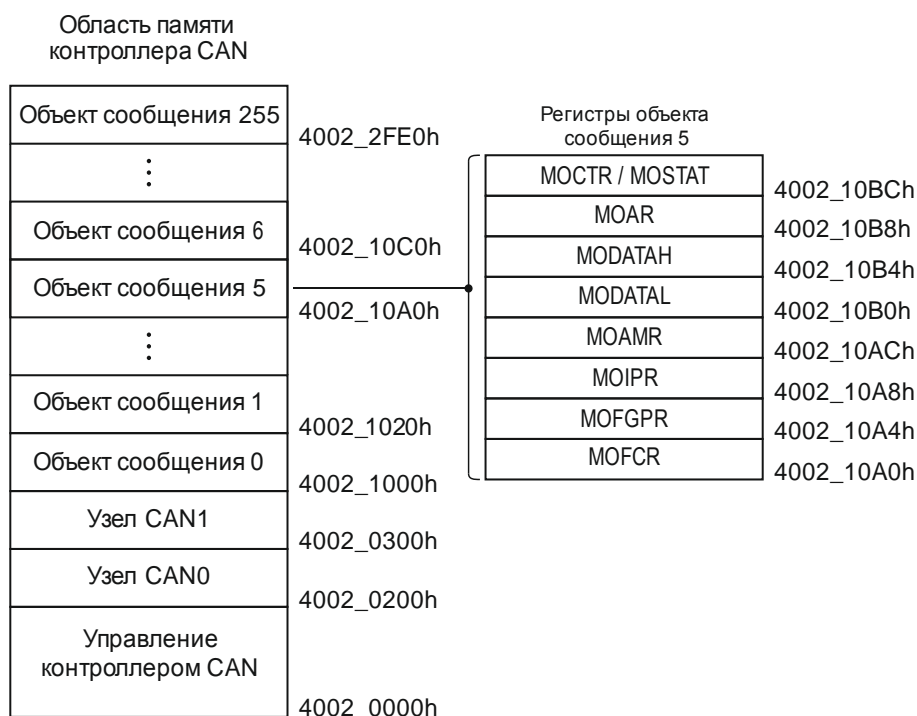


Рисунок 25.16 – Структура памяти регистров

Объекты сообщений контроллера CAN могут быть организованы в восемь списков, см. рисунок 25.17.

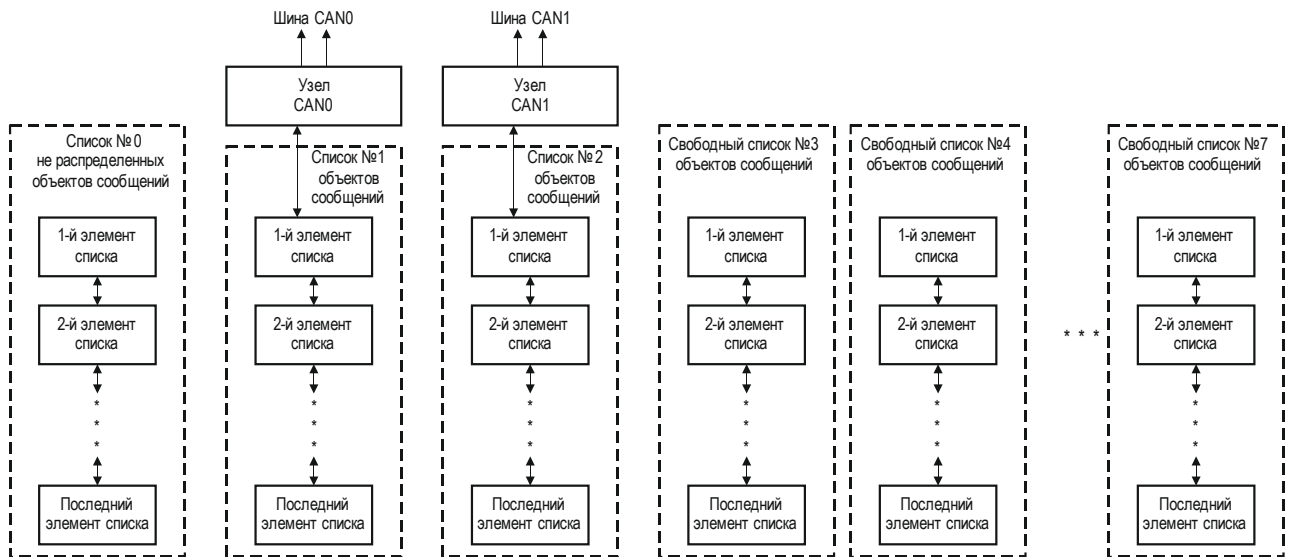


Рисунок 25.17 – Списки контроллера CAN

Каждый объект сообщения может быть добавлен в один из списков. Каждый узел CAN имеет свой список и соответствующий регистр списка. Регистр LIST1 отражает состояние списка №1 узла CAN0, регистр LIST2 – списка №2 узла CAN1.

Примечание – Узел может оперировать только с теми объектами сообщений, которые занесены в принадлежащий ему список.

Положение объекта сообщения Msg<sub>x</sub> в списке определяется посредством регистра MOSTAT, который содержит указатели на предшествующий ему и следующий за ним элементы списка (объекты). Нераспределенные между узлами CAN объекты сообщений по умолчанию организуются в отдельный список №0, состояние которого отражается в регистре LIST0. Остальные пять списков с номерами от 3 до 7 являются свободными (не принадлежат ни одному узлу) и имеют соответствующие регистры LIST3 – LIST7.

Примечание – Объекты сообщений, распределенные в списки с 3 по 7, не могут быть использованы узлами CAN.

Механизмы FIFO и шлюза оперируют с объектами сообщений независимо от их распределения по спискам, что дает возможность работы со всеми восемью списками. Следовательно, при использовании механизмов FIFO и шлюза следует внимательно следить за содержимым списков.

На рисунке 25.18 представлен вариант, когда объекты сообщений с номерами 3, 5 и 16 занесены в список №2, принадлежащий узлу CAN1. Состояние списка отражено в регистре LIST2.

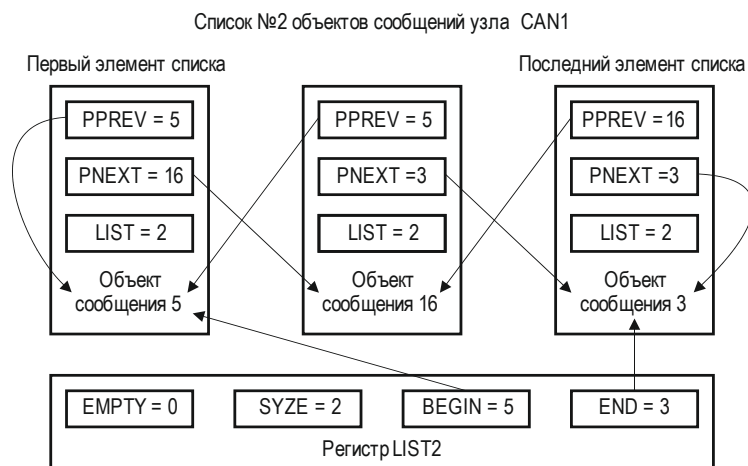


Рисунок 25.18 – Пример списка объектов сообщений

Значение поля BEGIN регистра LIST2 указывает на первый элемент списка (объект сообщения 5). Значение поля END указывает на последний элемент списка (объект сообщения 3). Количество элементов списка (количество объектов сообщений в списке) отражается в поле SIZE (значение SIZE всегда на единицу меньше количества элементов списка). Бит EMPTY является индикатором заполнения списка. Если список пуст, бит EMPTY установлен, в противном случае бит сброшен.

Каждый объект сообщения содержит номер списка (поле LIST), к которому он относится, а также указатели PNEXT и PPREV на следующий по списку объект сообщения и предшествующий, соответственно. Поле PPREV первого по списку объекта сообщения должно указывать на этот же объект. Поле PNEXT последнего по списку объекта сообщения должно указывать на этот же объект.

На рисунке 25.18 указатель PPREV пятого объекта сообщения (первого в списке) имеет значение 5h, а указатель PNEXT третьего объекта сообщения (последнего в списке) имеет значение 3h. Значение поля LIST всех трех объектов сообщений равно 2h.

Объект сообщения, у которого LIST = 0h относится к нулевому списку нераспределенных объектов. После сброса все объекты сообщений считаются нераспределенными. По умолчанию порядок элементов списка №0 следующий: объект сообщения (n – 1) является предыдущим объектом сообщения Msg\_x, а объект сообщения (Msg\_x + 1) – следующим.

Для просмотра структуры списка объектов сообщений узла достаточно обратиться к соответствующим регистрам LIST1/LIST2 и MOSTAT.

Структура списка управляется и изменяется посредством контроллера списка, который, в свою очередь, управляется панелью команд, основное назначение которой – упрощение внесения изменений в структуру списка, отслеживание этих изменений и проверка их корректности с помощью регистра PANCTR.

Панель команд запускается записью соответствующей команды в битовое поле PANCMD. До записи кода команды должны быть записаны соответствующие аргументы команды в битовые поля PANAR1 и PANAR2.

Примечание – Запись новых значений в поля PANAR1 и PANAR2 не изменяет сразу их содержимого. Новые значения сначала попадают в специальный теневой регистр. Далее, одновременно с записью кода команды в поле PANCMD, новые значения из теневого регистра переносятся в поля PANAR1 и PANAR2.

С записью корректного кода команды выставляется флаг BUSY, и в дальнейшем все попытки записи в регистр PANCTR игнорируются. Флаг BUSY остается активным, а панель команд заблокированной до тех пор, пока не завершится выполнение записанной команды.

После сброса микроконтроллера контроллер списка формирует список №0 нераспределенных объектов сообщений. Во время этой операции флаг BUSY установлен, и все обращения к объектам сообщений запрещены. По окончании этой операции флаг BUSY сбрасывается, и объекты становятся доступными.

В случае появления команды динамического распределения, по которой какой-либо элемент забирается из списка №0 и переносится в другой указанный список, наряду с битом BUSY, устанавливается бит RBUSY. Это указывает на то, что значения битовых полей PANAR1 и PANAR2 будут обновлены контроллером списка, следующим образом:

- номер объекта сообщения, переносимого из списка №0 нераспределенных объектов сообщений, записывается в PANAR1;
- если установлен бит ERR (седьмой бит поля PANAR2), значит список №0 пуст и выполнение команды завершается; если бит ERR сброшен – список №0 не пуст и команда выполняется.

Результаты выполнения команды динамического распределения записываются до того, как контроллер списка начнет процесс распределения. Как только результаты станут доступны, бит RBUSY сбрасывается. Это позволяет пользователю запрограммировать

настройки желаемого объекта сообщения, в то время как контроллер списка распределяет объекты. Во время операций со списками доступ к объектам сообщений не запрещен, но следует помнить, что любой доступ к регистрам объектов сообщений в течение процесса распределения объектов вносит задержку (в процесс), равную длительности доступа.

Код команды «нет операции» автоматически записывается в битовое поле PANCMD.

Новая команда может быть записана в любое время, когда бит BUSY сброшен.

Все битовые поля регистра PANCTR, исключая биты BUSY и RBUSY, могут быть записаны программно, что делает возможным сохранять и восстанавливать значения регистра PANCTR, если панель команд используется независимой подпрограммой обработки прерываний. Если возникает такая ситуация, то любые задачи, которые используют панель команд и которые могут прерывать выполнение других задач, тоже использующих панель команд, будут опрашивать состояние флага BUSY. До тех пор, пока флаг BUSY будет оставаться установленным, содержимое регистра PANCTR будет сохранено в соответствующей области памяти до операции восстановления. Как только подпрограмма обработки прерываний закончится, содержимое регистра PANCTR будет восстановлено.

До того, как объект сообщения, занесенный в список активного узла CAN, будет перенесен на другую позицию этого же списка или перенесен в другой список, бит MSGVAL регистра MOSTAT объекта сообщения Msg\_x должен быть очищен.

Примечание – Если требуется перераспределить объекты сообщений в списки повторно, необходимо приостановить работу узлов CAN (установить бит INIT регистра NCR), а после занесения объектов в списки возобновить ее (сбросить бит INIT).

## 25.5 Прием и передача сообщений

### Прием сообщения

После завершения приема сообщение сохраняется в объекте сообщения в соответствии с установленным алгоритмом, см. рисунок 25.19.

Помимо сохранения данных в объекте сообщения, контроллер CAN осуществляет обмен данными с ЦП.

При приеме сообщения информация сохраняется в объекте сообщения только в том случае, если установлен бит MSGVAL регистра MOSTAT. Если ЦП очищает бит MSGVAL, контроллер CAN останавливает запись в объект сообщения, и далее объект может быть реконфигурирован центральным процессором с последующей записью в него информации без участия контроллера CAN.

Полученное с шины сообщение может быть сохранено в объекте сообщения только в случае, если установлен бит RXEN. Контроллер CAN проверяет состояние бита RXEN только во время фильтрации принимаемого сообщения. После того, как сообщение принято, состояние бита не имеет значения и не оказывает влияния на дальнейшее сохранение данных в объекте сообщения.

Бит RXEN позволяет управлять блокированием объекта сообщения – после сброса бита RXEN полученное сообщение сохраняется в объекте сообщения, который получил приоритет, но в сохранении последующих сообщений этот объект не принимает участия.

Реконфигурация объекта сообщения центральным процессором во время работы контроллера CAN (например, сброс бита MSGVAL, изменение объекта сообщения и повторная установка бита MSGVAL) происходят следующим образом:

- объект сообщения получает приоритет;
- ЦП очищает бит MSGVAL для реконфигурации объекта сообщения;
- после реконфигурации ЦП снова устанавливает бит MSGVAL;
- завершается получение сообщения;
- если установлен бит MSGVAL, полученные данные сохраняются в объекте сообщения, генерируется запрос на прерывание, устанавливается соответствующий флаг;
- если сконфигурировано, производятся шлюзовые и FIFO операции.

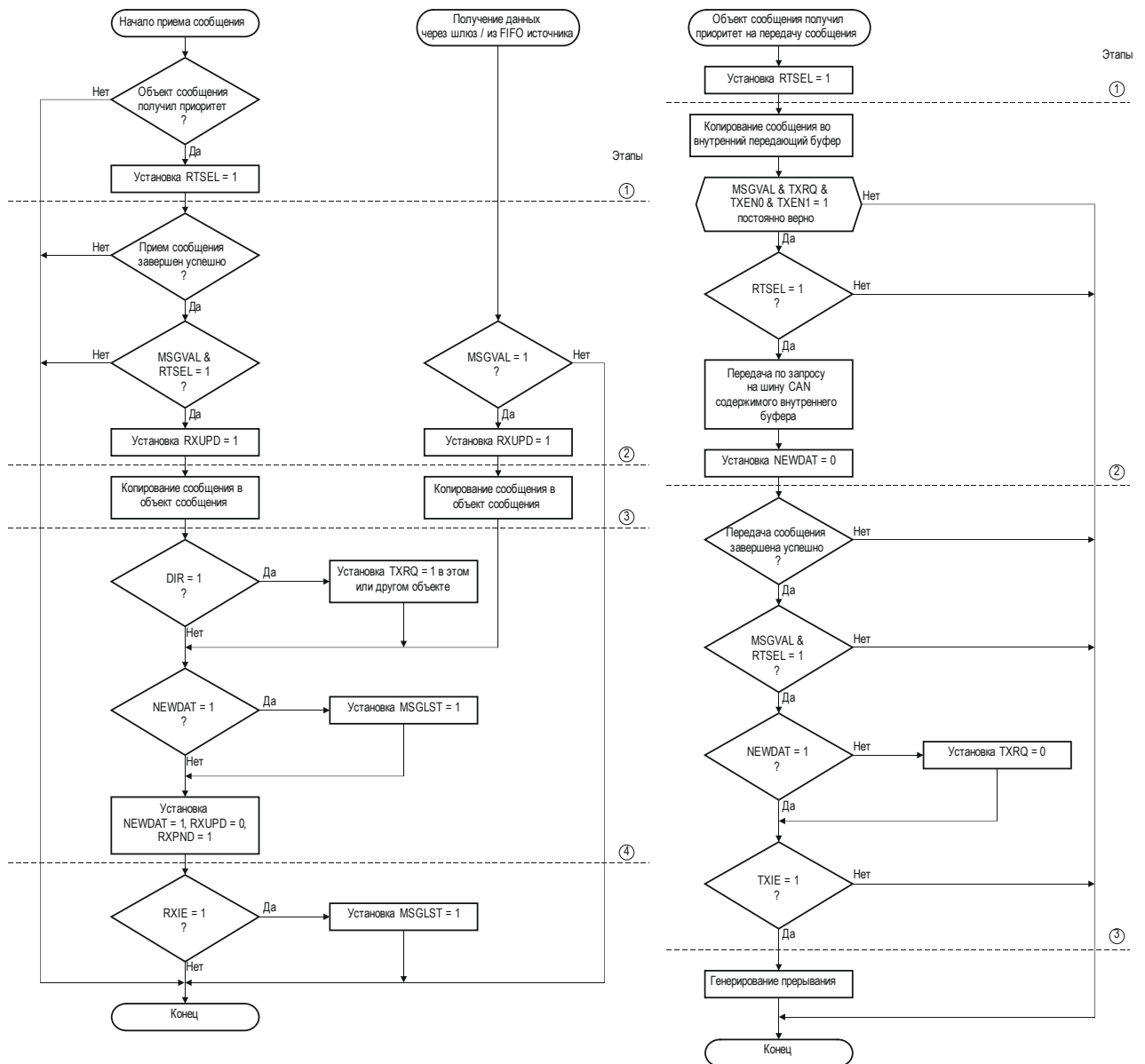


Рисунок 25.19 – Алгоритмы приема и передачи сообщения

Примечание – После реконфигурации объекта сохранение данных по завершении получения сообщения может быть нежелательным. Запретить запись данных в объект сообщения можно посредством бита RTSEL.

После получения объектом сообщения приоритета, его бит RTSEL устанавливается контроллером CAN, открывая, таким образом, объект сообщения для записи. После приема сообщения контроллер CAN дополнительно проверяет возможность записи в объект сообщения, а именно – установлен ли все еще бит RTSEL. И только в том случае, если бит RTSEL установлен, полученные данные сохраняются в объекте сообщения (вместе со всеми последующими действиями, которые указаны выше).

Если во время операций контроллера CAN объект сообщения становится некорректным (сброс бита MSGVAL), бит RTSEL должен быть сброшен до того, как бит MSGVAL будет установлен снова, или, по крайней мере, одновременно с ним. Это необходимо для предотвращения сохранения старой информации в объекте сообщения.

Реконфигурация объекта сообщения должна происходить следующим образом:

- сброс бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и далее установка бита MSGVAL.

Индикатором процесса сохранения (изменения) данных в объекте сообщения является флаг RXUPD, который выставляется с началом процесса сохранения (изменения) и сбрасывается с его окончанием.

После сохранения полученного сообщения (идентификатора, бита IDE, кода длины данных, поля данных, в случае сообщения данных) выставляется флаг NEWDAT. Если к моменту выставления (завершение сохранения/изменения данных) флаг NEWDAT был уже установлен, выставляется флаг MSGLST, который говорит о том, что произошла потеря данных.

Флаги RXUPD и NEWDAT позволяют произвести чтение корректных данных из объекта сообщения во время текущих операций контроллера CAN. Рекомендуемая последовательность действий следующая:

- сброс флага NEWDAT;
- чтение данных (идентификатор, данные и т. д.) из объекта сообщения;
- проверка флагов NEWDAT и RXUPD – оба флага должны быть сброшены.

В случае невыполнения этого условия – возвращение к первому действию;

- если флаги NEWDAT и RXUPD сброшены, то содержимое объекта сообщения корректно и не используется контроллером CAN в течение операции чтения.

Поведение флагов RXUPD, NEWDAT и MSGLST идентично как для сообщений данных, так и для сообщений удаленных запросов.

### **Передача сообщения**

Алгоритм передачи сообщений показан на рисунке 25.19. Одновременно с копированием данных (идентификатора, бита IDE, бита RTR, равного биту DIR, кода длины данных и собственно данных) из объекта сообщения, содержимое которого должно быть передано во внутренний передающий буфер соответствующего узла CAN, для контроля соблюдения четкой последовательности выполнения всех операций устанавливаются биты состояния.

Сообщение может быть передано только в случае, когда все четыре бита MSGVAL, TXEN0, TXEN1 и TXRQ установлены.

Бит RTSEL выставляется после того, как объект сообщения получает приоритет для передачи своего содержимого. Когда данные объекта сообщения копируются в передающий буфер, бит RTSEL проверяется, и если он установлен, сообщение передается. После успешной передачи сообщения бит RTSEL проверяется снова, и если он установлен, осуществляются дальнейшие операции.

Для полной и завершенной реконфигурации корректного объекта сообщения должны быть выполнены следующие шаги:

- очистка бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и установка бита MSGVAL.

Сброс бита RTSEL гарантирует как полное отключение объекта сообщения от текущей передачи, так и то, что никакие операции (копирование данных в передающий буфер, включая сброс бита NEWDAT, очистка бита TXRQ, прерывание сообщения и т. д.), относящиеся к старой конфигурации этого объекта сообщения, не повлияют на новую конфигурацию после установки бита MSGVAL.

После завершения передачи содержимого объекта сообщения в передающий буфер узла CAN, флаг NEWDAT аппаратно сбрасывается, тем самым обозначая, что объект сообщения открыт для записи новых данных.

Если после успешной передачи сообщения (на шину CAN) флаг NEWDAT все еще остается сброшенным (в объект сообщения не были записаны новые данные), флаг TXRQ аппаратно сбрасывается. Если же флаг NEWDAT был установлен программно (в связи с необходимостью передачи новых данных), флаг TXRQ не сбрасывается, тем самым разрешая передачу новых данных.

## 25.6 Фильтрация сообщений

### Фильтрация при получении сообщений

При получении узлом CAN сообщения определяется объект сообщения, в котором будут сохранены получаемые данные в случае успешного приема.

Объект сообщения считается корректным для приема, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла, который принимает сообщение;

- бит MSGVAL установлен;

- бит RXEN установлен;

- бит DIR равен битам RTR принимаемого сообщения. Если бит DIR установлен, объект сообщения (объект передачи) может принять только сообщение удаленного запроса. Если бит DIR сброшен (объект приема), объект сообщения может принять только сообщение данных;

- если бит MIDE установлен, то бит IDE получаемого сообщения оказывает следующее влияние:

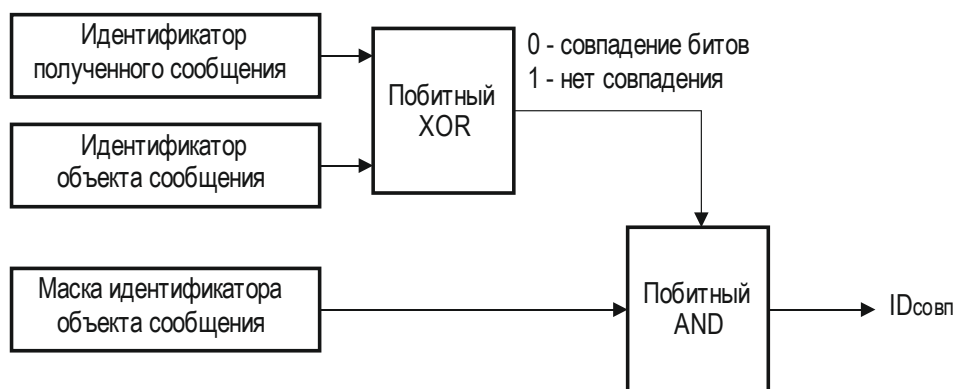
- если бит IDE (регистр MOAR) установлен, то бит IDE принимаемого сообщения должен быть равен единице (расширенный идентификатор);

- если бит IDE сброшен, бит IDE принимаемого сообщения должен быть равен нулю (стандартный идентификатор);

- если бит MIDE сброшен, значение бита IDE принимаемого сообщения не важно, т.е. допускаются сообщения, как со стандартным, так и с расширенным идентификатором;

- идентификатор полученного сообщения полностью (побитно) совпадает с идентификатором, хранящимся в регистре MOARn объекта сообщения, за исключением битов, закрытых маской регистра MOAMRn, значение которых не важно.

На рисунке 25.20 показан пример проверки идентификатора.



ID<sub>совп</sub> = 0: идентификатор ID полученного сообщения совпал с ID объекта сообщения

ID<sub>совп</sub> > 0: идентификатор ID полученного сообщения не совпал с ID объекта сообщения

Рисунок 25.20 – Проверка идентификатора полученного сообщения

Среди всех объектов сообщений, которые отвечают указанным выше критериям, для сохранения полученного сообщения выбирается объект с наивысшим приоритетом. Для задания приоритета используется поле PRI в регистре MOAR. Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI приоритетным считается объект сообщения, который предшествует следующему в списке.



### Фильтрация при передаче сообщений

Когда требуется передача содержимого какого-либо объекта сообщения, в соответствующих управляющих регистрах выставляются флаги, указывающие на необходимость передачи. Объект сообщения считается корректным для передачи, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла CAN;
- флаг MSGVAL установлен;
- флаг TXRQ установлен;
- флаги TXEN0 и TXEN1 установлены.

Может возникнуть ситуация, когда передачи требуют одновременно несколько объектов сообщений. Среди всех объектов, которые отвечают указанным выше критериям, для передачи выбирается объект с наивысшим приоритетом.

Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI разных объектов приоритет определяется следующим образом:

- при PRI = 10b – согласно правилам арбитража передачи сообщения;
- при PRI = 01b/11b приоритет имеет объект сообщения, который предшествует следующему в списке.

Объект сообщения, являющийся корректным для передачи и имеющий приоритет, будет осуществлять передачу первым. Остальные объекты сообщений будут переданы по очереди, согласно их приоритетам.

Объект сообщения определяется как стандартный объект сообщения, если в регистре MOFCR значение битового поля MMC равно нулю. Стандартный объект сообщения может принимать и передавать сообщения, согласно правилам, описанным выше.

На рисунке 25.21 показано формирование запроса на передачу объекта сообщения.

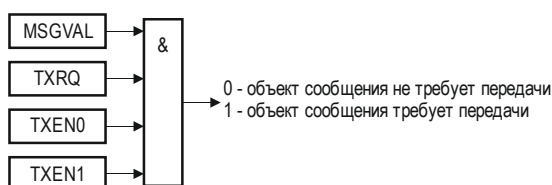


Рисунок 25.21 – Формирование запроса на передачу объекта сообщения

### 25.7 Удаленные запросы

После получения узлом CAN сообщения удаленного запроса и сохранения его в объекте сообщения, выставляется бит запроса передачи для ответа на удаленный запрос (отправка сообщения данных) или для автоматического повторения запроса.

В зависимости от состояния бита FRREN объекта сообщения, который принял сообщение удаленного запроса, возможны два варианта действий:

- если бит FRREN сброшен, то устанавливается флаг TXRQ этого объекта;
- если бит FRREN установлен, то устанавливается флаг TXRQ того объекта, на который указывает поле CUR объекта, принявшего удаленный запрос. При этом поле CUR не меняет своего значения.

#### Состояние регистров объекта сообщения, передающего сообщение удаленного запроса

У объекта сообщения, передающего сообщение удаленного запроса, в регистре MOSTAT должен быть сброшен бит DIR (объект передает сообщение данных) и установлены биты TXEN0, TXEN1, MSGVAL и TXRQ. Значение идентификатора в регистре MOAR передающего объекта сообщения должно быть равно значению идентификатора принимающего объекта сообщения (или совместно с регистром MOAMR

обеспечивать успешное прохождение фильтрации), чтобы сообщение удаленного запроса было принято принимающим объектом другого узла. Само сообщение удаленного запроса должно содержать идентификатор принимающего объекта сообщения, поэтому значение регистра MODATAL передающего объекта сообщения должно быть равно значению регистра MOAR принимающего объекта.

#### **Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса при FRREN = 0**

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR (объект принимает сообщение удаленного запроса), TXEN0 и TXEN1 (если отвечать на запрос будет сам), RXEN и MSGVAL. Регистры MODATAL и MODATAN должны содержать данные, которые будут переданы в ответ на запрос.

#### **Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса (при FRREN = 1) и содержащего данные для ответа на запрос**

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR, RXEN и MSGVAL. Битовое поле CUR должно указывать на номер объекта сообщения (должен находиться в том же узле, что и объект принявший сообщение удаленного запроса), содержащего данные, предназначенные для передачи в ответ на поступивший удаленный запрос.

В свою очередь у объекта сообщения, хранящего данные для отправки в ответ на запрос, должны быть установлены биты DIR, (объект передает сообщение данных), TXEN0, TXEN1 и MSGVAL. Бит TXRQ устанавливается автоматически при приеме сообщения удаленного запроса принимающим объектом сообщения.

Прием ответа на запрос (переданного сообщения данных) осуществляется стандартным объектом сообщения запрашивающего узла CAN (обмен данными происходит между объектом сообщения, хранящим данные для отправки в ответ на запрос, и объектом сообщения запрашивающего узла).

### **25.8 Дополнительные режимы передачи**

Дополнительно имеются два режима, каждый из которых может быть выбран индивидуально:

- режим передачи данных с защитой от повторений;
- режим однократной пересылки данных.

#### **Режим передачи данных с защитой от повторения**

Режим передачи данных с защитой от повторения выбирается установкой бита SDT регистра MOFCR.

После приема сообщения данных и сохранения его в объекте с установленным битом SDT, бит MSGVAL этого объекта аппаратно сбрасывается, чтобы исключить возможность повторного приема и записи в этот объект. Этот режим нельзя использовать для базового объекта FIFO структуры.

В ответ на сообщение удаленного запроса, принятое объектом с установленным битом SDT, будут отправлены данные из объекта сообщения, на который указывает поле CUR объекта, принявшего удаленный запрос. После этого бит MSGVAL объекта принявшего сообщение удаленного запроса сбросится.

**Примечание** – Объект, принявший сообщение удаленного запроса, не может быть источником данных, передаваемых в ответ на запрос. Это означает, что в данном режиме бит FRREN объекта, принявшего удаленный запрос, обязательно должен быть установлен.

### Режим однократной пересылки данных

Режим однократной пересылки данных выбирается установкой бита STT.

Бит TXRQ сбрасывается, когда содержимое объекта сообщения копируется в передающий буфер узла CAN. Таким образом, в дальнейшем, при неудачной (вследствие ошибок) пересылке сообщения по CAN-шине, повторной передачи не будет.

### 25.9 FIFO структура объектов сообщений

Регистр MOFGPR объекта сообщения Msg\_x содержит установки указателей на объекты сообщений, которые используются при операциях FIFO и шлюзовых операциях.

В случае сильной загрузки ЦП обработка серии сообщений может быть затруднена – например, вследствие получения и/или передачи большого числа сообщений за малые промежутки времени. Для таких случаев предусмотрена система буферов быстрого ввода-вывода, так называемая FIFO структура, которая может функционировать автоматически, и позволяет избежать потери принимаемых сообщений, минимизировать время подготовки сообщений к отправке, а также генерировать прерывания по окончании операций.

Допускается организация нескольких параллельных FIFO структур. Число структур и их составляющих зависит только от количества доступных объектов сообщений. FIFO структура может быть создана, изменена и удалена в любой момент времени, даже во время операций контроллера CAN.

На рисунке 25.22 представлена основная FIFO структура. Она состоит из одного базового объекта и n-ого числа вспомогательных объектов.

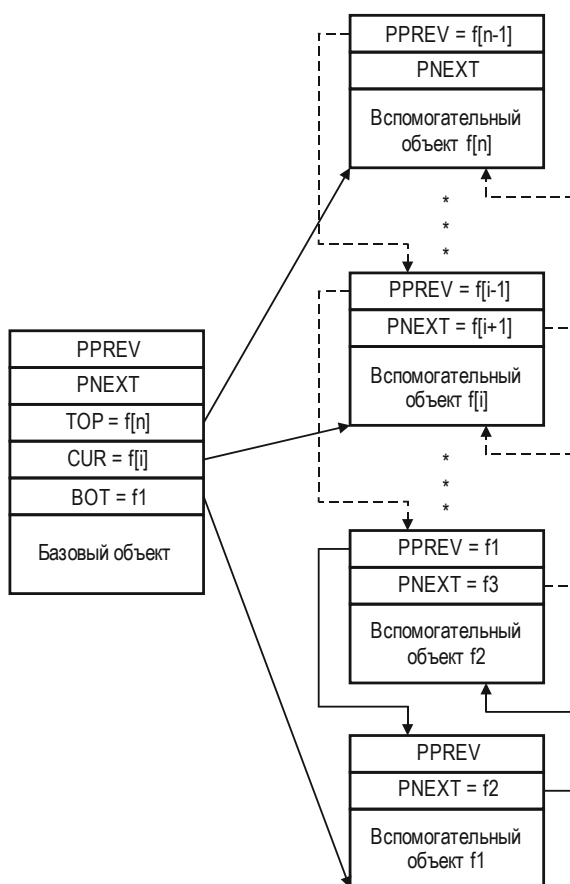


Рисунок 25.22 – FIFO структура с базовым объектом и n вспомогательными объектами

Вспомогательные объекты передающей FIFO-структуры объединяются последовательно в списки (подобно спискам объектов сообщений). Базовый объект может

быть занесен в любой список. Хотя на рисунке 25.22 базовый объект не относится ни к одному из списков, он может быть вставлен в любую последовательность вспомогательных объектов. Это означает, что базовый объект одновременно является и вспомогательным объектом (шлюзовые операции не возможны). Порядковые номера объектов сообщений (0, 1, 2 и т. д.) не имеют никакого значения при FIFO операциях с объектами.

Вспомогательные объекты должны быть определены в общий список (так как они последовательно связаны). С помощью указателей (битовые поля BOT, CUR и TOP) можно присоединять базовый объект к вспомогательному объекту, независимо от того, принадлежат базовый и вспомогательный объекты одному списку или разным спискам, но базовый должен быть первым в списке в таком случае.

Минимальная FIFO структура может состоять из одного объекта сообщения, который будет одновременно являться и базовым, и вспомогательным (фактически не используется). Максимальная FIFO структура может включать в себя все 256 объектов сообщений.

В базовом объекте FIFO границы установлены: поле BOT указывает на самый младший элемент FIFO структуры, поле TOP – на самый старший элемент, поле CUR – на вспомогательный объект, который в настоящий момент выбран контроллером CAN для передачи сообщения. Как только начинается передача, в CUR записывается номер следующего по списку вспомогательного объекта сообщения (CUR = PNEXT используемого объекта). Если значение битового поля CUR достигло номера старшего элемента списка (CUR = TOP), то следующим значением будет BOT (реализация автоматического перехода в начало списка). Таким образом, реализуется замкнутая FIFO структура, в которой битовые поля TOP и BOT устанавливают связь между началом и концом списка.

Битовое поле SEL позволяет определить вспомогательный объект, в пределах списка, для которого генерируется прерывание всякий раз, когда указатель CUR достигает значения указателя SEL. Также битовое поле SEL позволяет отследить окончание запланированной передачи серии сообщений или выдать прерывание, предупреждающее о том, что FIFO структура становится заполненной.

Вспомогательные объекты приемной FIFO-структуры могут принадлежать списку любого узла.

### **FIFO структура для приема**

FIFO структура для приема используется для буферизации входящих сообщений данных и удаленных запросов.

FIFO структура для приема активируется записью значения 0001b в битовое поле MMC регистра MOFCR базового объекта. Эта запись автоматически определяет объект как базовый объект приема FIFO. Типы вспомогательных объектов FIFO не имеют значения при операциях.

Когда базовый объект FIFO получает сообщение, оно сохраняется не в этом базовом объекте, а во вспомогательном объекте сообщения, на который указывает битовое поле CUR. При этом по умолчанию предполагается, что для вспомогательного объекта MMC = 0000b (действительное значение MMC игнорируется), и никаких операций фильтрации принимаемого сообщения не производится.

Одновременно с приемом сообщения текущее значение указателя CUR базового объекта меняется на номер следующего по списку вспомогательного объекта FIFO структуры. Этот вспомогательный объект будет использован для приема следующего сообщения.

Если установлен флаг OVIE регистра MOFCR базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения. Это прерывание генерируется на узле прерываний с указателем TXINP

базового объекта сразу после сохранения полученного сообщения во вспомогательном объекте. Прерывания генерируются, если это разрешено битом TXIE.

Следует помнить, что сообщение сохраняется в базовом и вспомогательном объектах FIFO, только если установлен бит MSGVAL.

Во избежание непосредственного приема сообщения вспомогательным объектом, как если бы он был независимым объектом и не принадлежал FIFO структуре, флаги RXEN всех вспомогательных объектов должны быть сброшены. Состояние флага RXEN неважно в случае, когда вспомогательный объект занесен в список, не связанный с узлом CAN.

### **FIFO структура для передачи**

FIFO структура для передачи используется для буферизации серий сообщений данных или удаленных запросов, которые должны быть отправлены. FIFO структура для передачи состоит из базового объекта и одного или более вспомогательных объектов.

FIFO структура для передачи активируется записью значения 0010b в поле MMC регистра MOFCR базового объекта. В отличие от FIFO структуры для приема, в битовые поля MMC вспомогательных объектов (FIFO структуры для передачи) должно быть записано значение 0011b. Указатели CUR всех вспомогательных объектов должны указывать на базовый объект FIFO передачи (чтобы инициализироваться программно).

Флаги TXEN1 всех вспомогательных объектов сообщений, за исключением одного, на который указывает указатель CUR базового объекта, должны быть программно сброшены. Флаг TXEN1 указанного объекта должен быть установлен. Указатель CUR базового объекта может быть инициализирован для любого вспомогательного объекта.

При определении корректности объектов сообщений FIFO структуры для начала FIFO-операций базовый объект должен быть определен первым как корректный, т. е. MSGVAL должен быть установлен.

В случае необходимости удаления FIFO структуры, прежде чем начнется операция удаления, все вспомогательные объекты, принадлежащие этой FIFO структуре, должны быть определены как некорректные (биты MSGVAL должны быть сброшены).

FIFO структура для передачи использует флаги TXEN1 всех своих объектов для выбора сообщения для передачи. В результате фильтрации право передавать сообщение получает тот объект, у которого выставлен флаг TXEN1. После передачи сообщения флаг TXEN1 аппаратно сбрасывается, а в указатель CUR записывается номер следующего объекта, требующего отправки сообщения, для которого уже выставлен (аппаратно) свой флаг TXEN1, и так далее для всей FIFO структуры.

Если установлен флаг OVIE регистра MOFCRn базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения. Это прерывание генерируется на узле прерываний с указателем RXINP базового объекта после завершения операций получения сообщения. Прерывания приема базового объекта генерируются, если это разрешено битом RXIE.

### **Программирование регистров для FIFO структуры**

1 Для передающего базового объекта:

- сбросить бит MSGVAL;

- задать поля CUR, BOT, TOP, SEL;

- записать значение 0010b в поле MMC, задать DLC, установить биты OVIE и RXIE (если необходимо).

Примечание – Состояние регистров MOAR и MOAMR передающего базового объекта не важно, поскольку в передаче участвуют передающие вспомогательные объекты и принимающий базовый объект. Поле RXINP указывает линию, на которую будет выдаваться прерывание переполнения (CUR = SEL).

2 Для передающих вспомогательных объектов:

- сбросить бит MSGVAL;
- установить биты DIR, TXEN1 (только для того вспомогательного объекта, на который указывает поле CUR передающего базового объекта, у остальных вспомогательных объектов бит TXEN1 должен быть сброшен), TXEN0;
- записать в поле CUR номер передающего базового объекта;
- записать значение 0011b в поле MMC, задать DLC.

Примечание – Значение регистров MOAR передающих вспомогательных объектов должно совпадать (или совместно с регистрами MOAMR обеспечивать успешное прохождение фильтрации) со значением регистра MOAR принимающего базового объекта, так как процесс передачи фактически происходит между ними (или иного принимающего объекта, если на приеме используется не FIFO структура).

3 Для принимающего базового объекта:

- установить бит RXEN;
- задать поля CUR, BOT, TOP, SEL;
- записать значение 0001b в поле MMC, задать DLC, установить биты OVIE и TXIE (если необходимо).

Примечание – Значение регистра MOAR принимающего базового объекта должно быть равно значению регистров MOAR передающих вспомогательных объектов передачи (или совместно с регистром MOAMR обеспечивать успешное прохождение фильтрации). Поле TXINP указывает, на какую линию будет выдаваться прерывание переполнения (прерывание после операции сохранения полученного сообщения во вспомогательных объектах при CUR = SEL).

4 Для принимающих вспомогательных объектов:

- сбросить бит RXEN (не требуется, если вспомогательные объекты занесены в список, не связанный с узлом CAN);
- задать поле DLC (состояние поля MMC не важно).

Примечание – Состояние регистров MOAR, принимающих вспомогательные объекты, не важно.

5 Установить бит MSGVAL в первую очередь у передающего базового объекта, а затем у всех остальных объектов.

6 Установить бит TXRQ для всех передающих вспомогательных объектов, начиная с того, на который указывает поле CUR передающего базового объекта.

## 25.10 Режим шлюза

Режим позволяет реализовывать автоматическую передачу информации через шлюз между двумя независимыми шинами CAN без участия ЦП.

Шлюз можно сформировать на уровне объектов сообщений и осуществлять передачу информации между узлами CAN. Шлюз может быть сформирован между двумя любыми объектами сообщений, принадлежащими разным узлам CAN. Количество шлюзов зависит только от количества объектов сообщений, допускающих формирование шлюзов.

Режим шлюза активируется записью значения 0100b в битовое поле MMC регистра MOFCR объекта сообщения Msg\_x, инициализирует его как шлюзовый объект-источник. Объект сообщения, который будет являться шлюзовым объектом-приемником, выбирается указателем CUR объекта-источника. Для формирования шлюза достаточно, чтобы объект-приемник был корректным (установлен бит MSGVAL). Остальные параметры не влияют на возможность осуществления передачи между объектами от источника к приемнику.

Шлюзовый объект-источник, см. рисунок 25.23, функционирует как обычный объект сообщения, с тем отличием, что возможны дополнительные действия контроллера CAN при приеме и сохранении сообщения в объекте-приемнике:

- 1 Если установлен флаг DLCC регистра MOFCRn объекта-источника, код длины данных DLC копируется из шлюзового объекта-источника в шлюзовый объект-приемник.
- 2 Если установлен флаг IDC объекта-источника, идентификатор ID и расширение IDE копируются из шлюзового объекта-источника в шлюзовый объект-приемник.
- 3 Если установлен флаг DATC объекта-источника, байты данных, хранящиеся в двух регистрах MODATAL и MODATAN объекта-источника, копируются из шлюзового объекта-источника в шлюзовый объект-приемник. Копируются все восемь байт данных, вне зависимости от значения поля DLC.

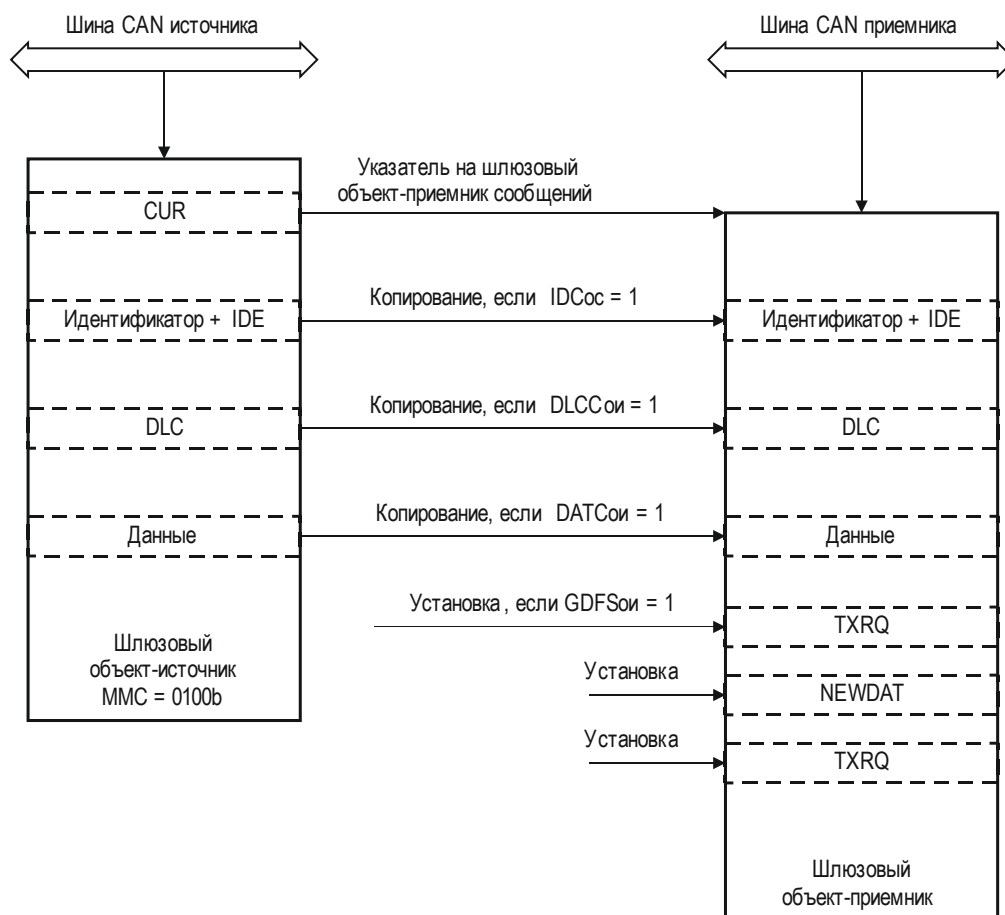


Рисунок 25.23 – Передача через шлюз от источника к приемнику

4 Если установлен флаг GDFS объекта-источника, то устанавливается бит запроса передачи TXRQ объекта-приемника.

5 Устанавливаются флаги RXPND и NEWDAT регистра MOSTAT объекта-приемника.

6 Если установлен флаг RXIE регистра MOSTAT объекта-приемника, то генерируется запрос на прерывание.

7 Указатель CUR объекта-источника переводится на следующий объект-приемник по правилам FIFO структуры. Сформировать шлюз между объектом-источником и одним объектом-приемником (значение указателя CUR будет оставаться неизменным) возможно программированием:

TOP = BOT = CUR = номер объекта-приемника.

Организация шлюза «объект-источник – объект-приемник» аналогична организации FIFO структуры «базовый объект – вспомогательный объект», что указывает на возможность формирования шлюза с интегрированным FIFO-приемником. При получении сообщения данных (объект-источник является объектом приема, т. е. его бит

DIR сброшен) и при получении удаленного запроса (объект-источник является объектом передачи) через шлюз используется один и тот же механизм.

Несмотря на то, что механизм удаленных запросов работает независимо от типа объекта сообщения, он наиболее полезен при использовании шлюзов, для формирования удаленных запросов на шине шлюзового объекта-источника после получения удаленного запроса на шине шлюзового объекта-приемника. В зависимости от значения бита FRREN шлюзового объекта-приемника, есть два варианта обработки удаленного запроса, возникшего с той стороны шлюза, где расположен объект-приемник (при условии, что происходит передача из объекта-источника в объект-приемник, т. е. DIR (источника) = 0 и DIR (приемника) = 1):

1 Обработка запроса шлюзового объекта-приемника с FRREN = 0b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-приемника устанавливается автоматически;
- сообщение данных с текущей информацией, хранящейся в объекте-приемнике, передается на шину приемника.

2 Обработка запроса шлюзового объекта-приемника с FRREN = 1b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-источника (объект должен быть указан в поле CUR объекта-приемника), устанавливается автоматически;
- сообщение данных передается объектом-источником на шину CAN источника;
- получатель удаленного запроса в ответ выдает сообщение данных на шину источника;
- сообщение данных сохраняется в объекте-источнике;
- сообщение данных копируется в объект-приемник (через шлюз);
- выставляется бит TXRQ объекта-приемника (при условии, что GDFS источника = 1);
- новые данные, сохраненные в объекте-приемнике, передаются на шину приемника, в ответ на удаленный запрос на шине приемника.

### **Рекомендации по записи в регистры в режиме шлюза при передаче удаленного запроса с FRREN = 1**

Обмен запрос - данные происходят в данном случае между стандартным объектом сообщения одного узла и объектом-приемником шлюза другого узла. Но при этом данные для ответа на запрос в шлюзовый объект-приемник поступают по шлюзу от объекта-источника. При получении удаленного запроса от объекта сообщения объектом-приемником флаг TXRQ устанавливается не у самого объекта-приемника, а у объекта-источника, благодаря установленному биту FRREN и битовому полю CUR (указывает на объект-источник) объекта-приемника. Данные из MODATAL и MODATAH объекта-источника копируются в MODATAL и MODATAH объекта-приемника (установлен бит DATC регистра MOFCR объекта-источника), вследствие чего автоматически устанавливается бит TXRQ регистра MOCTR объекта-приемника (установлен бит GDFS объекта-источника шлюза), и осуществляется передача сообщения данных (ответ на запрос) запрашивающему объекту сообщения.

После успешного приема/передачи сообщения ЦП получает уведомление о завершении операции для задания дальнейших действий, связанных с объектом сообщения.



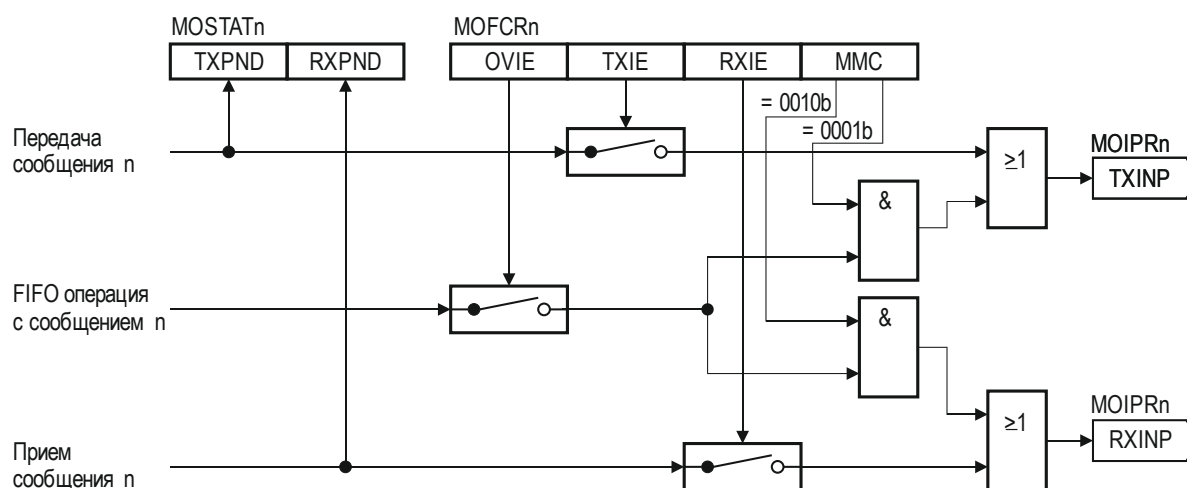
## 25.11 Прерывания объектов сообщений

После сохранения принятого сообщения в объект сообщения или успешной передачи формируется соответствующее прерывание. Каждый объект сообщения может формировать прерывания. Каждое прерывание направляется на одну из 16 выходных линий прерываний. Прерывания приема (после сохранения сообщения) также формируются после операций FIFO и шлюзовых операций. Флаги TXPND и RXPND всегда устанавливаются после успешной операции передачи/приема, независимо от состояния соответствующих флагов разрешения прерываний.

Объект сообщения может формировать FIFO прерывания. Если флаг OVIE регистра MOFCRn установлен, то формирование FIFO прерывания будет зависеть от типа объекта сообщений, см. рисунок 25.24:

- если объект сообщения является принимающим базовым объектом, то выходная линия прерываний для этого объекта определяется битовым полем TXINP регистра MOIPRn;

- если объект сообщения является передающим базовым объектом, то выходная линия прерываний определяется битовым полем RXINP.



MMC = 0001b: объект сообщения n является базовым объектом приема FIFO  
 MMC = 0010b: объект сообщения n является базовым объектом передачи FIFO

Рисунок 25.24 – Распределение прерываний

### Ждущие сообщения

Когда генерируется запрос на прерывание (после приема/передачи сообщения), в одном из восьми регистров ждущих прерываний MSPNDx (x от 0 до 7) выставляется флаг ждущего сообщения. Восемь регистров образуют область из  $32 \times 8$  бит – по два бита (один бит для операций приема и один бит для операций передачи) для каждого из объектов сообщений. Позиция флага ждущего сообщения определяется демультиплексорами DMUX, см. рисунки 25.25 и 25.26.

В зависимости от значения поля MPSEL регистра MCR, реализуется один из двух режимов выбора и установки флагов, ждущих сообщения:

- режим 1 в случае MPSEL = 0h;
- режим 2 в случае MPSEL = Fh.

Если нет необходимости в определении источника прерывания (прием или передача сообщения), то можно использовать любой из двух режимов, в противном случае, следует использовать второй режим.

В первом режиме установка флага ждущего сообщения происходит следующим образом:

- 7, 6 и 5 биты поля MPN выбирают регистр MSPNDx, в котором будет установлен флаг седьмого ждущего сообщения;
- пять младших бит поля MPN (на рисунке 25.25 выделены серым цветом) выбирают позицию флага (от 0 до 31), который будет установлен в выбранном регистре MSPNDx.

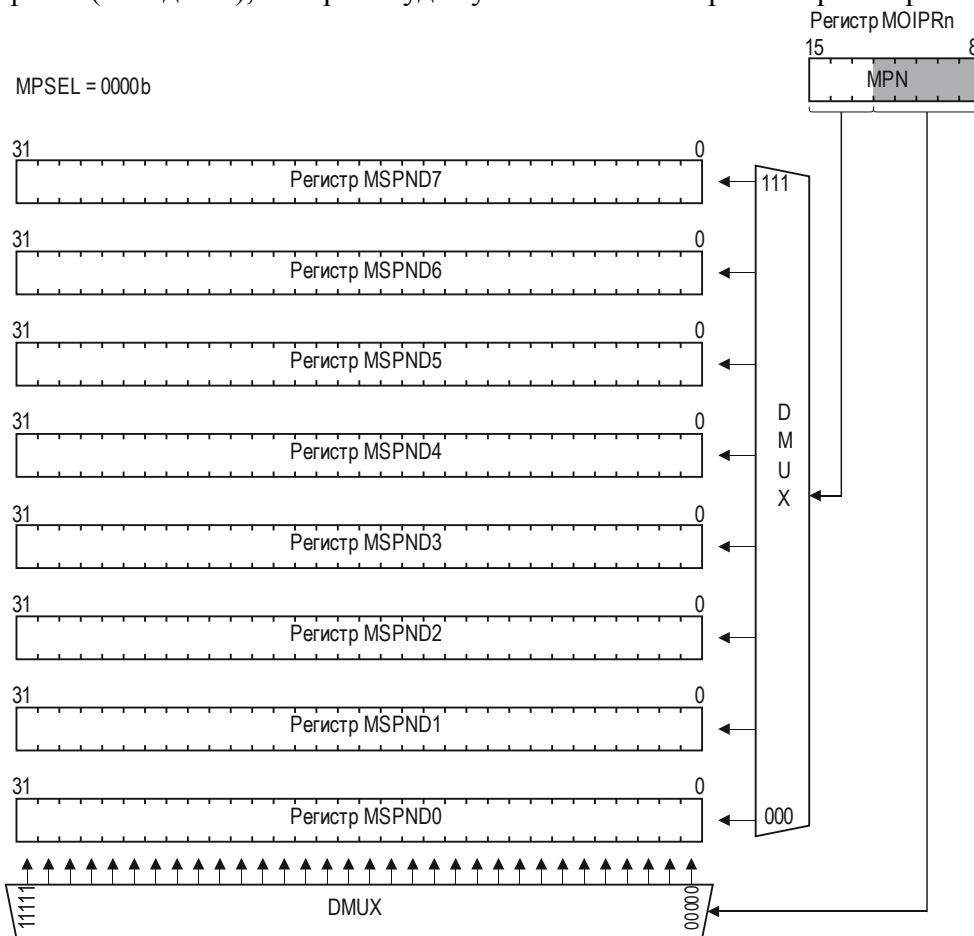


Рисунок 25.25 – Режим выбора и установки флагов при MPSEL = 0h

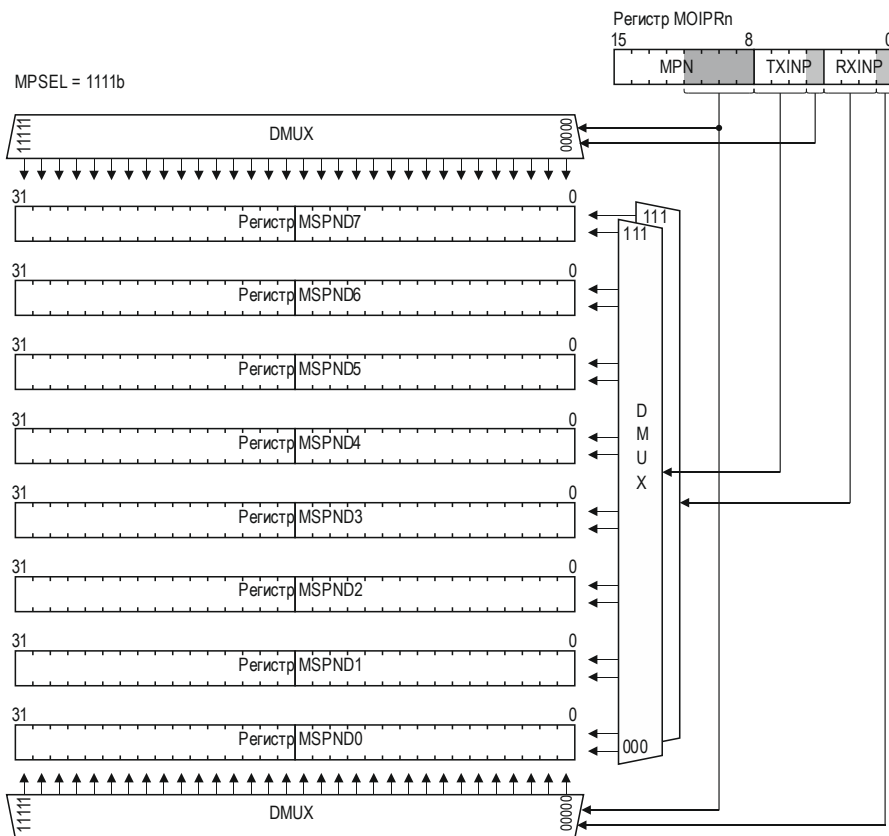


Рисунок 25.26 – Режим выбора и установки флагов при MPSEL = Fh

Во втором режиме при определении позиции флага ждущего сообщения принимаются в расчет значения поля MPN, полей RXINP (для приема) и TXINP (для передачи). При этом для флагов могут использоваться любые биты выбранного регистра MSPNDx. Установка флага ждущего сообщения происходит следующим образом:

- 3, 2 и 1 биты поля TXINP/RXINP выбирают регистр MSPNDx, в котором будет установлен флаг по окончанию передачи/приема сообщения;
- четыре младших бита поля MPN (на рисунке 25.26 выделены серым цветом) совместно с нулевыми битами полей TXINP и RXINP выбирают позицию флага (от 0 до 31). Фактически нулевой бит поля TXINP/RXINP выбирает старшее или младшее слово выбранного регистра MSPNDx, а четыре бита поля MPN задают позицию в выбранном слове.

Регистры MSPNDx могут быть записаны программно. Биты, в которые записываются единицы, остаются без изменений, а биты, в которые записываются нули, очищаются. Такой механизм записи позволяет избежать конфликта между одновременной аппаратной установкой и программной очисткой битов регистра.

Каждый регистр MSPNDx связан с соответствующим регистром индекса сообщения MSIDx, который отражает позицию самого младшего бита из всех установленных в регистре MSPNDx. Регистры MSIDx доступны только для чтения и обновляются незамедлительно после изменения (как аппаратного, так и программного) содержимого соответствующих регистров MSPNDx.

Регистр маски индекса сообщения MSIMASK содержит маску для регистров MSPNDx. Только незакрытые маской биты могут обслуживаться. Регистр MSIMASK используется одновременно для всех регистров MSPNDx и соответствующих им регистров MSIDx.

## 25.12 Программирование контроллера CAN

Для корректной работы контроллера CAN следует соблюдать порядок программирования регистров.

Для запуска контроллера CAN необходимо:

- записать регистр CLC;
- проверить, что сброшен бит DISR, регистр PANCTR = 00000000h и после этого записать регистр FDR.

Далее для конфигурирования узла CAN с номером x (от 0 до 3) выполнить:

- в регистре узла NCRx установить биты INIT и CCE, после чего регистры NBTRx и NPCRx станут доступны для записи и чтения, а регистр NECNTx – только для чтения;

- записать регистр NPCRx;
- записать регистр NIPRx;
- записать регистр NBTRx;
- записать регистр NFCRx (если необходимо);
- в регистре NCRx сбросить биты INIT и CCE, после чего регистры NBTRx и NPCRx будут не доступны для записи;

- распределить объекты сообщений в списки посредством регистра PANCTR.

Для корректной работы объектов сообщений регистры каждого из них должны быть проинициализированы. Для объектов, использование которых не предусматривается, достаточно записать ноль в бит MSGVAL регистра МОСТР.

Рекомендуемый порядок инициализации регистров объекта сообщения:

- установить бит DIR в регистре MOSTAT для передачи сообщения данных/приема удаленного запроса или сбросить бит DIR для приема сообщения данных/передачи удаленного запроса; установить биты TXEN0 и TXEN1 (для передачи) или RXEN (для приема) в регистре МОСТР;

- записать регистр MOFCR;
- записать регистр MOAR;
- записать регистр MOAMR (если необходимо);
- записать регистр MOFGPR (если будут использоваться FIFO структуры);
- записать регистр MOIPR;
- записать регистры MODATAL и MODATAH;
- установить бит MSGVAL корректности объекта сообщения в регистре МОСТР (для неиспользуемых объектов этот бит должен быть сброшен);
- для активирования передачи установить бит TXRQ регистра МОСТР.

## 26 Контроллер интерфейса Ethernet 10/100

Контроллер Ethernet 10/100 реализует стандарт IEEE 802.3. Он осуществляет прием/передачу данных по интерфейсу МП на скорости 10/100 Мбит/с. Прием/передача данных по интерфейсу МП осуществляется в/из буфера блока 32-разрядной памяти объемом 16 Кбайт.

Контроллер Ethernet 10/100 использует 32-разрядный интерфейс для связи с процессором или памятью и осуществляет обмен транслируемыми данными с процессором или памятью через 32-разрядную оперативную память объемом 16 Кбайт. Для накопления и формирования принимаемых и передаваемых пакетов имеются 2 FIFO: для приема – 36-разрядное объемом 4 Кбайт, для передачи – 40-разрядное объемом 2 Кбайт. Обмен данными с процессором или памятью осуществляется на частоте до 50 МГц. Обмен данными с устройством, работающим на физическую линию РНУ: для 100-Мбитного режима на частоте 25 МГц, для 10-Мбитного режима на частоте 2,5 МГц.

Интерфейсный блок контроллера Ethernet 10/100 содержит один контроллер прямого доступа к памяти (ПДП), имеющий два канала, которые используются для операций передачи и приема. Оба канала конкурируют за использование контроллера ПДП, реализуя циклический алгоритм обслуживания конкурирующих запросов.

Типовая передача данных в любом направлении использует кольцевой буфер в пределах назначенной для контроллера Ethernet 10/100 памяти. Кольцевой буфер для передачи определен закрытым связанным списком Тх-дескрипторов. Кольцевой буфер для приема определен закрытым связанным списком Rx-дескрипторов. Два кольцевых буфера формируются из равных 32-разрядных сегментов памяти, способных сохранять пакет максимальной длины. Эти кольцевые буферы должны быть кратными 1-Кбайтной области памяти (максимальный размер пакета данных), и должны располагаться последовательно, не затрагивая области других компонент (область дескрипторов и область кольцевых буферов) контроллера Ethernet 10/100. Предварительно проинициализированный контроллер ПДП может автономно и непрерывно заполнять/освобождать кольцевые буферы. Программное обеспечение может использовать систему прерываний или опрос флагов дескрипторов для поддержания синхронизации потоков данных между контроллером Ethernet 10/100 и процессором или памятью.

На рисунке 26.1 показан пример построения кольцевого буфера из трех дескрипторов. Адрес первого дескриптора задается регистром DMATXDESCR.

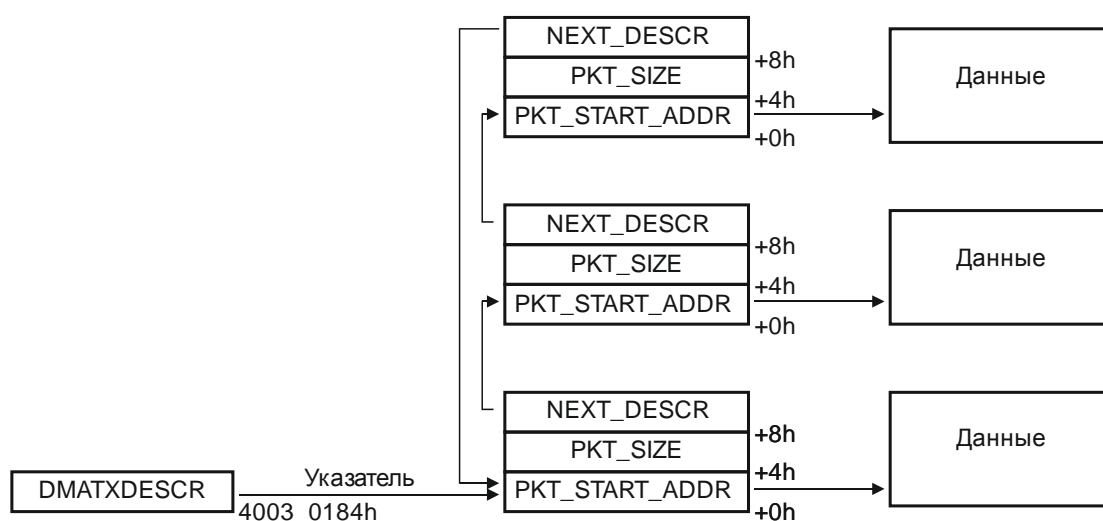


Рисунок 26.1 – Кольцевой буфер с Тх-дескрипторами

Дескриптор формируется тремя последовательно расположенными 32-разрядными ячейками ОЗУ. Назначение ячеек:

- PKT\_START\_ADDR – содержит адрес ячейки ОЗУ, где расположены данные для передачи (пакет данных);
- PKT\_SIZE – содержит размер пакета данных и флаги управления;
- NEXT\_DESCR – содержит адрес следующего дескриптора.

### 26.1 Компоненты дескриптора

Подробная информация о компонентах дескриптора представлена в таблицах ниже.

Таблица 26.1 – Стартовый адрес для пакета данных

<b>PKT_START_ADDR</b>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															-
Поле	Биты	Описание													
ADDR	31-2	Стартовый адрес пакета данных													
-	1-0	Зарезервировано													

Таблица 26.2 – Размер пакета данных

<b>PKT_SIZE</b>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMPTY FLAG	-										FTTP_OVERRIDES				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				SIZE											

Продолжение таблицы 26.2

Поле	Биты	Описание		
EMPTY_FLAG	31	Для операций передачи этот бит указывает на доступность данных передачи, связанных с пакетом. Для операций приема, этот бит показывает наличие места для сохранения принимаемого пакета. Установка этого флага используется для проверки правильности дескриптора. При завершении операции передачи, контроллер DMA устанавливает этот бит, что говорит, что эти данные использованы для передачи. При успешном завершении операции приема, контроллер DMA сбрасывает этот бит, что указывает на то, что отведенное место использовано для сохранения пакета. Первое действие гарантирует, что данные не будут переданы дважды, второе – что сохраненные данные не будут заперчены записью следующего пакета		
FTPP_OVERRIDES	20-16	Флаги управления FIFO во время обмена пакетами		
		Номер бита	Название	Описание
		20	FTCFRM	FIFO передает Control Frame
		19-18	FTPPADMODE	FIFO передает пакет в режиме PADMode
		17	FTPPGENFCS	FIFO передает пакет с генерацией FCS
	16	FTPPEN	Разрешение FIFO на передачу пакета	
SIZE	11-0	Поле для операций передачи. Содержит размер пакета передачи в байтах. При приеме это поле заполняет контроллер DMA. Значение этого поля до приема будет контроллером проигнорировано		
–	30-21, 15-12	Зарезервировано		

Таблица 26.3 – Адрес следующего дескриптора

Поле	Биты	Описание
<b>NEXT_DESCR</b>		
<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> <span>31</span><span>30</span><span>29</span><span>28</span><span>27</span><span>26</span><span>25</span><span>24</span><span>23</span><span>22</span><span>21</span><span>20</span><span>19</span><span>18</span><span>17</span><span>16</span> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;">DESCR_ADDR</div>		
<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> <span>15</span><span>14</span><span>13</span><span>12</span><span>11</span><span>10</span><span>9</span><span>8</span><span>7</span><span>6</span><span>5</span><span>4</span><span>3</span><span>2</span><span>1</span><span>0</span> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;">DESCR_ADDR</div>		
DESCR_ADDR	31-2	Адрес следующего дескриптора
–	1-0	Зарезервировано

## 26.2 Операции передачи

Перед передачей пакета, должна быть записана группа Tx дескрипторов, определяющих кольцевой буфер для операций передачи. Стартовые адреса начала всех сегментов должны быть 32-разрядные, сегменты равные по размерам должны быть достаточны для обработки пакета максимальной длины. Кроме того, в поле SIZE дескриптора передачи должна быть записана длина пакета, а 31 бит EMPTY\_FLAG должен быть установлен, чтобы указать, что кольцевой буфер пока не содержит достоверных данных.

Четыре младших бита регистра DMAINT также должны быть установлены, чтобы специфицировать типы генерируемых прерываний.

После этого процессор должен записать в кольцевые буферы и в дескрипторы, которые связаны с этими сегментами памяти, данные для передачи одного или более пакетов. Затем в поле SIZE записывается длина пакета, и бит EMPTY\_FLAG очищается, что сигнализирует контроллеру ПДП о наличии достоверных данных для передачи. Далее данные в кольцевые буферы разрешено записывать, если установлен бит EMPTY\_FLAG соответствующего дескриптора.

Местоположение точки входа в кольцевой буфер указывается адресом соответствующего дескриптора в регистре DMATXDESCR, см. рисунок 26.1.

Для начала передачи необходимо установить бит TXENABLE в регистре DMATXCTRL. После этого контроллер ПДП прочитает регистр DMATXDESCR и определит адрес стартового дескриптора. Далее:

- если EMPTY\_FLAG установлен, то дескриптор не связан с достоверными данными. В этом случае контроллер ПДП прекратит последовательную передачу пакетов, установит бит TXUNDERRUN в регистре DMASTAT и сбросит бит TXENABLE. Если разрешено прерывание, то оно будет сгенерировано. Для возобновления передачи следует установить бит TXENABLE;

- если EMPTY\_FLAG сброшен, контроллер ПДП прочитает адрес начала буферного сегмента пакета данных и определит его размер.

Далее передача пакета будет стартовать, если FIFO подтвердит контроллеру ПДП о наличии в FIFO достаточного места для приема передаваемого пакета максимальной длины.

Если передача завершена успешно, то контроллер ПДП установит EMPTY\_FLAG, установит флаг TXPKTSENT в регистре DMATXSTAT и, если разрешено, сгенерирует прерывание, и увеличит на единицу число, записанное в поле TXPKTCOUNT. После этого контроллер ПДП перейдет к обработке следующего пакета.

Если произойдет ошибка в канале связи при передаче, то контроллер ПДП прекратит последовательную передачу пакетов, установит бит BUSERROR, сбросит бит TXENABLE и, если разрешено, сгенерирует.

Для последующей передачи потребуется обновление регистра DMATXSTAT для того, чтобы установить новую стартовую позицию в кольцевом буфере, и установить бит TXENABLE.

## 26.3 Операции приема

Перед приемом пакета должна быть записана группа Rx дескрипторов, определяющих кольцевой буфер для операций приема. Стартовые адреса начала всех сегментов должны быть 32-битные, сегменты должны быть одинаковыми по размеру и достаточными для обработки пакета максимальной длины. Поле SIZE дескриптора приема должно быть заполнено, а бит EMPTY\_FLAG установлен, чтобы указать, что кольцевой буфер приема не содержит принятых пакетов.

Биты с седьмого по четвертый регистра DMAINT должны быть установлены, чтобы специфицировать типы генерируемых прерываний.



В регистр DMARXDESCR должен быть записан адрес стартового Rx дескриптора.

Для начала работы следует установить бит RXENABLE в регистре MAC1 для разрешения контроллеру ПДП обрабатывать принимаемый пакет. После этого встроенный контроллер ПДП читает регистр DMARXDESCR для определения адреса первого дескриптора, затем читает этот дескриптор. Далее:

- если EMPTY\_FLAG сброшен, то это означает, что предыдущий принятый пакет еще не был считан программой. В этом случае контроллер ПДП прекращает последовательный прием пакетов, устанавливает бит RXOVERFLOW в регистре DMARXSTAT, сбрасывает бит RXENABLE, и если разрешено прерывание, то оно будет сгенерировано. Любой последующий прием будет возможен только после обновления регистра DMARXDESCR и установки бита RXENABLE;

- если EMPTY\_FLAG установлен, то это означает, что область памяти доступна для сохранения пакета.

Контроллер ПДП начнет прием пакета, если FIFO сообщит контроллеру о наличии принятого пакета. Если прием пакета закончен успешно, то контроллер ПДП запишет значение числа принятых байт в поле PACKET\_SIZE дескриптора приема, сбросит бит EMPTY\_FLAG и установит бит RXPKTREC в регистре DMARXSTAT. Если разрешено, будет сгенерировано прерывание и увеличено на единицу значение в счетчике RXPKTCOUNT.

Если FIFO сообщит, что имеется принятый пакет, то контроллер ПДП начнет передачу этого пакета в следующий сегмент кольцевого буфера.

Программное обеспечение должно обработать прерывание считыванием пакета из кольцевого буфера прима с последующей установкой бита EMPTY\_FLAG дескриптора, помечая этот сегмент кольцевого буфера как доступный для сохранения следующего пакета.

Если произошла ошибка при приеме, то контроллер ПДП прекращает последовательную обработку принимаемых пакетов, устанавливает бит BUSERROR в регистре DMARXSTAT, сбрасывает бит RXENABLE и, если разрешено, генерирует прерывание.

Для следующего приема следует обновить регистр DMARXDESCR, записью правильной стартовой позиции кольцевого буфера, и установить бит RXENABLE.

## 27 Контроллер МПИ (по ГОСТ Р 52070–2003)

Модуль представляет собой устройство, поддерживающее обмен данными с другими устройствами (контроллерами) посредством магистрального последовательного интерфейса (МПИ) в соответствии с ГОСТ Р 52070–2003 (аналогом является стандарт MIL-STD-1553B).

В состав микроконтроллера входят два модуля МПИ: MILSTD $x$ , где  $x$  равен 0 или 1.

На физическом уровне модуль представляет собой последовательную шину данных (экранированная витая пара), к которой подключены устройства. Допустимыми устройствами являются:

- контроллер шины (КШ);
- монитор шины (МШ);
- удаленные терминалы (УТ).

Контроллер шины является ведущим устройством. Он единственный инициирует любой обмен информацией и контролирует работу сети. Контроллер шины может обращаться к любому из удаленных терминалов (максимальное количество 31), каждому из которых присвоен уникальный 5-битный адрес. Монитор шины – пассивное устройство, подключенное к шине данных и занимающееся только отслеживанием и записью передаваемой по шине информации. Возможны следующие способы передачи данных по шине МПИ:

- передача от контроллера шины удаленному терминалу («прием»);
- передача от удаленного терминала к контроллеру шины («передача»);
- передача от одного удаленного терминала к другим удаленным терминалам;
- групповое сообщение от контроллера шины для удаленных терминалов;
- групповое сообщение от одного удаленного терминала для других удаленных терминалов.

Для получения более подробной информации о протоколе следует обратиться к ГОСТ Р 52070–2003.

Каждый модуль МПИ имеет две шины приема-передачи – основная А, и резервная В (рисунок 27.1). Каждый модуль может функционировать в одном из трех режимов:

- контроллер шины;
- удаленный терминал;
- монитор шины.

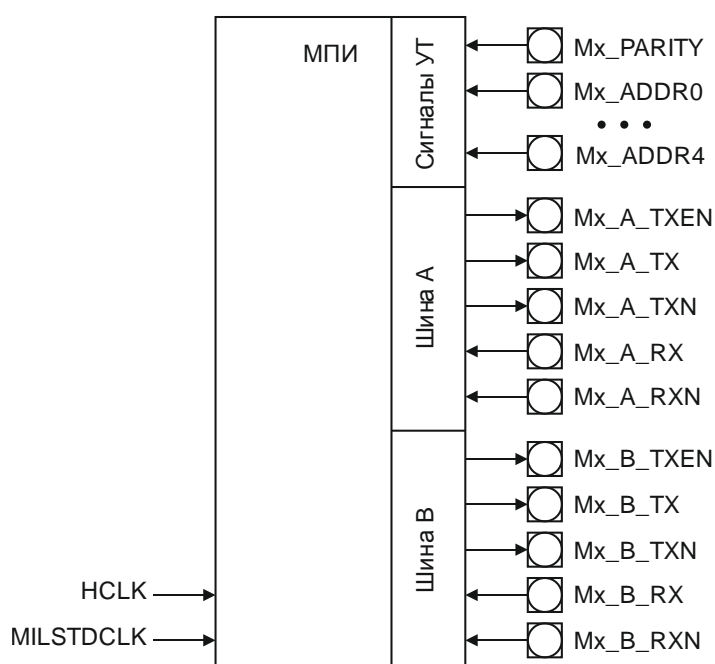


Рисунок 27.1 – Внешние выходы МПИ

Модуль работает в двух тактовых доменах HCLK и MILSTDCLK (шины AMBA и кодека 1553 соответственно), которые синхронизированы между собой. Частота тактового сигнала кодека 1553 должна составлять 20 МГц, а шины AMBA – не менее 10 МГц.

Допустима задержка прохождения сигнала до одного такта сигнала кодека 1553 в обоих тактовых доменах.

У модуля есть два отдельных входа сброса для двух тактовых доменов.

По умолчанию, AMBA домены модулей МПИ не тактируются и находятся в состоянии сброса. Для начала работы нужно подать тактовый сигнал с помощью регистра HCLKCFG, а также вывести блок из состояния сброса, осуществив запись единиц в соответствующие биты регистра HRSTCFG.

Частота тактирования кодека МПИ и вывод его из состояния сброса настраивается с помощью регистров MILSTDCFG0, MILSTDCFG1.

## **27.1 Режимы работы**

МПИ содержит три отдельных блока: контроллер шины, удаленный терминал и монитор шины с общим кодом 1553. Какие именно блоки представлены в контроллере, можно увидеть, прочитав биты BCSUP (регистр статуса и конфигурации КШ BCSTCON), RTSUP (регистр статуса УТ RTSTAT), BMSUP (регистр статуса МШ BMSTAT).

Управление режимами работы контроллера осуществляется с помощью регистров блоков КШ, УТ и МШ. Для начала работы необходимо разрешить работу нужному блоку контроллера интерфейса, изначально все блоки являются неактивными.

Блоки КШ и УТ не могут быть одновременно активными на шине. МШ только отслеживает трафик по шине, его работа не зависит от разрешения работы других блоков интерфейса.

### **Регистры МПИ**

Конфигурация и управление интерфейсом осуществляется с помощью регистров. Для каждого из блоков КШ, УТ, МШ существует отдельный блок управляющих регистров, также есть общие для всех блоков регистры.

Некоторые битовые поля управляющих регистров КШ и УТ защищены «ключом».

Ключ – битовое поле в том же регистре, в которое должно быть внесено конкретное значение, иначе запись не будет осуществлена. Цель использования ключей – гарантия безопасности использования интерфейса, например, при разрешении работы КШ или при изменении адреса УТ.

### **Прерывания**

У МПИ есть единственный вывод для прерывания, однако его источники могут быть различными. Какое именно событие может вызывать прерывание, определяется регистром разрешения прерываний IENR.

### **Кодек 1553**

Внутренний кодек 1553 модуля МПИ принимает и передает данные по шине, генерирует и проверяет синхросигналы, проводит проверку на четность.

Логика контроля методом обратной передачи (loop-back) проверяет, что каждое отправленное слово также было и на приемных входах. Если переданное слово не было получено обратно, то передатчик останавливается и формирует сигнал ошибки, который отражается в результирующем слове.

## **27.2 Контроллер шины (КШ)**

Контроллер шины инициирует любой обмен информацией в сети и контролирует работу сети. Контроллер шины может обращаться к любому из удаленных терминалов по

его адресу, посылать и принимать как одиночные сообщения, так и последовательность сообщений.

Для организации последовательности сообщений составляется список передач. В области памяти программно формируются дескрипторы передач и переходов, буферы принятых данных, данных для передачи, кольцевой буфер указателей прерываний. При запуске последовательности сообщений (установкой бита SCSRT в регистре ВСАСТ) из списка передачи выполняется заданная последовательность передач, также в список передач записываются результирующие слова, принятые данные записываются в соответствующие буферы.

### **Синхронизация**

В каждом дескрипторе передачи можно задавать таймслот STIME. Если время завершения текущей передачи меньше, чем ее таймслот, то модуль приостанавливает работу на оставшееся до следующей передачи время. Это позволяет пользователю точно управлять временем передачи.

Если передача занимает больше времени, чем ее таймслот, то недостающее время вычитается из таймслота следующей команды. Следующая передача также может занимать время из таймслота идущей за ней передачи, и так далее. Модуль может отслеживать вплоть до одной секунды времени, заимствованного из следующих таймслотов, при этом паузы не будут вставляться до тех пор, пока баланс времени остается положительным (кроме пауз между сообщениями, предусмотренных стандартом).

Для максимально быстрого выполнения последовательности передач необходимо установить нулевые значения для всех таймслотов последовательности. Чтобы сгруппировать несколько передаваемых сообщений, необходимо сдвинуть все таймслоты к последней передаче.

Последовательность сообщений может быть остановлена или переведена в состояние ожидания путем записи в регистр ВСАСТ. При работе в состоянии ожидания таймеры последовательности продолжают работать, поэтому при возобновлении передачи сообщений временные интервалы останутся корректными. Если же последовательность сообщений была остановлена, то таймеры будут сброшены.

Если в следующем дескрипторе передачи установлен бит внешней синхронизации, то для начала следующей передачи контроллер интерфейса будет ожидать синхроимпульс, который может быть установлен/сброшен программно записью в биты регистра ВСАСТ. В этом случае таймеры последовательности сбрасываются, таймслот STIME обнуляется, и начинается передача следующего сообщения. Если синхроимпульс приходит до окончания передачи, он захватывается, и передача сообщения начинается без задержки. Захваченный синхроимпульс не учитывается при остановке последовательности передач (но не при переходе в режим ожидания).

### **Выбор шины**

В каждом дескрипторе передачи есть бит выбора шины BUS, с помощью которого можно выбрать шину, по которой будет проводиться передача («0» – выбор шины «А», «1» – выбор шины «В»).

Так же для выбора шины, по которой будет проводиться передача, используется регистр КШ замены шины для УТ ВСBSW, каждый бит которого управляет выбором шины для соответствующего удаленного терминала. Данный регистр является опциональным, проверить его наличие можно прочитав битовое поле ВCFEAT регистра ВСSTCON.

Запись «1» в бит регистра ВСBSW приводит к инвертированию бита выбора шины соответствующего УТ для всех передач («0» – теперь выбор шины В, а «1» – выбор шины

«А»). С помощью этого регистра можно выбрать шину для передачи всех УТ, при этом отсутствует необходимость изменения дескрипторов передачи.

При использовании регистра BCBSW для выбора шины, может быть разрешен бит сохранения шины STBUS в дескрипторе передачи, при этом значение регистра замены шины будет обновляться в зависимости от результата передачи. В случае успешной передачи по шине А соответствующие биты регистра BCBSW устанавливаются в «0», в случае успешной передачи по шине В биты устанавливаются в «1», если передача не прошла, то значение битов меняется на противоположное.

#### **Дополнительный список передач**

Для контроллера интерфейса может быть составлен дополнительный «асинхронный» список передач того же формата, что и основной список передач. В то время, пока выполняется основной список передач, может быть запущено и выполнение дополнительного списка передач. После завершения передачи из основного списка, происходит сравнение ее таймслота с таймслотом передачи из асинхронного списка. Если таймслот передачи из асинхронного списка меньше времени ожидания следующей передачи, то выполняется передача из асинхронного списка.

Если передача из асинхронного списка не будет завершена вовремя, то ее таймслот будет увеличен за счет таймслота следующей передачи из основного списка. Таким образом, чтобы не нарушить временные соотношения для передач из основного списка, необходимо устанавливать таймслоты для команд из асинхронного списка с учетом наихудших вариантов.

Для запрещения выполнения передач из асинхронного списка необходимо установить бит EXCL в дескрипторе передач.

Сообщения из асинхронного списка передач не могут быть запущены в случае ожидания синхроимпульса или в режиме ожидания при истечении текущего таймслота, поскольку не определен момент времени для начала следующей передачи.

#### **Генерация прерываний**

Прерывание может быть сгенерировано после передачи каждого сообщения из списка (переданного с ошибкой или без ошибки). Некорректное сообщение всегда вызывает прерывание и останавливает выполнение последовательности передач. Перед тем, как будет сгенерировано прерывание по передаче, адрес соответствующего дескриптора передачи записывается в кольцевой буфер КШ обработки прерываний по передаче, при этом значение регистра BCTRP инкрементируется.

Прерывание по отдельной ошибке свидетельствует о наличии ошибок DMA. Если ошибка DMA произошла при чтении или записи дескриптора, то исполняемая последовательность передач будет переведена в режим ожидания. Ошибки DMA в буферах данных приводят к ошибке в соответствующем сообщении, при этом формируется код ошибки TFRST в результирующем слове (описание кодов ошибок приведено в таблице 27.4).

Управление прерываниями осуществляется с помощью регистра масок прерываний.

#### **Формат списка передач**

Список передач КШ состоит из массива дескрипторов передачи и переходов, как показано в таблице 27.1. Каждая запись массива должна быть выровнена по 128-битной границе (16 байт). Два незадействованных слова из массива переходов могут использоваться для хранения служебных данных.

Таблица 27.1 – Формат дескрипторов передачи

Смещение	Формат дескриптора передачи	DMA, з/ч	Формат перехода	DMA, з/ч
00h	Слово 0 дескриптора передачи (таблица 27.2)	ч	Условие перехода (таблица 27.6)	ч
04h	Слово 1 дескриптора передачи (таблица 27.3)	ч	Адрес перехода, выравнивание 128 бит	ч
08h	Указатель буфера данных, выравнивание 16 бит. Для буферов записи, если установлен бит 0, то принятые данные не используются, и указатель игнорируется. Это может быть использовано в режиме передачи от УТ к УТ, когда КИШ не использует передаваемые данные.	ч	Не используется	-
0Ch	Результирующее слово, записывается аппаратно (таблица 27.4)	з	Не используется	-

Таблица 27.2 – Слово 0 дескриптора передачи КИШ

Поле	Биты	Описание	
-	31	Должен содержать нулевое значение для определения в качестве дескриптора	
WTRIG	30	Бит ожидания внешнего синхроимпульса	
EXCL	29	Запрещение выполнения передач из асинхронного списка	
IRQE	28	Прерывание по ошибке после завершения передачи	
IRQN	27	Стандартное прерывание после каждой передачи	
SUSE	26	Перевод последовательности передач в режим ожидания (или остановка выполнения передач из асинхронного списка) при ошибке	
SUSN	25	Перевод последовательности передач в режим ожидания после каждой передачи	
RETMD	24-23	Режим повторов	
		00	Повтор передач только по той же самой шине
		01	Повтор передач чередуется по обеим шинам
		10	Повтор передач сначала по той же самой шине, затем по другой
		11	Зарезервировано, не использовать

Продолжение таблицы 27.2

Поле	Биты	Описание
NRET	22-20	Количество автоматических повторов передач для шины (включая первую попытку передачи). Для режима повтора RETMD = 00b количество повторов равно (NRET + 1). Для режима повтора RETMD = 01b и RETMD = 10b количество повторов равно 2 * (NRET + 1)
STBUS	19	Бит сохранения шины. Если этот бит установлен, и передача по шине прошла успешно, то передача продолжается по той же шине («0» для шины А, «1» для шины В в регистре BCBSW). Если передача по шине прошла некорректно, то выбирается альтернативная шина.
GAP	18	Бит увеличения паузы между сообщениями. При установке данного бита после передачи добавляется дополнительное время к паузе между сообщениями (в соответствии с битовым полем RTTO слова 1 дескриптора передачи)
STIME	15-0	Таймслот. Выделенное для передачи время, единицей измерения являются 4 мкс. При досрочном окончании передачи вставляется задержка.
-	17-16	Зарезервировано

Таблица 27.3 – Слово 1 дескриптора передачи КИШ

Поле	Биты	Описание																																																																											
Слово 1 дескриптора передачи																																																																													
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> </tr> <tr> <td>DUM</td><td>BUS</td><td colspan="4">RTTO</td><td colspan="4">RTAD2</td><td colspan="4">RTS2</td> </tr> <tr> <td colspan="16"> </td> </tr> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="4">RTAD1</td><td>TR</td><td colspan="4">RTSA1</td><td colspan="4">WCMC</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	DUM	BUS	RTTO				RTAD2				RTS2																				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RTAD1				TR	RTSA1				WCMC			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																																																														
DUM	BUS	RTTO				RTAD2				RTS2																																																																			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																														
RTAD1				TR	RTSA1				WCMC																																																																				
DUM	31	Ложная передача. Если установлен этот бит, то сигналы на шине не генерируются, а передача считается успешно завершённой. При такой передаче возможно использование битовых полей EXCL, IRQN, SUSN, STBUS, GAP, STIME, другие битовые поля и указатель буфера данных игнорируются.																																																																											
BUS	30	Бит выбора шины 0 Для передачи выбирается шина А 1 Для передачи выбирается шина В																																																																											
RTTO	29-26	Задержка УТ. Дополнительная задержка ответного слова УТ (сверх стандартного значения), единицей измерения являются 4 мкс (0000b – 14 мкс, 1111b – 74 мкс).																																																																											
RTAD2	25-21	Адрес второго УТ (в режиме передачи от одного УТ к другому УТ)																																																																											
RTSA2	20-16	Подадрес второго УТ (в режиме передачи от одного УТ к другому УТ)																																																																											

Продолжение таблицы 27.3

Поле	Биты	Описание	
RTAD1	15-11	Адрес УТ. В поле записывается адрес УТ, к которому обращается КШ. Значение 1Fh является групповым адресом и служит для обращения ко всем удаленным терминалам одновременно.	
TR	10	Бит направления передачи	
		0 Прием	
		1 Передача	
RTSA1	9-5	Подадрес/Управление.	
		00h или 1Fh	Режим «Управление». Если в поле записано 00h или 1Fh , то значение в поле WCMC является кодом команды управления
		01h–1Eh	Режим «Подадрес». В поле находится адрес подчиненного устройства (абонента), подключенного непосредственно к выбранному удаленному терминалу. В этом случае в поле WCMC указывается количество слов данных для передачи/приема
WCMC	4-0	Количество данных/Код команды управления. Режим работы битового поля устанавливается полем RTSA1	
<p>Примечание – Задержка УТ RTTO также может быть использована в качестве дополнительной паузы между сообщениями (если установлен бит GAP в слове 0 дескриптора передачи).</p> <p>Примечание – Биты 15-0 соответствуют командному слову в стандарте MIL-STD-1553B. Конфигурация битовых полей RTAD1, RTSA1, RTAD2, RTSA2, WCMC, TR для различных режимов работы показана в таблице 27.5.</p>			

Таблица 27.4 – Результирующее слово дескриптора передачи

Поле	Биты	Описание
Результирующее слово		
<p>The diagram shows a 32-bit word with bit positions 31 down to 0. Bit 31 is labeled '0'. Bits 23-16 are grouped as 'RT2ST'. Bits 15-8 are grouped as 'RTST'. Bits 7-6 are grouped as 'RETCNT'. Bit 3 is labeled '-'. Bits 2-1 are grouped as 'TFRST'.</p>		
-	31	Всегда записан «0»
RT2ST	23-16	Статус второго УТ (УТ-приемника в режиме передачи от одного УТ к другому УТ). В другом режиме работы поле содержит нулевые значения. Статус второго УТ определяется аналогично статусу первого УТ, в соответствии с полем RTST.



Продолжение таблицы 27.4

Поле	Биты	Описание
RTST	15-8	Статус УТ (УТ-передатчика в режиме передачи от одного УТ к другому УТ)
		бит 15   Ошибка в сообщении (ME)
		бит 14   Бит распознавания сообщений, «0» для ответного слова (INSTR)
		бит 13   Запрос на обслуживание (SREQ)
		бит 12   Принято групповое сообщение (BCCMD)
		бит 11   Бит занятости абонента (BUSY)
		бит 10   Неисправность абонента(SSF)
		бит 9   Бит подтверждения принятия управлением шины (DBCA)
		бит 8   Неисправность УТ (TFLG)
RETCNT	7-4	Количество выполненных повторов передач
TFRST	2-0	Статус передачи (определяется по последней выполненной передаче)
		000   Передача прошла успешно (или был установлен бит DUM в слове 1 дескриптора передачи)
		001   Нет ответа от УТ (УТ-передатчика в режиме передачи от одного УТ к другому УТ)
		010   Нет ответа от второго УТ (УТ-приемника в режиме передачи от одного УТ к другому УТ)
		011   В ответном слове УТ установлены биты ошибок (ME, BUSY, установлен бит распознавания сообщений INSTR)
		100   Ошибка протокола (ошибка декодера, неверное число слов данных, ошибочные временные соотношения для слов данных)
		101   Неверный формат дескриптора передачи
		110   Ошибка данных буфера DMA
		111   Передача прервана в результате ошибки обратной передачи (loop-back)
-	30-24, 3	Зарезервировано
<p>Примечание – Код ошибки 011b возникает только в случае соответствия числа переданных данных числу принятых, в противном случае используется код 100b.</p> <p>Код ошибки 011b может возникнуть при правильном выполнении команд управления «передать последнюю команду», «передать последнее ответное слово», так как после выполнения этих команд ответное слово не очищается.</p>		

### Инициализация КШ

1 Составить список передач для КШ (массив из дескрипторов передач, переходов, выравнивание каждой записи по 16 байт).

2 Записать в регистр BCLNP начальный адрес списка передач.

3 Для разрешения прерывания необходимо установить соответствующие биты в регистре IENR, а также установить соответствующие биты в Слове 0 дескриптора передачи (IRQN IRQE) или бит IRQC Условия перехода.

4 Для начала передачи: в регистр BCAST записать ключ в поле BCKEY, установить бит SCSRT.

### Форматы сообщений

Форматы делятся на две группы – форматы основных сообщений и форматы групповых сообщений (по ГОСТ Р 52070–2003). Форматы групповых сообщений

применяются для передачи информации, предназначенной нескольким удаленным терминалам без получения от них ответных слов. Сообщение является групповым, если в нем адрес удаленного терминала равен 1Fh. Каждый удаленный терминал, который может принять команду общего вызова, после ее обнаружения устанавливает соответствующий флаг в ответном слове, но само слово не передает.

Сообщение считается командой управления (КУ), если в поле RTSA1 слова 1 дескриптора передачи содержатся коды 0h или 1Fh. В поле WCMC слова 1 дескриптора передачи содержится код команды. Команды управления применяются только для управления удаленными терминалами (не для обмена данными). Наличие команд управления позволяет контроллеру шины не только контролировать работу шины, но и обрабатывать, и исправлять обнаруженные ошибки.

В таблице 27.5 представлены 10 допустимых форматов сообщений, с указанием необходимой конфигурации битовых полей слова 1 дескриптора передачи.

Таблица 27.5 – Конфигурация битовых полей слова 1 дескриптора передачи для разных форматов сообщений

Формат передачи	Значение управляющих битовых полей		Направление передачи данных
Формат 1 – Передача данных от КШ к УТ	RTAD1	Адрес УТ (0-30)	Чтение (2-64 байта)
	RTSA1	Подадрес УТ (1-30)	
	RTAD2	Не влияет	
	RTSA2	0	
	WCMC	Количество слов данных (0-32)	
	TR	0	
Формат 2 – Передача данных от УТ к КШ	RTAD1	Адрес УТ (0-30)	Запись (2-64 байта)
	RTSA1	Подадрес УТ (1-30)	
	RTAD2	Не влияет	
	RTSA2	0	
	WCMC	Количество слов данных (0-32)	
	TR	1	
Формат 3 – Передача данных от УТ к УТ	RTAD1	Адрес УТ-приемника (0-30)	Запись (2-64 байта)
	RTSA1	Подадрес УТ-приемника (1-30)	
	RTAD2	Адрес УТ-передатчика (0-30)	
	RTSA2	Подадрес УТ-передатчика (1-30)	
	WCMC	Количество слов данных (0-32)	
	TR	0	
Формат 4 – Передача КУ от КШ к УТ	RTAD1	Адрес УТ (0-30)	-
	RTSA1	0h или 1Fh	
	RTAD2	Не влияет	
	RTSA2	Не влияет	
	WCMC	Код КУ (0h-8h)	
	TR	1	
Формат 5 – Передача КУ от КШ к УТ и получение от него одного слова данных	RTAD1	Адрес УТ (0-30)	Запись (2 байта)
	RTSA1	0h или 1Fh	
	RTAD2	Не влияет	
	RTSA2	Не влияет	
	WCMC	Код КУ (10h, 12h, 13h)	
	TR	1	

Продолжение таблицы 27.5

Формат 6 – Передача КУ и одного слова данных от КШ к УТ	RTAD1	Адрес УТ (0-30)	Чтение (2 байта)
	RTSA1	0h или 1Fh	
	RTAD2	Не влияет	
	RTSA2	Не влияет	
	WCMC	Код КУ (11h, 14h, 15h)	
	TR	0	
Формат 7 – Передача данных в групповом сообщении от КШ к УТ	RTAD1	1Fh	Чтение (2-64 байта)
	RTSA1	Подадрес УТ (1-30)	
	RTAD2	Не влияет	
	RTSA2	0	
	WCMC	Количество слов данных (0-32)	
	TR	0	
Формат 8 – Передача данных в групповом сообщении от УТ к УТ	RTAD1	1Fh	Запись (2-64 байта)
	RTSA1	Подадрес УТ-приемника (1-30)	
	RTAD2	Адрес УТ-передатчика (0-30)	
	RTSA2	Подадрес УТ-передатчика (1-30)	
	WCMC	Количество слов данных (0-32)	
	TR	0	
Формат 9 – Передача групповой КУ от КШ к УТ	RTAD1	1Fh	-
	RTSA1	0h или 1Fh	
	RTAD2	Не влияет	
	RTSA2	Не влияет	
	WCMC	Код КУ (1h, 3h-8h)	
	TR	1	
Формат 10 – Передача групповой КУ и одного слова данных от КШ к УТ	RTAD1	1Fh	Чтение (2 байта)
	RTSA1	0h или 1Fh	
	RTAD2	Не влияет	
	RTSA2	Не влияет	
	WCMC	Код КУ (11h, 14h, 15h)	
	TR	0	

Формирование слова условия перехода происходит в соответствии с таблицей 27.6.

Таблица 27.6 – Слово условия перехода

Слово условия перехода		
Поле	Биты	Описание
-	31	Бит является идентификатором условия перехода, должен быть установлен в «1»
-	30-27	Зарезервировано, необходимо записывать нулевые значения

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	0			IRQC	ACT	MO DE	RT2CC								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCC								STCC							

Продолжение таблицы 27.6

IRQC	26	Прерывание, если выполнено условие перехода
ACT	25	Управление переходом при выполнении условия
		0   Перевод последовательности передач в режим ожидания 1   Осуществление перехода
MODE	24	Логический режим работы
		0   Режим «ИЛИ» (любой из битов, установленных в битовых полях RT2CC, RTCC, устанавливается и в полях RT2ST, RTST), результат выполнения операции «ИЛИ» – в поле маски STCC 1   Режим «И» (все биты, установленные в RT2CC, RTCC, устанавливаются в RT2ST, RTST), результат выполнения операции «И» – в поле маски STCC.
RT2CC	23-16	Код условия перехода для второго УТ. Битовое поле является маской для соответствующих битов RT2ST в результирующем слове (для последней передачи)
RTCC	15-8	Код условия перехода для УТ. Битовое поле является маской для соответствующих битов RTST в результирующем слове (для последней передачи)
STCC	7-0	Статус условия перехода. Маска для статусных битов последней передачи
<p>Примечание – Чтобы получить всегда срабатывающее условие перехода, необходимо установить поле MODE в значение «0», а STCC = FFh (STCC = 00h – чтобы условие никогда не срабатывало). Значение 8000_00FFh при этом может быть использовано как указатель на окончание списка передач.</p>		

### 27.3 Удаленный терминал (УТ)

Удаленный терминал выполняет команды контроллера шины. Каждому удаленному терминалу присваивается адрес в диапазоне от 00h до 1Eh. УТ считывает команды, адресованные собственному адресу УТ (или групповые команды) и проверяет разрешения использования команд. Если команда разрешена, то осуществляется ее выполнение, иначе в ответном слове выставляется флаг ME (бит 10 ответного слова). Разрешение для команды на прием/передачу данных проверяется по Управляющему слову таблицы подадресов УТ, а для команды управления (КУ) – по регистру УТ управления кодами команд RTMOD.

Для контроля работы удаленного терминала со стороны контроллера шины используется ответное слово, которое отправляется контроллеру в конце передачи каждого сообщения или по прямому запросу. Ответное слово не передается после приема групповой передачи и в случае обнаружения ошибки в принятых данных.

Таблица 27.7 – Формат и назначение битов ответного слова

Ответное слово															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTADDR					ME	INS TR	SREQ	-			BC CMD	BUSY	SSF	DBCA	TFLG
Поле	Биты	Описание													
RTADDR	15-11	Собственный адрес УТ													
ME	10	Ошибка в сообщении													
INSTR	9	Бит распознавания сообщений													
SREQ	8	Запрос на обслуживание													
BCCMD	4	Получена групповая команда													
BUSY	3	Абонент занят													
SSF	2	Флаг подсистемы													
DBCA	1	Бит подтверждения принятия управления шиной													
TFLG	0	Флаг УТ													
-	7-5	Зарезервировано													

### Передача данных

УТ использует трехуровневую структуру для передачи данных. Первый уровень – таблица подадресов, где для каждого подадреса задается соответствующее слово управления подадресом, указатели для дескрипторов передачи и дескрипторов приема. Каждый дескриптор в свою очередь включает в себя дескриптор управляющего/статусного слова, указатель буфера данных, указатель на следующий дескриптор, формируя тем самым список или кольцо дескрипторов. Буферы данных могут находиться в любом месте памяти, выравнивание адреса должно составлять 16 бит.

Команда на прием/передачу данных, поступившая к УТ, сверяется с таблицей подадресов. Если команда является корректной, то происходит передача из соответствующего буфера или прием в соответствующий буфер DMA. После передачи данных дескрипторы управляющего/статусного слова обновляются в соответствии с результатом окончания передачи, указатель таблицы подадресов меняется на указатель на следующий дескриптор.

Если разрешено логирование происходящих событий, то соответствующая запись о событии будет помещена в область кольцевого буфера сбора логов. Также может быть разрешено прерывание по передаче. Для определения того, какая передача вызвала прерывание, может использоваться регистр позиции прерывания в логе событий УТ (RTELIP). Поэтому для разрешения прерываний должно быть разрешено логирование.

Если УТ получена корректная команда, но она не может быть выполнена (нет соответствующего дескриптора или данные недоступны в отведенное для отклика время), то формируется сигнал прерывания по ошибке доступа к таблице УТ, удаленный терминал не отвечает на команду КШ. При такой ошибке может автоматически устанавливаться статусный бит «Флаг УТ» (TFLAG).

### Коды команд управления

С помощью регистра УТ управления кодами команд (RTMOD) устанавливается, какая именно команда управления является разрешенной, какая вызывает прерывание, разрешено ли логирование. Запрещающие КУ контролируются теми же самыми битовыми полями. Для команд управления, которые могут быть использованы в групповых сообщениях, предусмотрено два битовых поля – для управления командой в групповом и в стандартном режиме.

В таблице 27.8 приведены коды команд управления и действия, вызываемые ими. Некоторые команды не вызывают никаких действий, их нужно реализовывать программно.

Таблица 27.8 – Коды команд управления УТ

Код команды	Описание команды управления. Действие, вызываемое командой	Управляющие биты в регистре RTMOD	Возможность применения	
			Логирование, вызов прерывания	Разрешение после сброса
00h	Принять управление шиной. Если в регистре статуса шины УТ RTBST установлен бит DBCA, то формируется отклик «Подтверждение принятия управления шиной».	17-16	+	-
01h	Синхронизация. Обновляется битовое поле SYTM в регистре синхронизации УТ RTSYNC.	3-0	+	+
02h	Передать ответное слово. УТ передает ответное слово. Команда все время разрешена. Нельзя вести логирование или запрещать команду.	-	-	+
03h	Начать самоконтроль УТ. Нет предопределенных действий для команды.	21-18	+	-
04h	Блокировать передатчик. УТ перестает отвечать на команды по другой шине (но не по той, по которой получил данную команду управления)	11-8	+	+
05h	Разблокировать передатчик. Отменяет действие предыдущей команды «Блокировать передатчик», принятой раньше по этой же шине.	11-8	+	+
06h	Маскировать флаг УТ (TFLAG). Маскирует флаг УТ в ответных словах.	25-22	+	-
07h	Отмена маскирования флага УТ (TFLAG). Отменяет действие предыдущей команды «Маскировать флаг УТ»	25-22	+	-
08h	Установить УТ в исходное состояние. Сброс таймеров, отмена действий команд «Блокировать передатчик», «Блокировать признак неисправности УТ». В регистре статуса шины УТ RTBST сбрасываются биты TFLG и SREQ.	29-26	+	-

Продолжение таблицы 27.8

Код команды	Описание команды управления. Действие, вызываемое командой	Управляющие биты в регистре RTMOD	Возможность применения	
			Логирование, вызов прерывания	Разрешение после сброса
09h-0Fh	Зарезервировано, не использовать	-	-	-
10h	Передать векторное слово. Передача векторного слова из статусного регистра слов УТ RTSW.	13-12	+	-
11h	Синхронизация со словом данных. Обновляются битовые поля SYTM и SYD в регистре синхронизации УТ RTSYNC.	7-4	+	+
12h	Передать последнюю команду. Передается последняя команда, принятая УТ. Команда все время разрешена. Нельзя вести логирование или запрещать команду.	-	-	+
13h	Передать слово ВIT. Передача слова ВIT из статусного регистра слов УТ RTSW.	15-14	+	-
14h	Блокировать выбранный передатчик. Нет предопределенных действий для команды.	-	-	-
15h	Разблокировать выбранный передатчик. Нет предопределенных действий для команды.	-	-	-
16h-1Fh	Зарезервировано, не использовать	-	-	-

#### Логирование событий

Логи представляют из себя 32-битные записи определенного формата. Формат записи показан в таблице 27.9. При передаче данных биты 23-0 лога аналогичны битам 23-0 Дескриптора управляющего и статусного слова УТ.

Таблица 27.9 – Формат лога событий УТ

Формат лога событий УТ															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ SR	TYPE		SAMC				TIMEL								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEL						BC	SZ					TRES			
Поле	Биты	Описание													
IRQSR	31	Источник прерывания. Устанавливается в «1», если передача вызвала прерывание													
TYPE	30-29	Тип передачи													
		00	Передача данных												
		01	Прием данных												
		10	Команда управления												
		11	-												
SAMC	28-24	Подадрес/Код команды управления. Если TYPE = 00b или TYPE = 01b, то в поле SAMC содержится подадрес, если TYPE = 10b, то в поле SAMC содержится код команды управления.													
TIMEL	23-10	Младшие 14 бит таймера													
BC	9	Групповое сообщение.													
SZ	8-3	Количество принятых или переданных 16-битных слов данных (0-32).													
TRES	2-0	Результат передачи													
		000	Передача прошла успешно												
		001	Замена передачи (отмена текущей передачи, так как была получена новая команда по другой шине)												
		010	Ошибка DMA или превышение времени доступа к памяти.												
		011	Ошибка протокола (ошибка временного интервала для слов данных или ошибка декодера).												
		100	В полученном ответном слове установлены биты BUSY или ME и не были отправлены данные.												
		101	Передача прервана из-за ошибки контроля обратной связи.												
		011- 111	-												

### Формат таблицы подадресов

Таблица 27.10 – Запись таблицы подадресов для числа подадресов N

Смещение	Описание	Доступ DMA
10h * N + 00h	Управляющее слово для N-го подадреса (таблица 27.11).	Чтение
10h * N + 04h	Указатель дескриптора передачи, выравнивание 16 байт (3h – некорректный указатель).	Чтение/Запись
10h * N + 08h	Указатель дескриптора приема, выравнивание 16 байт (3h – некорректный указатель).	Чтение/Запись
10h * N + 0Ch	Не используется	-
Примечание – Общее количество подадресов равно N, где 0 < N < 31. Для подадресов 0h и 1Fh не используется таблица подадресов.		



Таблица 27.11 – Управляющее слово таблицы подадресов УТ

Управляющее слово таблицы подадресов УТ															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												WR AP	IGN DV	BC RXE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXEN	RX LOG	RX IRQ	RXSZ				TXEN	TX LOG	TX IRQ	TXSZ					
Поле	Биты	Описание													
-	31-19	Зарезервировано. Для совместимости необходимо записывать нулевые значения.													
WRAP	18	Разрешение циклического возврата данных. Разрешение тестового режима для данного подадреса: приемник отсылает обратно принятые данные. Это достигается копированием указателя дескриптора завершенной передачи в указатель дескриптора передачи после каждой успешной передачи.													
IGNDV	17	Бит игнорирования корректности данных. Если установлен бит IGNDV, то поступающие данные будут приняты (и перезапишут буфер), даже если бит DV = 1 в дескрипторе приема (вместо того, чтобы не отвечать на полученные сообщения)													
BCRXEN	16	Разрешение приема групповых сообщений по данному подадресу.													
RXEN	15	Разрешение приема для данного подадреса.													
RXLOG	14	Логирование всех операций приема данных (используется только если RXEN = 1).													
RXIRQ	13	Прерывание по приему данных – каждая операция приема данных будет вызывать прерывание (только если установлены биты RXEN, RXLOG).													
RXSZ	12-8	Максимальное допустимое для этого подадреса количество 16-битных слов данных для приема (00h – 32 слова данных).													
TXEN	7	Разрешение передачи для данного подадреса.													
TXLOG	6	Логирование всех операций передачи данных (используется только если TXEN = 1).													
TXIRQ	5	Прерывание по передаче данных – каждая операция передачи данных будет вызывать прерывание (только если установлены биты TXEN, TXLOG).													
TXSZ	4-0	Максимальное допустимое для этого подадреса количество 16-битных слов данных для передачи (00h – 32 слова данных).													
Примечание – если поле WRAP = 1, то нельзя устанавливать значение TXSZ большим, чем RXSZ, так как это может привести к чтению данных за границами буфера.															

Таблица 27.12 – Формат дескрипторов УТ

Смещение	Описание	Доступ DMA
00h	Управляющее и статусное слово (таблица 27.13).	Запись/Чтение
04h	Указатель буфера данных, выравнивание 16 бит.	Чтение
08h	Указатель на следующий дескриптор, выравнивание 16 бит (Значение 0000_0003h при этом может быть использовано как указатель на окончание списка)	Чтение

Таблица 27.13 – Дескриптор управляющего и статусного слова УТ

Дескриптор управляющего/статусного слова			
Поле	Биты	Описание	
DV	31	Бит корректности данных. Должен быть установлен в «0» программно до начала передачи и в «1» после передачи. Если бит DV = 1 в текущем дескрипторе передачи еще до начала передачи, считается, что произошла ошибка дескриптора таблицы. Чтобы такая ошибка не учитывалась, необходимо установить бит IGNDV в таблице подадресов.	
IRQEN	30	Переопределение разрешения прерывания. Логирование и разрешение прерывания, независимо от значений битов разрешения прерываний в управляющем слове таблицы подадресов.	
-	29-26	Зарезервировано. Для совместимости необходимо устанавливать в «0» и маскировать при чтении.	
TTIME	25-10	Время передачи. После окончания передачи в поле TTIME записывается значение таймера УТ.	
BC	9	Групповое сообщение. Устанавливается аппаратно, если завершившееся сообщение было групповым.	
SZ	8-3	Количество переданных 16-битных слов данных (0-32)	
TRES	2-0	Результат передачи	
		000	Передача прошла успешно
		001	Замена передачи (отмена текущей передачи, так как была получена новая команда по другой шине)
		010	Ошибка DMA или превышение времени доступа к памяти.
		011	Ошибка протокола (ошибка временного интервала для слов данных или ошибка декодера).
		100	В полученном ответном слове установлены биты BUSY или ME и не были отправлены данные.
		101	Передача прервана из-за ошибки контроля обратной связи.
011-111	-		

### Инициализация УТ

- 1 Инициализировать таблицу подадресов;
- 2 В регистр основного адреса таблицы подадресов УТ RTSADDR записать начальный адрес таблицы подадресов;
- 3 Записать в регистр конфигурации УТ: ключ RTKEY, адрес данного УТ, установить бит RTEN.

## 27.4 Монитор шины (МШ)

Монитор шины отслеживает и записывает информацию, передаваемую по шинам данных, а также сохраняет результаты мониторинга в кольцевом буфере. Монитор шины может работать как отдельно, так и совместно с КШ или с УТ.

### Фильтрация

Фильтрация – одна из функций МШ, управляемая регистрами BMADF, BMSADF, BVMODF.

Передачи могут фильтроваться по адресу УТ, подадресу или коду команды управления. Условия фильтрации логически умножаются. Если все биты трех регистров фильтрации и биты UDWL, IMCL регистра VMCON установлены в «1», то МШ запишет лог всех передач, прошедших по шине.

МШ имеет возможность при фильтрации по подадресу или коду команды управления определять слова, относящиеся к тому же сообщению. Поддерживаются все 10 форматов сообщений. При появлении неожиданного слова логика фильтрации будет перезапущена. Лог слов данных, не относящихся к какому-либо сообщению, может быть записан при установке бита в управляющем регистре.

Фильтрационную логику можно перезапустить вручную – для этого нужно сначала сбросить бит в VMEN регистра конфигурации МШ VMCON, а затем установить.

Функция фильтрации может быть выключена, при этом в лог будут внесены все события на шине, все регистры фильтрации становятся доступными только для чтения (все биты регистров будут считываться как «1»). Однако, можно контролировать, заносятся ли в лог ошибки кода Манчестера, ошибки четности.

### Отсутствие отклика

По протоколу MIL-STD-1553В битовое поле «Под адрес/Управление» командного слова может содержать значение 0h в старшем бите при использовании в режиме управления кода 00h или при передаче сообщения по подадресу 8h. Таким образом, структура командного слова будет аналогична структуре ответного слова, содержащего значение 0h в бите распознавания сообщений (бит INSTR). Поэтому в некоторых случаях может возникнуть неопределенность. Например, разрешена фильтрация по подадресу или коду команды управления, УТ не отвечает по подадресу, затем КШ снова отправляет команду УТ по подадресу 8h (или с кодом управления «00h») по той же самой шине. Это может привести к тому, что второе командное слово будет распознано как ответное слово и соответственно будет отфильтровано.

МШ может использовать бит распознавания сообщений и зарезервированные биты для исключения неопределенности. Это означает, что неопределенность не возникнет при использовании подадресов 1-7, 9-30 или кода управления 1Fh. Также неопределенность не возникнет, если используется только фильтр адреса УТ и не используется фильтр по подадресу или коду команды управления.

### Формат записи в лог

Каждая запись в лог состоит из двух 32-битных слов, как показано в таблицах 27.14 и 27.15.

Таблица 27.14 – Слово 0 записи в лог МШ

Слово 0 записи в лог МШ																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1				-												TIME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																TIME
Поле	Биты	Описание														
-	31	Зарезервирован, всегда записан как «1»														
TIME	23-0	Значение таймера														
-	30-24	Зарезервировано														

Таблица 27.15 – Слово 1 записи в лог МШ

Слово 1 записи в лог МШ																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
0						-							BUS	WST	WTP	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																WD
Поле	Биты	Описание														
-	31	Зарезервирован, всегда записан как «0»														
BUS	19	Шина, по которой получены данные														
		0	Шина А													
		1	Шина В													
WST	18-17	Статус слова														
		00	Нет ошибок в слове													
		01	Ошибка кода Манчестера													
		10	Ошибка четности													
		11	-													
WTP	16	Тип слова														
		0	Слово данных													
		1	Командное или ответное слово													
WD	15-0	Значение слова данных														
-	30-20	Зарезервировано														

## 28 Контроллер интерфейса I2C

В состав микроконтроллера входят два идентичных контроллера интерфейса I2C, обеспечивающих полную поддержку двухпроводного последовательного синхронного интерфейса I2C/SMBus. Результат такой совместимости – легкое соединение со многими запоминающими устройствами и устройствами ввода-вывода, включая EEPROM, SRAM, счетчики, АЦП, ЦАП, периферийные устройства.

### Функциональные возможности модуля контроллера I2C:

- совместимость с протоколами SMBus 1.1 и SMBus 2.0, ACCESS.Bus, I2C 2.1;
- поддержка скоростного/стандартного (FS) и высокоскоростного (HS) режимов;
- программирование действий мастера/ведомого;
- возможность подключения к шине нескольких ведущих устройств, т. е. поддержка режима мультимастер (MM);
- один программно задаваемый адрес;
- 7- или 10-битная адресация ведомого;
- поддержка адреса общего вызова.

### Особые возможности протокола SMBus:

- отслеживание времени простоя линии SCL;
- наличие функции отслеживания ошибок в пакетах данных PEC с использованием метода расчета контрольной суммы CRC;
- поддержка адреса отклика мастера;
- поддержка полинга и контроля прерываний.

## 28.1 Протокол шины

Протокол I2C использует двухпроводной интерфейс для двухсторонней связи между устройствами, подключенными к шине. Двухнаправленная шина состоит из двух линий: данных SDA и тактового сигнала SCL. Эти линии подключены к источнику питания через подтягивающие резисторы. Шинные формироваватели любых устройств, подключаемых к шине, выполняются по схеме с открытым коллектором или открытым стоком. Устройства могут выставить только низкий уровень на соответствующей линии. Следовательно, обе линии SDA и SCL реализуют функцию «монтажное И».

Протокол поддерживает режим мультимастер, в котором шина может контролироваться одним или несколькими устройствами из подключенных к шине. Каждое устройство, подключенное к шине, имеет свой адрес и может быть как приемником, так и передатчиком (некоторые только приемниками).

### Операции с данными

Устройство, которое начинает передачу данных, становится мастером. Мастер генерирует тактовый сигнал SCL, а также инициирует и завершает передачу данных по шине. За один такт сигнала SCL передается один бит данных по линии SDA, см. рисунок 28.1.

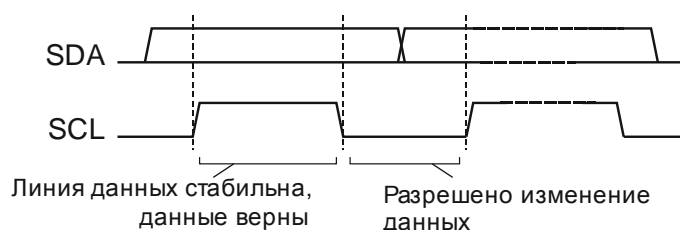


Рисунок 28.1 – Передача бита данных

Данные валидны (верны), пока уровень сигнала на линии SCL высокий. Когда на линии SCL низкий уровень сигнала, данные могут меняться.

### Старт и стоп

Состояние старта формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из высокого уровня в низкий, см. рисунок 28.2.

Состояние стопа (останова) формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из низкого уровня в высокий, см. рисунок 28.2.

Состояния старта и стопа формирует только мастер.

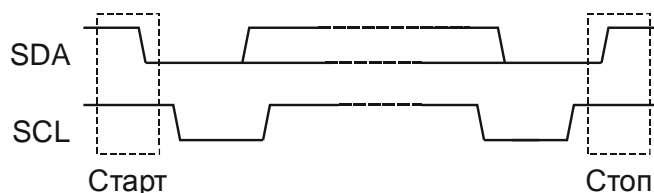


Рисунок 28.2 – Состояния старта и стопа

После того, как сформировано состояние старта, шина считается занятой и другие устройства не должны пытаться управлять ею. Шина считается занятой до тех пор, пока не будет сформировано состояние стопа. В середине передачи может быть сформировано состояние повторного старта, если мастеру нужно обратиться к другому ведомому или если требуется изменение направления передачи данных без потери контроля над шиной, см. рисунок 28.3.

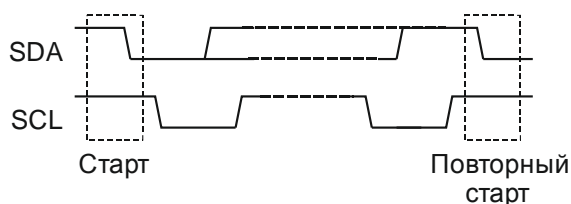


Рисунок 28.3 – Состояние повторного старта

### Арбитраж

Арбитраж выполняется в момент времени, когда на линии SCL находится «1». Два устройства могут сгенерировать стартовое состояние в одно и то же время. Далее арбитраж будет продолжаться до тех пор, пока одно из устройств сформирует «0», а другое – «1» на линии SDA. Устройство, которое установило «1» на линии SDA, проигрывает арбитраж. На рисунке 28.4 приведен пример арбитража.

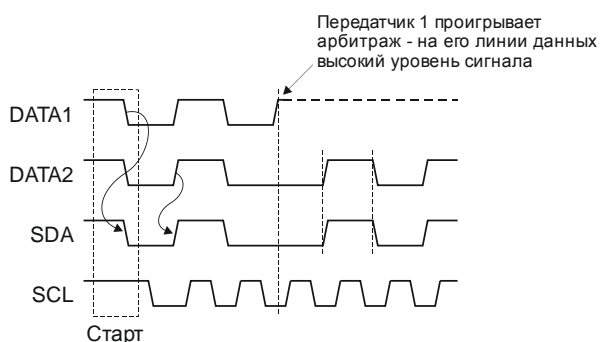


Рисунок 28.4 – Арбитраж на линии SDA

Два устройства передают свои данные DATA1 и DATA2 на линию SDA. В момент времени, когда очередной бит данных DATA1 равен «1», а бит данных DATA2 равен «0», второе устройство выигрывает арбитраж и продолжает передачу своих данных, а первое устройство прекращает передачу.

Если устройство проигрывает арбитраж во время передачи первого байта после старта (во время передачи адреса ведомого), оно становится ведомым приемником и мониторит передаваемый адрес на случай совпадения. Арбитраж также может быть проигран в режиме мастера приемника во время квитирования или в режиме ведомого передатчика во время ответа на адрес отклика на сигнал предупреждения.

В случае проигрывания арбитража в битовом поле MODE регистра ST устанавливается соответствующий код и генерируется прерывание.

### Синхронизация

Синхронизация тактовых сигналов разных устройств, подключенных к шине I2C, реализуется в случаях, когда несколько устройств являются мастерами, и выполняется с использованием той особенности, что линия SCL реализована как монтажное «И» линий тактовых сигналов этих устройств. Для примера рассмотрим синхронизацию двух мастеров с линиями тактовых сигналов CLK1 и CLK2, см. рисунок 28.5.

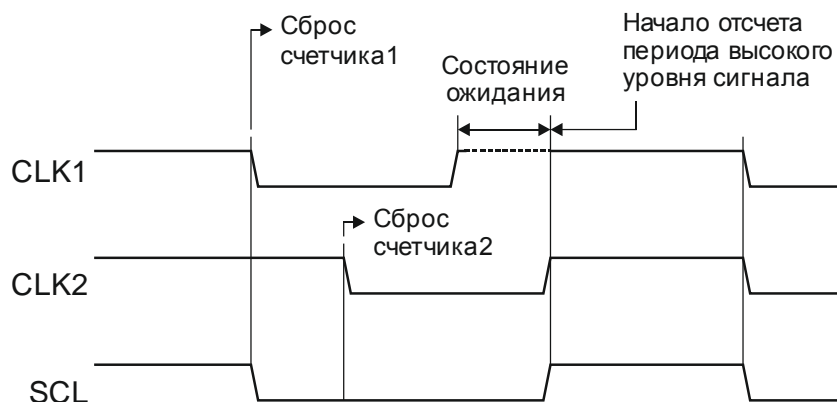


Рисунок 28.5 – Синхронизация

Линия SCL переводится в состояние «0» сразу, как только один из мастеров выставляет на своей линии тактового сигнала низкий уровень сигнала (сигнал CLK1 на рисунке 28.5). При этом его внутренний счетчик длительности низкого уровня сигнала сбрасывается и начинает отсчет. Второй мастер выставляет низкий уровень позже, и его счетчик также сбрасывается (сигнал CLK2 на рисунке 28.5).

Как только внутренний счетчик первого мастера переполнится, мастер выставит на линии CLK1 высокий уровень сигнала. Тем не менее, линия SCL будет по-прежнему оставаться в состоянии «0», удерживаемая вторым мастером. В связи с этим, первый мастер перейдет в состояние ожидания, см. рисунок 28.5. Когда переполнится счетчик второго мастера, он выставит на линии CLK2 высокий уровень сигнала, и в этот момент линия SCL перейдет в состояние «1». С этого момента внутренние счетчики длительности высокого уровня сигнала обоих мастеров начнут синхронный отсчет.

Каждая передача данных состоит из начального состояния «старт», состояний передач битов и состояния «стоп». Данные передаются старшим битом MSB вперед. Передача каждого байта завершается квитированием, т. е. приемник подтверждает окончание приема сигналом подтверждения ACK. Ведомое устройство может увеличивать паузу между тактовыми импульсами, удерживая на линии SCL сигнал низкого уровня, пока происходит обработка принятых данных или подготовка данных для следующей

передачи. Этот процесс может происходить после передачи любого бита/байта, см. рисунок 28.6.

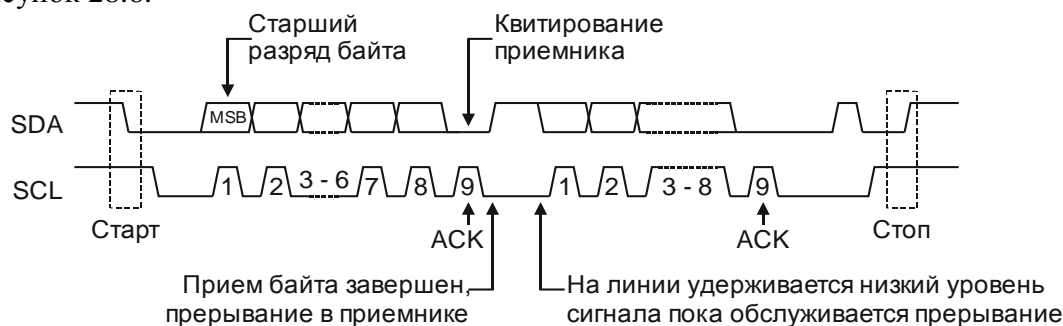


Рисунок 28.6 – Передача данных

### Квитирование

Каждый байт посылки должен быть завершен квитированием, т. е. ответом на прием сигнала запроса подтверждения приема (бит ACK). На рисунках 28.6 и 28.7 показано положение момента квитирования в пределах посылки.

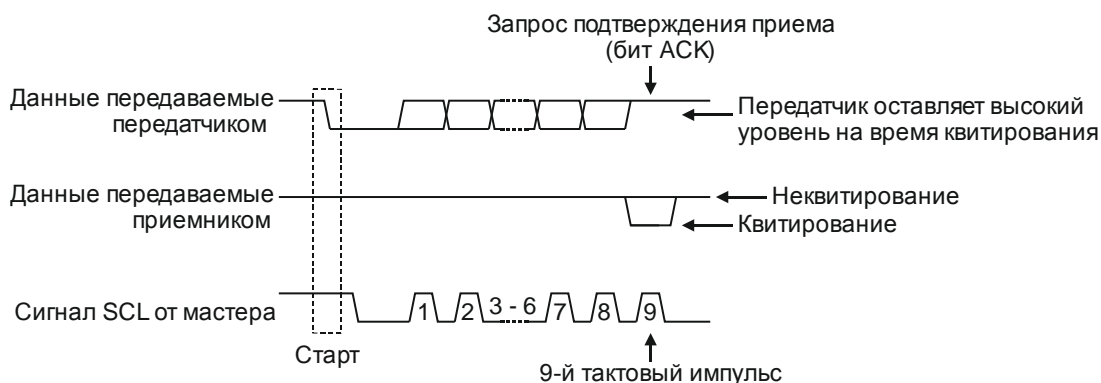


Рисунок 28.7 – Квитирование и неквитирование бита подтверждения ACK

Бит запроса подтверждения приема генерируется мастером. Передатчик (мастер или ведомый) в момент девятого такта синхросигнала оставляет линию SDA в состоянии «1» (бит ACK). В свою очередь, приемник должен сбросить линию SDA в «0» в течение времени, пока на линии SCL удерживается высокий уровень девятого импульса тактового сигнала, т. е. квитировать прием, см. рисунок 28.7. Если приемник не отвечает на запрос подтверждения и не подтверждает прием байта, то он оставляет линию SDA без изменений в состоянии «1», т. е. не квитирует прием.

Примечание – Все устройства, подсоединенные к шине I2C, в обязательном порядке должны квитировать бит ACK при получении байта с их собственным адресом. Этот механизм используется для отслеживания наличия отключившихся (самостоятельно или по каким-то причинам) от шины устройств.

Ведомое устройство имеет право не квитировать бит ACK в следующих случаях:

- если ведомый не может принять данные или он занят. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и прервать передачу. Как альтернатива, ведомый может затянуть период низкого уровня сигнала тактирования на линии SCL для завершения своих операций и продолжить передачу;
- если ведомый обнаружил некорректную команду или некорректные данные. В этом случае ведомый должен неквитировать принятый байт. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и повторить передачу;
- если мастер функционирует как приемник, то, приняв байт, он должен сообщить ведомому об окончании данных неквитированием бита ACK, посланного ведомым. После



этого ведомый передатчик должен освободить линию SDA для того, чтобы мастер смог сгенерировать состояние завершения передачи (состояние стопа).

### Формат передачи данных с 7-битной адресацией

На рисунке 28.8 показана передача адреса и двух байт данных. Каждому устройству, подключенному к шине, присваивается уникальный 7-битный адрес. Первые семь бит, передаваемые после старта, представляют собой адрес ведомого, восьмой бит R/W# определяет направление передачи – от ведомого (чтение, если R/W# = «1») или к ведомому (запись, если R/W# = «0»).

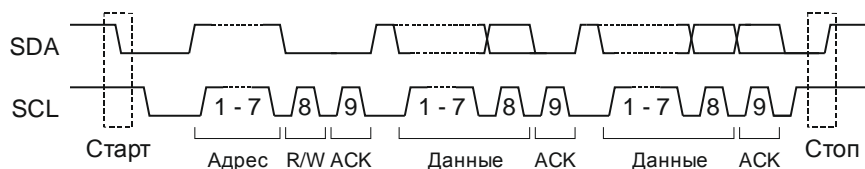


Рисунок 28.8 – Передача данных с 7-битной адресацией

Каждый ведомый, получивший байт адреса, сравнивает его со своим собственным адресом. Если адрес распознается как «свой», ведомый квитирует прием и далее, в зависимости от состояния бита R/W#, становится передатчиком или приемником.

Протокол SMBus/I2C позволяет генерировать адрес общего вызова для одновременного обращения ко всем устройствам, подключенным к шине. Первым передается адрес общего вызова (00h), затем следует байт назначения общего вызова. Ведомые, которые ожидают данные, квитируют этот байт и становятся приемниками, остальные игнорируют общий вызов.

Протокол SMBus/I2C поддерживает уникальную функцию – распознавание адреса отклика на сигнал предупреждения (Alert Response Address – ARA). В системах с несколькими ведомыми каждое устройство может послать мастеру сигнал предупреждения. Для этого используется дополнительная третья линия ALERT#, физически идентичная линиям SDA и SCL, реализованная по принципу монтажное «И». К этой линии также подключаются все устройства. Когда какому-то ведомому (или нескольким ведомым) необходимо обратиться к мастеру, он (или они) выставляет (выставляют) на линии ALERT# низкий уровень сигнала – это сигнал предупреждения, см. рисунок 28.9.

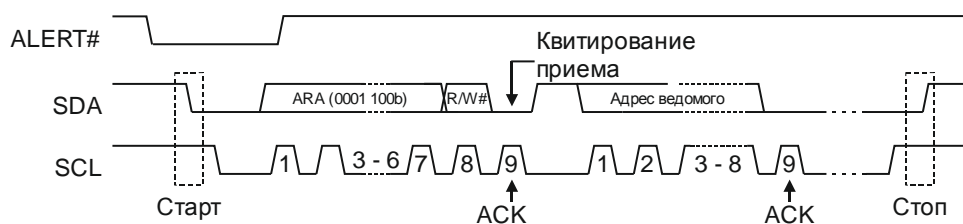


Рисунок 28.9 – Передача адреса отклика на сигнал предупреждения

Мастер, обнаружив «0» на линии ALERT#, обращается ко всем ведомым, посылая адрес отклика на сигнал предупреждения ARA. Адрес состоит из семи битов (000\_1100b) и бита R/W# = «1» (чтение). Ведомый, который отправил сигнал предупреждения, получив сигнал ARA, квитирует его и затем отправляет свой 7-битный адрес (восьмой бит может быть как «0», так и «1»), сообщая, таким образом, ведомому, какое именно устройство послало сигнал предупреждения. Кроме этого, ведомый, который выставлял «0» на линии ALERT#, должен перестать удерживать линию, чтобы на ней установился высокий уровень сигнала. В том случае, если несколько устройств посылали сигнал предупреждения, то после получения ARA, свой адрес передает то устройство, которое

захватывает шину по стандартным правилам арбитража. Если после обслуживания ведомого мастер все еще обнаруживает на линии ALERT# низкий уровень сигнала, он понимает это как то, что сигнал предупреждения посылался несколькими ведомыми. Мастер снова отправляет сигнал ARA и затем общается со следующим ведомым. Появление на линии ALERT# высокого уровня сигнала означает, что все ведомые, которые требовали обращения, обслужены.

Примечание – Описываемый в настоящем техническом описании модуль I2C не имеет выделенной линии ALERT#. При необходимости, пользователь может задействовать свободный вывод микроконтроллера и программно реализовать возможность передачи сигнала предупреждения от ведомого к мастеру. В свою очередь, функция распознавания адреса отклика ARA и последующей отправки собственного адреса реализована полностью. Включить функцию можно установкой бита SMBARE в регистре CTL0.

### Формат передачи данных с 10-битной адресацией

10-битная адресация позволяет адресовать до 1 024 ведомых устройств с использованием резервной комбинации 1111\_0xxb, которая передается по линии SDA сразу после старта. 10-битный формат полностью совместим с 7-битным форматом и может использоваться одновременно с ним, что позволяет соединять по шине I2C устройства с разной адресацией.

Основной идеей формата является передача 10-битного адреса в двух первых байтах, следующих сразу после старта. В первом байте передается значение 1111\_0xxb, где «xx» – это два старших бита адреса и бит R/W# (на рисунке 28.10 обозначен символом «W» – запись), который должен быть равен «0», чтобы ведомый понял, что в следующем байте будут переданы остальные 8 бит адреса. Во втором байте передаются 8 бит адреса, см. рисунок 28.10.

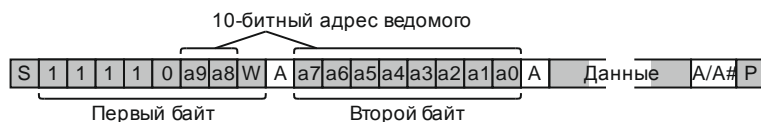


Рисунок 28.10 – Передача данных ведомому с 10-битным адресом (для расшифровки применяемых обозначений следует обратиться к таблице 28.1)


Чтобы осуществить чтение ведомого, которого адресует мастер после второго байта адреса, следует отправить бит повторного старта и затем комбинацию 1111\_0xxb и бит R/W# (обозначен символом «R» – чтение), который на этот раз равен «1», см. рисунок 28.11.



Рисунок 28.11 – Получение данных от ведомого с 10-битным адресом

На рисунках 28.10 и 28.11 биты посылки условно обозначены буквами S, W и др. или состояния битов указаны непосредственно «0» или «1». В дальнейшем на подобных рисунках, поясняющих содержимое посылки при передаче или приеме данных, будут применяться такие же и другие обозначения. Все обозначения, которые будут использоваться, указаны в таблице 28.1 с подробными пояснениями.

Таблица 28.1 – Условные обозначения, принятые на рисунках, показывающих содержимое посылок данных или адресов на линии SDA

Обозначение	Расшифровка обозначения
S	Состояние старта. Символом «S» обозначается стартовый бит посылки
SR	Состояние повторного старта
R/W	Бит указания направления передачи. В тексте настоящего описания он упоминается как R/W#. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1»
R	Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «R», то это указывает на то, что в данной посылке бит R/W# должен быть равен «1», т. е. направление передачи данных происходит от ведомого к мастеру (чтение)
W	Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «W», то это указывает на то, что в данной посылке бит R/W# должен быть равен «0», т. е. направление передачи данных происходит от мастера к ведомому (запись)
A/A#	Бит квитированного/неквитированного приема, посылаемый приемником в ответ на запрос передатчика подтвердить прием. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1»
A	Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло квитирование, т. е. приемник установил линию SDA в «0». В тексте настоящего описания квитированный бит запроса подтверждения приема обозначается как ACK
A#	Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A#», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло неквитирование, т. е. приемник не изменил линию SDA и оставил ее в состоянии «1». В тексте настоящего описания, неквитированный бит запроса подтверждения приема обозначается как NACK
P	Состояние окончания передачи. Символом «P» обозначается стоповый бит посылки
Код мастера	8-битный код мастера. Значение 0000_1xxx <sub>b</sub> , где «xxx» – уникальный код каждого мастера в системе нескольких устройств
Адрес	7-битный адрес ведомого, передаваемый мастером
Адрес ведомого	Адрес ведомого, передаваемый во втором байте посылки. В режиме HS это 7-битный адрес, на что указывает идущий следом бит R/W#, в остальных случаях это восемь младших бит 10-битного адреса
Данные	Байт или несколько байт данных
GC	Байт адреса общего вызова (0000_0000 <sub>b</sub> )
AR	Адрес отклика (0001_100 <sub>b</sub> )
	Изображение числа в овале с линией, прикрепляющей его к изображению передачи битов, обозначает код операции и указывает момент, в который этот код записывается в поле MODE регистра ST
Цветное поле	Серым цветом обозначены биты, передаваемые от мастера к ведомому
Белое поле	Белым цветом обозначены биты, передаваемые от ведомого к мастеру

## 28.2 Функциональное описание

Структурная схема модуля I2Cx (где x - 0 или 1) представлена на рисунке 28.12. Далее приводится краткое описание назначения блоков модуля.

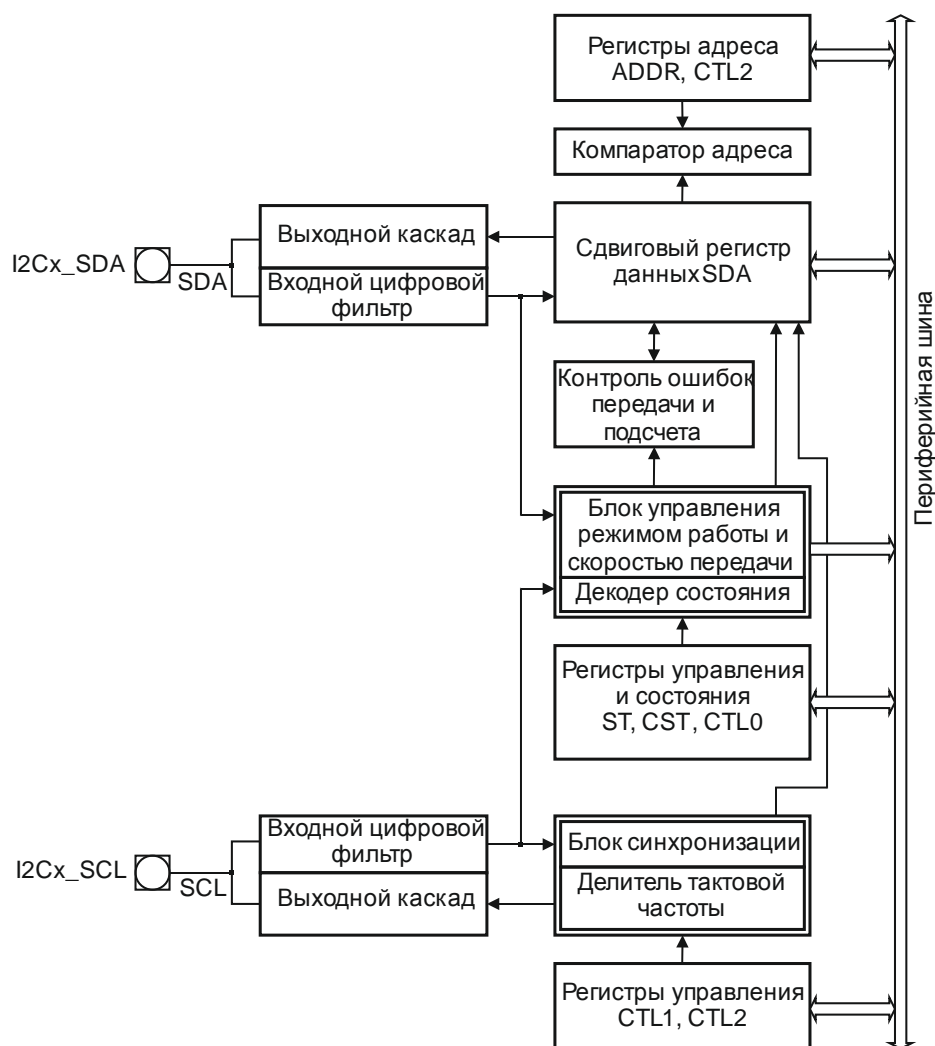


Рисунок 28.12 – Структурная схема модуля I2C

### Входные и выходные каскады линий SDA и SCL

Для обеих линий используются входные шумовые фильтры. В режиме FS эти фильтры подавляют любые импульсы входного сигнала, длительность которых не превышает один такт системного синхросигнала. Выходные каскады включают в себя понижающие (до уровня «0») устройства с открытым стоком. Функционирование входных и выходных каскадов зависит от состояния модуля I2C, т.е. модуль включен или выключен.

### Управление режимом работы и опрос состояния

Управление модулем I2C осуществляют блоки управления режимом работы и скоростью передачи, регистров управления и состояния. В состав этих блоков входят следующие регистры:

- ST – содержит биты, отражающие текущую конфигурацию модуля I2C (мастер или ведомый, передатчик или приемник) и бит флага прерывания;
- CST – является одновременно регистром управления шиной и регистром состояния шины;

- CTL0 – управляет генерированием состояний старта, повторного старта и останова, а также квитированием;
- CTL1 и CTL2 – устанавливают параметры тактового сигнала в режиме мастера и контролируют режим 10-битной адресации.

### Регистры адреса и компаратор адреса

В регистр адреса ADDR может быть записан 7-битный адрес, который является адресом устройства при работе его в режиме ведомого. Распознавание адреса включается установкой бита SAEN.

Компаратор адреса сравнивает принятый 7-битный адрес со значением, хранящимся в поле ADDR. Если разрешено распознавание адреса общего вызова (установлен бит GCMEN регистра CTL0), то компаратор сравнивает принятый адрес со значением 0000\_000b. Если разрешено распознавание адреса отклика на сигнал предупреждения (установлен бит SMBARE регистра CTL0), то компаратор сравнивает принятый адрес со значением 0001\_100b.

Если включен режим 10-битной адресации (одновременно установлены биты SAEN и S10EN регистров ADDR и CTL2, соответственно), компаратор сравнивает старшие пять битов первого полученного байта со значением 1111\_0b, а следующие два бита – со значением второго и первого битов поля S10ADR регистра CTL2. Старший бит второго полученного байта сравнивается со значением нулевого бита поля S10ADR, а оставшиеся семь битов – со значением битового поля ADDR регистра ADDR, см. рисунок 28.13.

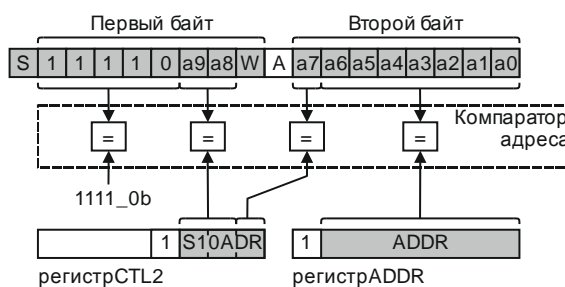


Рисунок 28.13 – Компаратор адреса в режиме 10-битной адресации

### Сдвиговый регистр данных

Регистр SDA представляет собой сдвиговый регистр, используемый для приема и передачи данных. Старший бит регистра передается/принимается первым, младший бит – последним. Запись в регистр SDA возможна только, если установлен бит INT регистра ST. Регистр может быть прочитан в любой момент времени, но прочитанные данные будут гарантированно достоверными только при установленном бите INT. Регистр SDA не очищается при сбросе и хранит случайные данные до тех пор, пока не будет перезаписан программно или аппаратно после приема байта.

### Генерация тактового сигнала и синхронизация

Последовательный тактовый сигнал (выходной сигнал модуля I2C в режиме мастера) формируется генератором на базе системного тактового сигнала (с частотой  $F_{PCLK}$ ).

Модуль I2C может функционировать в двух глобальных режимах – стандартном/скоростном FS и высокоскоростном HS.

В режиме FS используется 15-битный предделитель. Значение младших 6 бит определяется значением битового поля SCLFRQ регистра CTL1, а нулевой бит всегда равен нулю (деление производится только на четное число). Значение старших 8 бит задается полем SCLFRQ регистра CTL3. Блок синхронизации тактового сигнала

производит синхронизацию генератора тактового сигнала и выходного сигнала SCL с тактовым сигналом других устройств, подключенных к шине.

В режиме HS используется 12-битный делитель. Значение младших 4 бит определяется значением битового поля HSDIV регистра CTL2, нулевой бит при этом всегда равен нулю. Значение старших 8 бит задается полем HSDIV регистра CTL4.

Определяемое спецификацией протокола SMBus наименьшее время ожидания на линии SCL составляет 25 мс. Если пауза между двумя тактовыми импульсами превысила 25 мс, то устройство должно прервать текущую передачу. Мастер должен сформировать состояние старта в процессе передачи или после ее окончания. Водомый должен освободить шину. Устройства, обнаружившие данное состояние, должны восстановить свои соединения и ожидать формирования состояния старта в пределах 10 мс.

Для отслеживания периодов ожиданий на шине в модуле I2C имеется счетчик времени ожидания. Функциональная схема счетчика показана на рисунке 28.14.

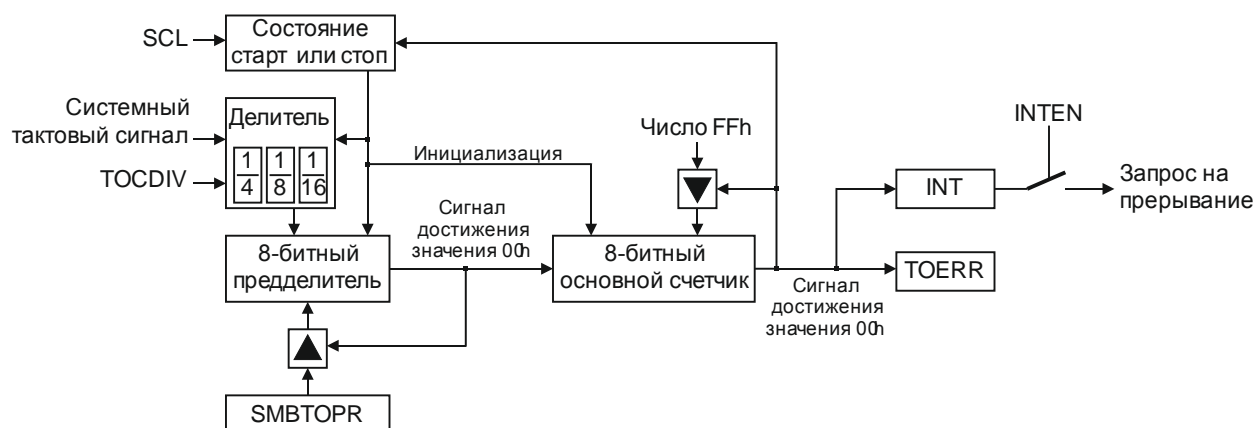


Рисунок 28.14 – Функциональная схема счетчика времени ожидания

Счетчик времени ожидания состоит из делителя, 8-битного программируемого делителя и 8-битного основного счетчика. Все элементы счетчика времени ожидания начинают работу по отрицательному фронту сигнала на линии SCL (если работа счетчика разрешена). Положительный фронт сигнала на линии SCL сбрасывает значения делителя, делителя и основного счетчика. Делитель считает вниз, начиная со значения, записанного в регистр TOPR. После достижения нуля счетчиком делителя, он загружается значением из регистра TOPR. Основной счетчик считает вниз от значения FFh. Каждое достижение нуля делителем декрементирует значение основного счетчика. Обнуление основного счетчика и загрузка его значением FFh вызывает остановку основного счетчика делителя и делителя и установку флага TOERR в регистре CST. Дополнительно устанавливается флаг INT, и если разрешено, генерируется прерывание.

Период времени ожидания определяется следующим выражением

$$T_{\text{ожид}} = T_{\text{PCLK}} \times \text{TOCDIV} \times (\text{SMBTOPR} + 1) \times 256, \quad (28.1)$$

где  $T_{\text{PCLK}}$  – период системного тактового сигнала с частотой  $F_{\text{PCLK}}$ .

### Арбитраж и обнаружение ошибок на шине I2C

Арбитраж в режиме мастера передатчика может быть потерян в случае, когда два мастера одновременно формируют состояние старта и начинают передачу данных. Потеря арбитража может происходить как во время передачи адреса, так и во время передачи данных.

В случае потери приоритета при передаче байта адреса, мастер переходит в режим ведомого приемника и начинает принимать адрес. Если принятый адрес оказался «своим»,

модуль I2C далее функционирует в режиме ведомого. Если принятый адрес не оказался «своим», то модуль I2C переходит в режим безадресного ведомого.

В случае потери приоритета при передаче байта данных модуль I2C сразу переходит в режим безадресного ведомого.

### **Обнаружение и исправление ошибок на шине I2C**

Состояние ошибки на шине I2C возникает в том случае, если во время передачи адреса/данных или во время квитирования на шине I2C обнаруживаются состояния старта или стопа. При обнаружении ошибки на шине I2C выполняются действия:

- в поле MODE регистра ST записывается код ошибки 1Fh;
- генерируется прерывание (если разрешено);
- модуль I2C переходит в режим безадресного ведомого;
- линии SDA и SCL освобождаются.

Обнаружение ошибки на шине I2C может вызвать у простой шины некорректное формирование состояния старта и отключение модуля I2C. Поэтому для возврата к нормальной работе следует выполнить действия:

- выключить и снова включить модуль I2C (бит ENABLE в регистре CTL1);
- в течение времени простоя проверить, не подключен ли другой активный мастер к шине I2C (бит BB регистра CST должен быть обнулен);
- в режиме мастера сформировать состояние старта, передать адрес и затем сформировать состояние останова, таким образом, проведя синхронизацию всех ведомых устройств (в том числе и тех, которые не обнаружили ошибку на шине I2C).

### **Режим IDLE**

Переход в режим IDLE происходит при отключении внешнего сигнала тактирования модуля I2C записью нуля в бит I2CEN регистра APB\_CLK. Переход в режим IDLE подобен программному выключению модуля I2C (очистка бита ENABLE в регистре CTL1). Регистры CTL0, ST и CST очищаются, чтобы гарантировать нормальный старт после возобновления функционирования модуля.

Выход из режима IDLE осуществляется записью единицы в бит I2CEN и включением модуля I2C битом ENABLE.

## **28.3 Инициализация и функционирование**

В целом модуль I2C поддерживает два базовых режима – режим FS и режим HS.

Стандартный/скоростной режим или режим FS – стандартный режим работы, в котором модуль функционирует по умолчанию. Диапазон частот сигнала на линии SCL – от 763 Гц до 6,25 МГц (при  $f_{PCLK} = 100$  МГц).

Высокоскоростной режим или режим HS – режим работы, который включается программно. Режим HS значительно превосходит режим FS по скорости – диапазон частот сигнала на линии SCL от 8,14 кГц до 16,67 МГц (при  $f_{PCLK} = 100$  МГц).

Все операции режима HS начинаются в режиме FS в следующем порядке:

- стартовое состояние;
- 8-битный код мастера (значение 0000\_1xxxh, где «xxx» – уникальный код каждого мастера в системе нескольких устройств);
- неквитирование.

Арбитраж на шине I2C происходит в момент передачи несколькими мастерами своих уникальных кодов. Выигравший арбитраж мастер захватывает шину. В связи с такой организацией режима HS дальнейший арбитраж и синхронизация на шине не реализуются.

После выполнения вышеуказанных шагов устройства, поддерживающие режим HS, переключаются в этот режим. Мастер генерирует состояние повторного старта SR, а затем



передает адрес ведомого и бит направления передачи R/W#, см. рисунок 28.15. Расшифровка обозначений, принятых на рисунке 28.15, приведена в таблице 28.1.

Все передачи в режиме HS по формату идентичны передачам режима FS, что делает эти два режима полностью совместимыми.

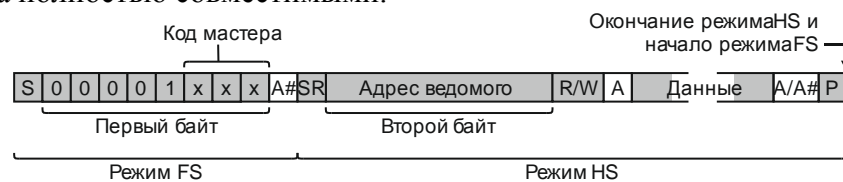


Рисунок 28.15 – Переход в режим HS и обратно в режим FS

Выход из режима HS происходит генерированием состояния окончания передачи P, после которого все устройства переключаются обратно в режим FS. В каждом из двух базовых режимов – FS и HS – модуль I2C может функционировать как мастер или ведомый, получать или передавать информацию.

Далее все режимы работы модуля I2C будут рассмотрены подробно.

### Инициализация

Для начала работы следует произвести инициализацию:

- 1 Включить модуль I2C установкой бита ENABLE в регистре CTL1.
- 2 Если активен режим мастера, записать нужный коэффициент деления в битовое поле SCLFRQ (регистры CTL1, CTL3) для выбора периода тактового сигнала SCL (для режима HS записать коэффициент деления в поле HSDIV регистров CTL2, CTL4).
- 3 Если активен режим ведомого, необходимо:
  - записать «собственный» адрес ведомого в битовое поле ADDR и установить бит SAEN регистра ADDR;
  - для реализации 10-битной адресации записать старшие биты адреса в битовое поле S10AD и установить бит S10EN регистра CTL2;
  - для включения функции распознавания адреса общего вызова установить бит GCMEN в регистре CTL0;
  - для включения функции распознавания адреса отклика установить бит SMBARE в регистре CTL0.
- 4 При необходимости отслеживания периодов ожидания на шине I2C записать желаемые значения в регистр TOPR и в битовое поле TOCDIV (регистр CST) для отсчета времени ожидания на линии SCL. Для автоматического отслеживания времени ожидания записать ненулевое значение в битовое поле TOCDIV регистра CST.
- 5 Для разрешения формирования запроса на прерывание установить бит INTEN в регистре CTL0.

### Функционирование

Модуль I2C может работать в режиме мастера или ведомого. Также он может функционировать как передатчик или приемник, т. е. модуль I2C поддерживает девять режимов:

- безадресный ведомый;
- мастер передатчик в режиме FS;
- мастер передатчик в режиме HS;
- мастер приемник в режиме FS;
- мастер приемник в режиме HS;
- ведомый передатчик в режиме FS;
- ведомый передатчик в режиме HS;
- ведомый приемник в режиме FS;
- ведомый приемник в режиме HS.

Передача информации по шине I2C состоит из последовательности различных действий (начало передачи, прием данных и др.). Каждое действие называется состоянием



(состояние старта, состояние останова и др.). После того, как то или иное состояние сформировано, его код аппаратно записывается в регистр ST в битовое поле MODE и может быть прочитано программно. В таблицах 28.2 и 28.3 приводятся все возможные состояния, их мнемонические обозначения и коды. На квитирование или неквитирование приема указывает запись «ACK» или «NACK», соответственно. Так, например, если мастер отправил байт адреса ведомому, который после получения квитировал прием, то на это будет указывать «ACK», а в поле MODE регистра ST будет записан код 04h, соответствующий состоянию с мнемоническим обозначением «MTADPA». Более подробно каждый режим работы модуля I2C будет рассмотрен далее. На рисунках 28.16 – 28.23, поясняющих работу модуля I2C в том или ином режиме, приняты обозначения, расшифровка которых приводится в таблице 28.1. Для получения дополнительной информации и понимания работы модуля I2C можно воспользоваться приложением Б.

Таблица 28.2 – Коды функционирования модуля I2C в режиме FS

Режим	Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK	
Общий	00h	IDLE	IDLE, нет доступной валидной информации о статусе	–	
Мастер в режиме FS	–	01h	STDONE	Сформировано состояние старта	–
		02h	RSDONE	Сформировано состояние повторного старта	–
		03h	IDLARL	Потеря арбитража, переход в режим безадресного ведомого	–
	Передача	04h	MTADPA	Отправлен адрес ведомого	ACK
		05h	MTADNA	Отправлен адрес ведомого	NACK
		06h	MTDAPA	Отправлен байт данных	ACK
		07h	MTDANA	Отправлен байт данных	NACK
	Прием	08h	MRADPA	Отправлен адрес ведомого	ACK
		09h	MRADNA	Отправлен адрес ведомого	NACK
		0Ah	MRDAPA	Принят байт данных	ACK
		0Bh	MRDANA	Принят байт данных	NACK
	–	0Ch	MTMCER	Отправлен код мастера, обнаружена ошибка	ACK
–	0Dh – 0Fh		Зарезервировано. Не использовать!	–	
Ведомый в режиме FS	Прием	10h	SRADPA	Принят адрес	ACK
		11h	SRAAPA	Принят адрес после потери арбитража	ACK
		12h	SRDAPA	Принят байт данных	ACK
		13h	SRDANA	Принят байт данных	NACK
	Передача	14h	STADPA	Принят адрес	ACK
		15h	STAAPA	Принят адрес после потери арбитража	ACK
		16h	STDAPA	Отправлен байт данных	ACK
		17h	STDANA	Отправлен байт данных	NACK
	Передача адреса отклика	18h	SATADP	Принят адрес отклика на предупреждение	ACK
		19h	SATAAP	Принят адрес отклика на предупреждение после потери арбитража	ACK
		1Ah	SATDAP	Отправлены данные в ответ на получение адреса отклика	ACK
		1Bh	SATDAN	Отправлены данные в ответ на получение адреса отклика	NACK

Окончание таблицы 28.2

Режим		Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK
Ведомый в режиме FS	–	1Ch	SSTOP	Обнаружено состояние останова ведомого	–
		1Dh	SGADPA	Принят адрес общего вызова	ACK
		1Eh	SDAAPA	Принят адрес общего вызова после потери арбитража	ACK
Общий		1Fh	BERROR	Обнаружена ошибка на шине (некорректное состояние старта или останова)	–
Примечание – Диапазон значений кодов 0Dh–0Fh зарезервирован и не доступен для использования. Дополнительная информация находится в приложении Б данного ТО.					

Таблица 28.3 – Коды функционирования модуля I2C в режиме HS

Режим		Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK
Мастер в режиме HS	–	21h	HMTMCOK	Код мастера передан успешно, переход в режим HS	–
		22h	HRSDONE	Сформировано состояние повторного старта	–
		23h	HIDLARL	Потеря арбитража, переход в режим HS безадресного ведомого	–
	Передача	24h	HMTADPA	Отправлен адрес ведомого	ACK
		25h	HMTADNA	Отправлен адрес ведомого	NACK
		26h	HMTDAPA	Отправлен байт данных	ACK
		27h	HMTDANA	Отправлен байт данных	NACK
	Прием	28h	HMRADPA	Отправлен адрес ведомого	ACK
		29h	HMRADNA	Отправлен адрес ведомого	NACK
		2Ah	HMRDAPA	Принят байт данных	ACK
		2Bh	HMRDANA	Принят байт данных	NACK
Ведомый в режиме HS	Прием	30h	HSRADPA	Принят адрес	ACK
		32h	HSRDAPA	Принят байт данных	ACK
		33h	HSRDANA	Принят байт данных	NACK
	Передача	34h	HSTADPA	Принят адрес	ACK
		36h	HSTDAPA	Отправлен байт данных	ACK
		37h	HSTDANA	Отправлен байт данных	NACK
Примечание – Диапазоны значений кодов 2Ch–2Fh и 38h–3Fh, а также коды 20h, 31h, 35h зарезервированы и недоступны для использования. Дополнительная информация находится в приложении Б данного ТО.					

**Режим безадресного ведомого**

Режим работы по умолчанию (MODE = 00h). После включения модуль I2C начинает функционировать в режиме безадресного ведомого и непрерывно мониторит шину. При обнаружении состояния старта или повторного старта переходит в режим ведомого приемника. Для перехода в режим мастера передатчика нужно сформировать корректное состояние старта.

Переключение в режим безадресного ведомого происходит в случаях:

- стартовое состояние не было успешно сформировано, так как другое устройство удерживало на линии SCL низкий уровень сигнала;

- произошла потеря арбитража во время передачи байта данных в режиме мастера передатчика или во время передачи бита R/W# в режиме мастера приемника;
- произошла потеря арбитража во время ответа на полученный адрес отклика;
- неквитирование принятого адреса в режиме ведомого приемника (адрес не совпал со «своим» или запрещен);
- неквитирование в конце переданного байта в режиме ведомого передатчика;
- обнаружено состояние останова;
- обнаружена ошибка на шине;
- модуль I2C был сброшен;
- модуль I2C был выключен.

### **Режим FS мастера передатчика**

Включение режима FS мастера передатчика:

1 Переход в режим мастера передатчика происходит после успешного формирования состояния старта. Первый байт, передаваемый мастером сразу после старта, состоит из адреса ведомого и бита направления.

2 В зависимости от состояния бита направления (R/W#), модуль I2C далее функционирует как мастер передатчика (если R/W# = «0») или как мастер приемника (если R/W# = «1»). Для перехода в режим HS мастер может передать код мастера (0000\_1xxxh) вместо первого байта адреса.

3 Переход в режим мастера произойдет после установки бита START в регистре CTL0. Если бит BB в регистре CST сброшен, т. е. шина свободна, будет сгенерировано состояние старта. Если бит BB = 1b, то бит START останется установленным, а состояние старта будет сгенерировано по истечении времени, равного одному такту сигнала тактирования на линии SCL, после освобождения шины.

4 Как только стартовое состояние будет сгенерировано успешно, бит START сбросится, модуль I2C перейдет в состояние STDONE (в поле MODE запишется значение 01h), установится флаг INT, и линия SCL будет удерживаться в «0» до тех пор, пока флаг INT не будет сброшен. Если разрешено битом INTEN (регистр CTL0), сгенерируется прерывание.

Передача адреса и данных:

1 Пока удерживается флаг INT, программа записывает адрес ведомого и бит направления передачи в регистр данных SDA (адрес записывается в биты с седьмого по первый).

2 После записи в регистр SDA флаг INT сбрасывается программно установкой бита CLRST в регистре CTL0.

3 После сброса флага INT и по истечении времени, требуемого для установки данных, на линии SCL появляется тактовый сигнал и данные, хранящиеся в регистре SDA, начинают передаваться по линии SDA.

4 После завершения передачи байта и получения ответа на запрос подтверждения передачи (ACK), т. е. после девятого такта сигнала тактирования на линии SCL, аппаратная часть анализирует квитирование/неквитирование передачи и устанавливает соответствующий код в поле MODE.

5 Во время передачи линии SCL и SDA постоянно мониторятся с целью выявления возможных конфликтов с другими устройствами, подключенными к шине. В случае обнаружения конфликта передача прерывается и в поле MODE записывается код 11b (состояние SRAAPA – переход в режим ведомого приемника после потери арбитража) или код 03h (состояние IDLARL – переход в режим безадресного ведомого после потери арбитража).

6 Если бит направления равен единице и не обнаружено ошибок на шине, модуль I2C переходит в режим мастера приемника.

7 Если бит направления равен нулю и передача адреса ведомого завершена успешно (значение кода в поле MODE не равно 05h/1Fh), устанавливается флаг INT, указывая на то, что ожидается запись первого байта данных в регистр SDA для дальнейшей передачи, и, если разрешено, генерируется прерывание. Пока флаг INT будет оставаться установленным, линия SCL будет удерживаться в «0».

8 Байт данных записывается программно в регистр SDA и передача продолжается.

9 Если ведомый приемник не квитирует отправленный ему байт данных, в поле MODE записывается код 0Bh (состояние MRDANA). На линии SCL будет установлен низкий уровень сигнала и, если разрешено, сгенерировано прерывание.

Для отслеживания ошибок в пакетах данных применяется механизм вычисления контрольной суммы (CRC) для нескольких байт данных. В режиме мастера передатчика установка бита PECNEXT в регистре CST вызовет перенос содержимого регистра ошибок (не доступен программно) в регистр SDA и инициирует передачу байта CRC (байт контрольной суммы) ведомому. Передача байта CRC должна выполняться после передачи последнего байта данных и перед формированием состояния останова или повторного старта.

Мастер передатчика контролирует шину и может адресовать любое ведомое устройство и изменять направление передачи без потери контроля над шиной, используя возможность формирования состояния повторного старта. Для формирования состояния следует:

1 Установить бит START.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг прерывания INT.

После этих действий будет освобождена линия SCL, сгенерировано состояние повторного старта и сгенерировано прерывание. В поле MODE будет записан код 02h (состояние RSDONE).

Модуль I2C может быть выведен из режима мастера передатчика генерированием состояния останова. Для этого необходимо:

1 Установить бит STOP в регистре CTL0.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг INT.

Вышеуказанные действия приведут к незамедлительному формированию состояния останова и очистке бита STOP.

Состояние останова может быть сформировано только если модуль I2C функционирует как мастер и контролирует шину (в поле MODE находится любое значение кода из диапазона 01h – 0Bh).

Дополнительно можно обратиться к приложению Б данного технического описания.

На рисунке 28.16 представлено графическое пояснение к описанию режима.

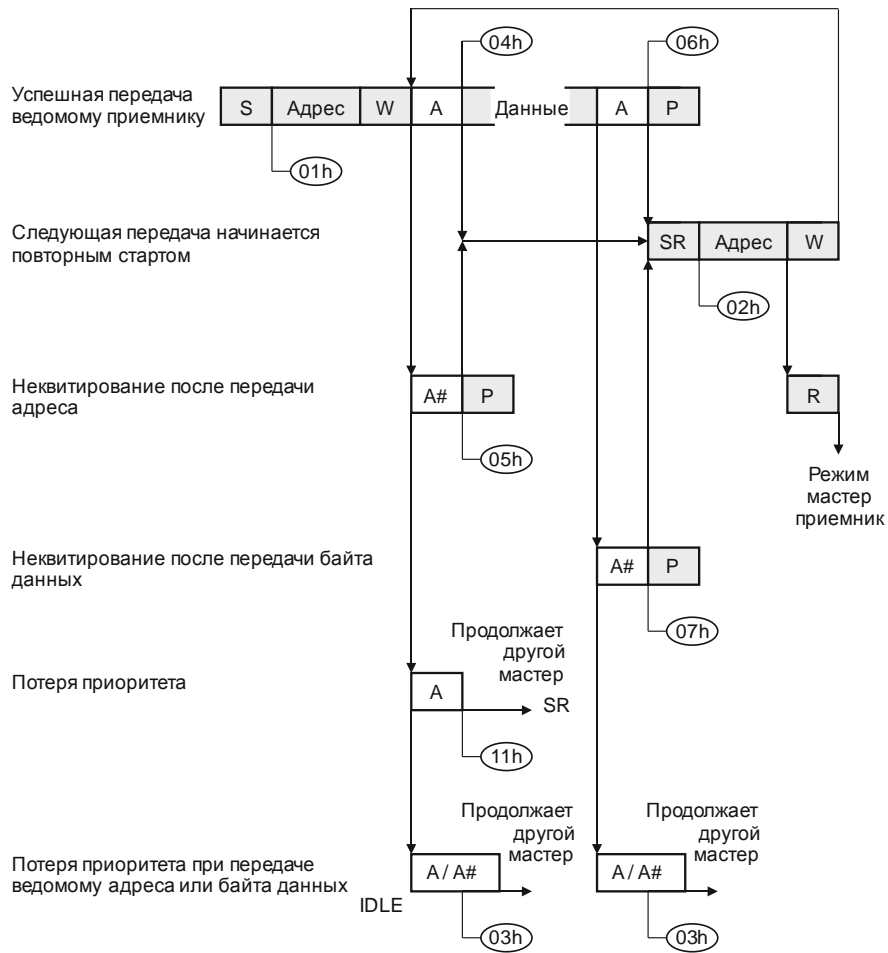


Рисунок 28.16 – Режим FS мастера передатчика

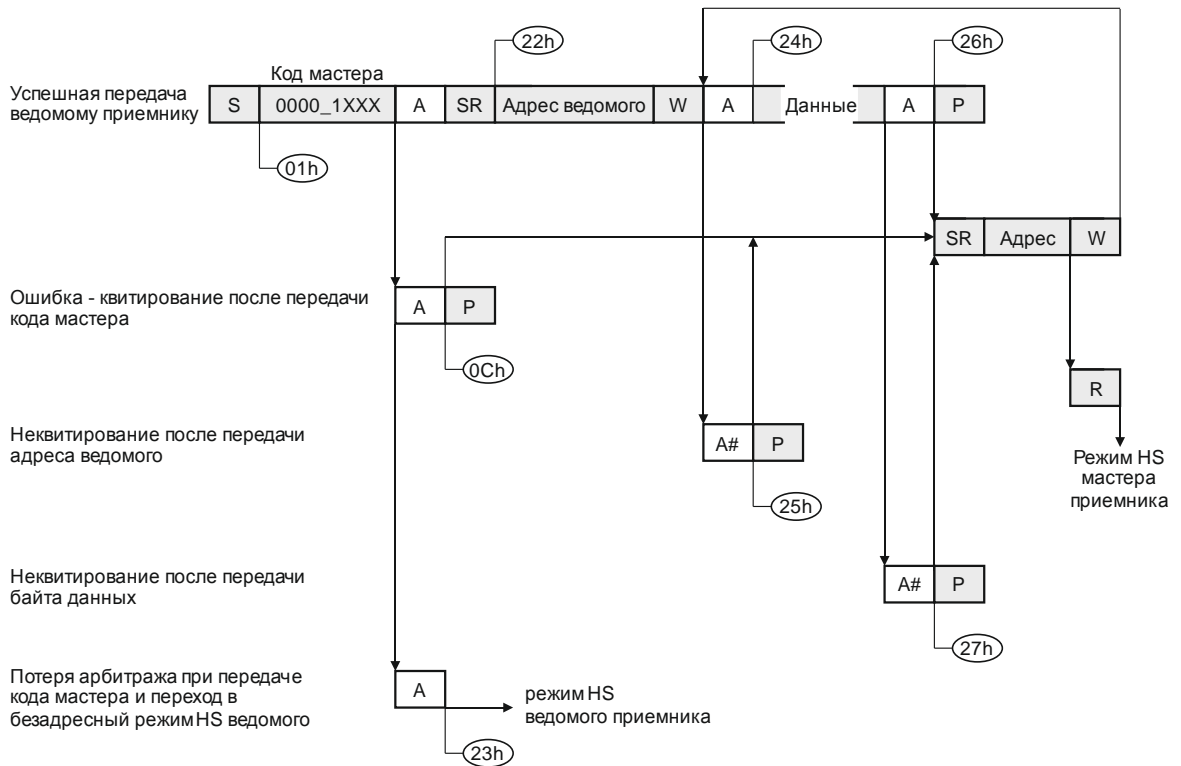


Рисунок 28.17 – Режим HS мастера передатчика

### **Режим HS мастера передатчика**

Переход в режим HS мастера передатчика происходит в том случае, если после состояния старта мастер передает код мастера (0000\_1xxx<sub>b</sub>) вместо адреса ведомого. По окончании передачи кода мастера устанавливается флаг INT и, если разрешено, генерируется прерывание. Вслед за успешной передачей кода мастера в поле MODE записывается код 21h (состояние HMTMCOК), и мастер переходит в режим HS.

Далее необходимо сформировать состояние повторного старта, записав единицу в бит START, и сбросить флаг INT, записью единицы в бит CLRST.

После сгенерированного состояния повторного старта устанавливается флаг INT, и в поле MODE записывается код 22h (состояние HRSDONE). Дальнейший порядок действий по передаче адреса и данных аналогичен описанному режиму FS мастера передатчика.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.17 представлено графическое пояснение к описанию режима.

### **Режим FS мастера приемника**

Переход в режим мастера приемника происходит после успешной передачи адреса ведомого с единичным битом направления (R/W# = «1»). В режиме мастера приемника модуль I2C получает данные от ведомого устройства, поэтому теряет контроль над шиной SDA. В тоже время мастер продолжает тактировать передачу и должен отвечать на бит ACK каждого принятого байта.

После каждого принятого байта устанавливается флаг INT, и пользовательская программа читает полученные данные из регистра SDA. Линия SCL удерживается в «0», пока установлен флаг INT. После сброса флага INT может стартовать прием следующего байта. После этого (согласно протоколу SMBus) состояния повторного старта или стопа не должны генерироваться мастером, поскольку мастер теперь не является единственным контролером линии SDA. В конце приема каждого байта мастер не квитирует прием, сообщая, таким образом, ведомому об успешном приеме.

После приема предпоследнего байта перед сбросом флага INT следует записать ноль в бит ACK регистра CTL0. В тоже время, если требуется отправка байта CRC, следует установить бит PECNEXT в регистре CST. После сброса флага INT будет принят последний байт данных и не квитируется. По окончании приема мастер возвращается в режим передатчика и теперь может сгенерировать состояние повторного старта или останова.

Если механизм отслеживания ошибок включен, то последний переданный от ведомого байт будет байтом CRC. В случае если результат вычисления контрольной суммы не нулевой, то установится флаг ошибки PECFAULT в регистре CST.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.18 представлено графическое пояснение к описанию режима.

### **Режим HS мастера приемника**

Переход в режим HS мастера приемника происходит, если после переданного кода мастера и последовавшего за ним состояния повторного старта, производится передача адреса ведомого с битом направления R/W# = «1». Модуль I2C переходит в режим HS мастера приемника, устанавливается флаг INT, а в поле MODE записывается соответствующий код из диапазона 28h – 2Bh.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.19 представлено графическое пояснение к описанию режима.

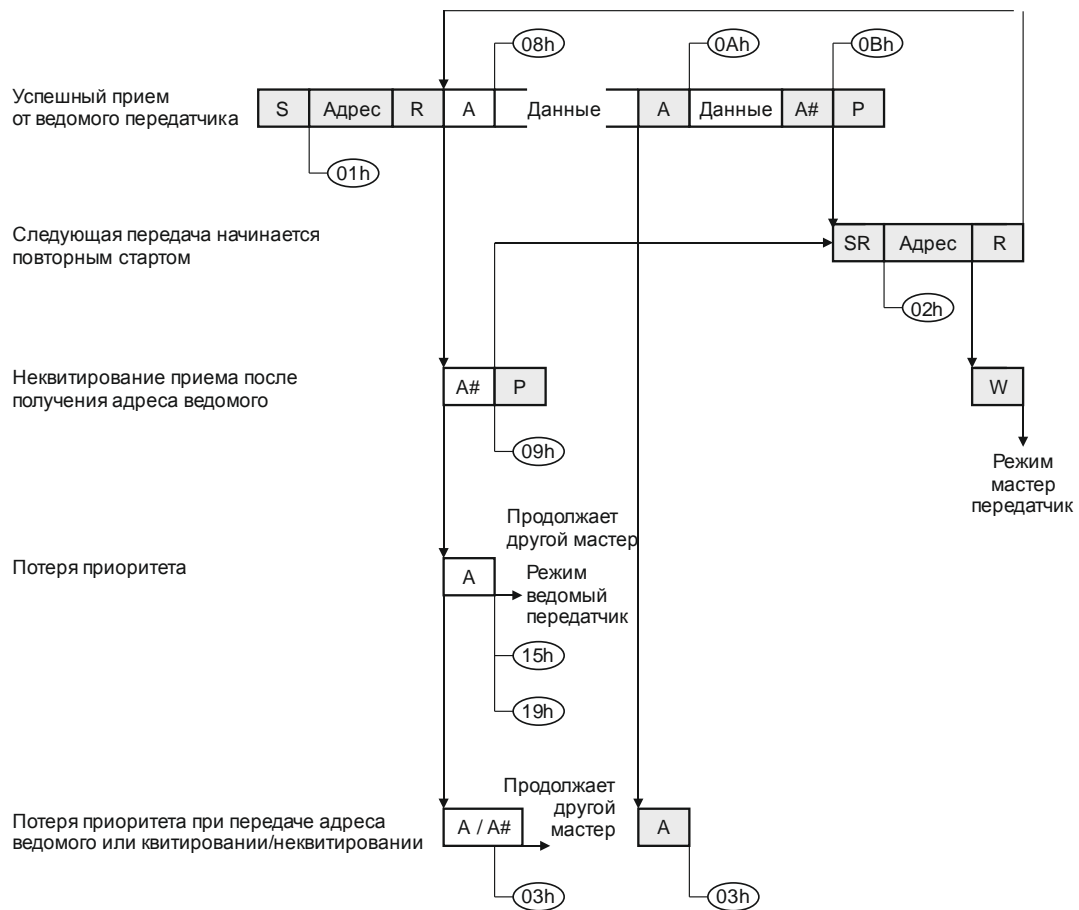


Рисунок 28.18 – Режим FS мастера приемника

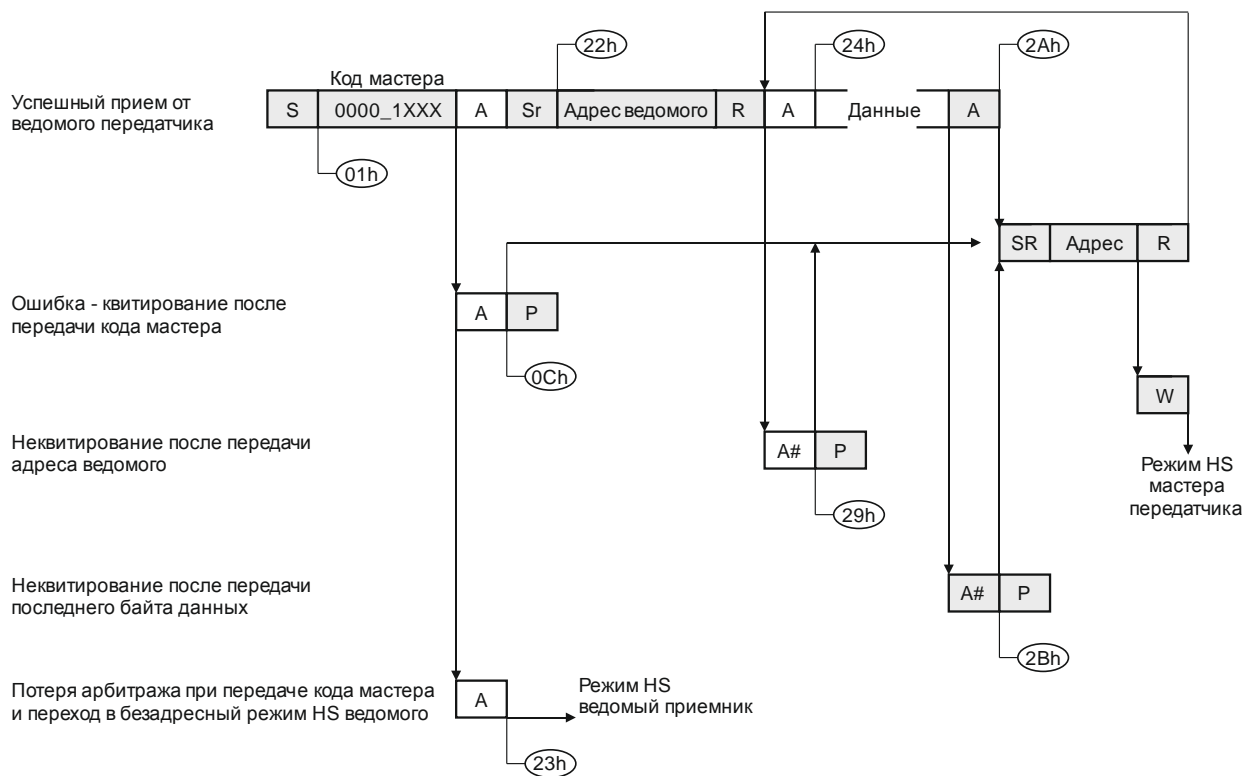


Рисунок 28.19 – Режим HS мастера приемника

### Режим FS ведомого приемника

В этом режиме данные принимаются от мастера передатчика. Ведомый квитирует или не квитирует прием каждого байта.

После включения модуль I2C мониторит шину. При обнаружении состояния старта, модуль I2C переключается в режим ведомого приемника и начинает принимать семь бит адреса и бит направления передачи от мастера. Мастер передатчика может переключиться в режим ведомого приемника вследствие потери арбитража при передаче адреса.

После получения байта адреса ведомый сравнивает полученный адрес:

- по полю ADDR регистра ADDR, если установлен бит SAEN;
- со значением 0000\_000b (адрес общего вызова), если установлен бит GCMEN;
- со значением 0001\_100b (адрес отклика), если установлен бит SMBARE.

Квитирование приема производится, если принятый адрес совпал с «собственным» (запрограммированным пользователем) адресом общего вызова или адресом отклика. После обнаружения совпадения адреса и квитирования в поле MODE записывается соответствующий код и устанавливается флаг INT. Также, если разрешено битом INTEN, генерируется прерывание. Принятый байт (адрес и бит направления) переписывается в регистр SDA.

В зависимости от состояния бита направления, модуль I2C переходит в режим ведомого передатчика (если R/W# = «1») или остается в режиме ведомого приемника (R/W# = «0»).

После каждого принятого байта устанавливается флаг INT, указывающий на то, что необходимо прочитать данные из регистра SDA, а линия SCL удерживается в «0». После программного чтения регистра SDA флаг INT сбрасывается (записью единицы в бит CLRST), и линия SCL освобождается.

Установка битов SAEN и S10EN включает режим 10-битной адресации ведомого приемника. После обнаружения состояния старта ведомый последовательно принимает два байта, в которых содержится адрес.

Последовательность передачи бит в посылке при 10-битной адресации была рассмотрена ранее в подразделе 28.1 настоящего ТО.

Механизм распознавания адреса изложен в подразделе 28.2 настоящего ТО и показан на рисунке 28.13.

После корректного приема ведомым двух байтов и совпадения принятого адреса с собственным, байты сохраняются в регистре SDA и сдвиговом регистре, прием квитируется, устанавливается флаг INT, а в поле MODE записывается соответствующий код состояния – 10h или 17h.

Если включен механизм обнаружения ошибок, последний байт, принятый от мастера передатчика, будет байтом CRC. Если результат вычисления контрольной суммы не нулевой, устанавливается флаг ошибки PECFAULT и передача не квитируется. Программа пользователя должна «знать» о количестве передаваемых мастером байт и устанавливать бит PECNEXT перед чтением предпоследнего байта из регистра SDA и потом сбрасывать флаг INT. В результате будет аппаратно рассчитана контрольная сумма, и результат отправлен мастеру в момент передачи бита ACK. Если ошибок нет, будет выполнено квитирование (отправлен «0» в ответ на запрос ACK), если ошибки есть – неквитирование (отправлена «1» в ответ на запрос ACK).

Если ведомому приемнику нужно сообщить мастеру, что он не может более принимать данные, следует сначала установить бит ACK, а затем – бит CLRST (для сброса флага INT). Далее будет принят последний байт данных, который не будет квитирован (бит ACK = 1b), и установится флаг INT. После этого программа может прочитать последний полученный байт из регистра SDA и сбросить флаг INT, после чего модуль I2C освободит шину.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.20 представлено графическое пояснение к описанию режима.



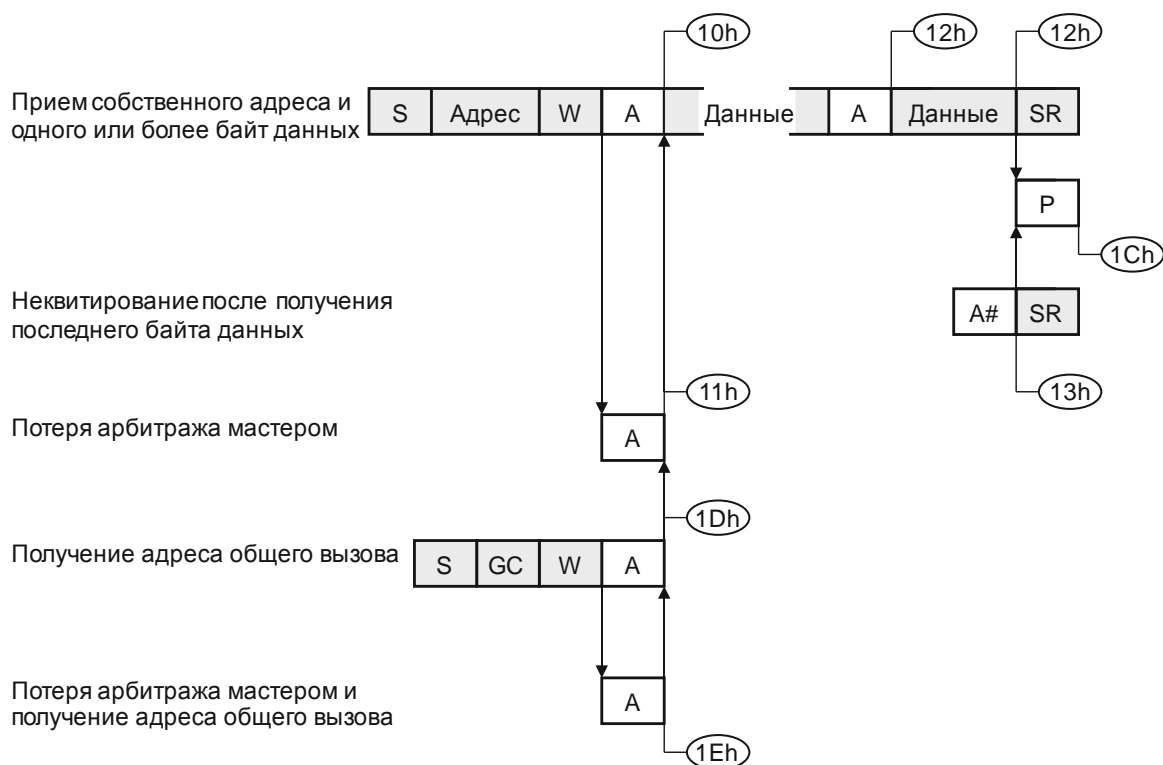


Рисунок 28.20 – Режим FS ведомого приемника



Рисунок 28.21 – Режим HS ведомого приемника

### Режим HS ведомого приемника

Включение режима происходит после получения валидного кода мастера (0000\_1xxxh). После передачи кода мастера формируется состояние повторного старта, а затем передается адрес ведомого с нулевым битом направления ( $R/W\# = \langle 0 \rangle$ ). После получения байта адреса ведомый проверяет его на совпадение (см. ранее «Режим FS ведомого приемника»).

Дополнительно можно обратиться к приложению Б.

На рисунке 28.21 представлено графическое пояснение к описанию режима.

### **Режим FS ведомого передатчика**

В этом режиме данные передаются от ведомого передатчика к мастеру приемника. Ведомый проверяет ответ мастера на бит ACK.

Переход в режим передатчика происходит из режима ведомого приемника. После получения собственного адреса и бита направления, равного единице ( $R/W\# = \langle 1 \rangle$ ), ведомый становится передатчиком. Флаг INT устанавливается, указывая на то, что в регистр SDA следует записать данные.

Пока установлен флаг INT, линия SCL удерживается в «0». После записи данных в регистр SDA следует сбросить флаг INT. После этого, по истечении времени, необходимого для установки данных на линии SDA, линия SCL освобождается, и данные начинают передаваться.

Передача данных аналогична передаче в режиме мастера передатчика. После каждого успешного приема байта устанавливается флаг INT, а в поле MODE записывается соответствующий код. Линия SCL удерживается в состоянии «0» до тех пор, пока флаг INT остается установленным. Флаг INT должен сбрасываться только после записи данных в регистр SDA. Каждый последующий байт должен записываться в регистр SDA до тех пор, пока в поле MODE не появится код 17h (состояние STDANA), указывающий на то, что мастер «не желает» далее принимать данные.

Вывод ведомого из режима передатчика осуществляется только мастером приемника. Мастер приемника должен не квитировать последний (согласно запланированному количеству) полученный байт данных. При обнаружении неквитирования переданных данных, модуль I2C переходит в режим безадресного ведомого, и в поле MODE записывается код 00h (состояние IDLE). Далее ведомый мониторит шину в ожидании состояния старта или повторного старта.

Для работы в режиме с 10-битной адресацией следует осуществить действия аналогичные, описанным для режима FS ведомого приемника.

Сначала модуль I2C переходит в режим ведомого приемника и получает 10-битный адрес. Если программно не требуется никаких действий, то флаг INT не устанавливается, линия SCL не удерживается в «0», и поле MODE содержит соответствующую информацию о состоянии. Далее (см. ранее «Формат передачи данных с 10-битной адресацией»), вслед за вторым байтом адреса, может последовать состояние повторного старта, и затем повторная передача первого байта адреса, с той лишь разницей, что бит направления содержит единицу ( $R/W\# = \langle 1 \rangle$ ). Таким образом, после приема трех байт, если принятый 10-битный адрес окажется «своим», установится флаг INT, и ведомый переключится в режим передатчика. В поле MODE запишется один из двух кодов – 14h или 15h.

Если включен механизм распознавания ошибок, то последний отправленный ведомым передатчиком байт будет байтом CRC. Программа должна «знать» количество байт, посылаемых в пакете данных, и после отправки всех байт устанавливать бит PECNEXT (вместо записи очередных данных в регистр SDA) для того, чтобы в регистр SDA записался байт контрольной суммы.

В модуле I2C поддерживается функция распознавания адреса отклика, который передается мастером шины ко всем ведомым. Ведомое устройство, получившее адрес отклика (0001\_100b), переключается в режим передатчика и начинает передавать свой собственный адрес (подробнее – см. «Формат передачи данных с 7-битной адресацией»).

Для включения функции распознавания адреса отклика следует установить бит SMBARE в регистре CTL0.

Модуль I2C реагирует на адрес отклика только при работе в режиме ведомого. В ответ на получение адреса отклика начать передачу адресов могут несколько ведомых. Ведомый, выигравший арбитраж, продолжает передачу, остальные – освобождают шину.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.22 представлено графическое пояснение к описанию режима.

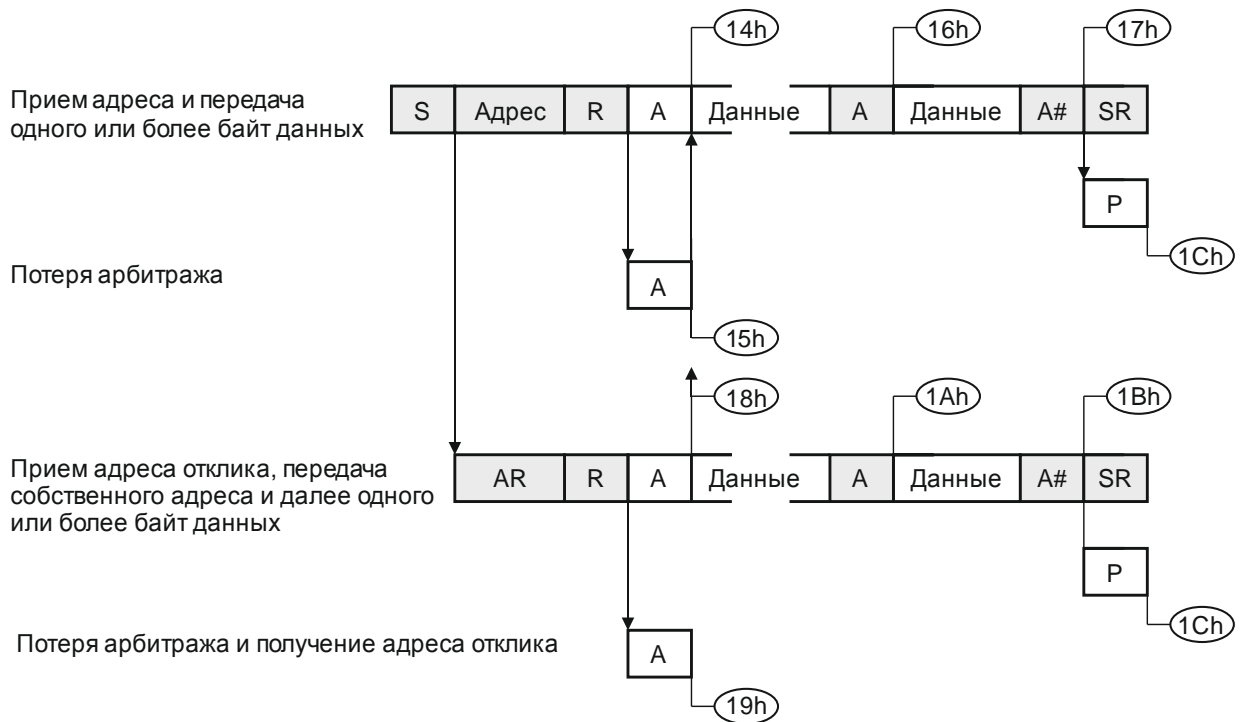


Рисунок 28.22 – Режим FS ведомого передатчика

### Режим HS ведомого передатчика

Модуль I2C переходит в режим HS ведомого после получения валидного кода мастера (0000\_1xxx). Далее следует состояние повторного старта и передача адреса ведомого с единичным битом направления ( $R/W\# = \langle 1 \rangle$ ). После этого ведомый переключается в режим HS ведомого передатчика. Функционирование в этом режиме в целом идентично режиму FS ведомого передатчика, с теми отличиями, что поддерживается более высокая скорость передачи, а значения кодов состояний (поле MODE) находятся в диапазоне 34h – 37h.

Дополнительно можно обратиться к приложению Б.

На рисунке 28.23 представлено графическое пояснение к описанию режима.

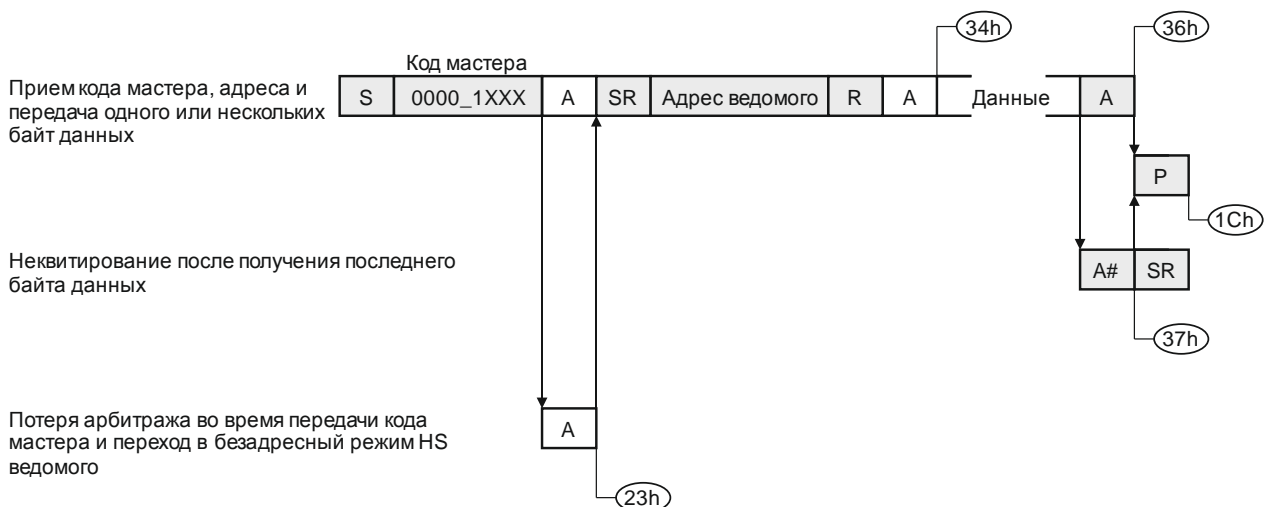


Рисунок 28.23 – Режим HS ведомого передатчика

### **Дополнительная информация о работе модуля**

1 Когда модуль I2C выключен, бит ВВ регистра CST очищен. Включение модуля в системе, с более чем одним мастером, может произойти в момент времени, когда по шине идет передача. Бит ВВ не сможет это показать. Во избежание создания ошибок на шине модуль I2C должен синхронизироваться с сигналами на шине прежде, чем сделать попытку стать мастером. Для этого следует дождаться момента, когда на шине не будет выявлена активность, т. е. периодически проверять бит ВВ через периоды времени, равные периоду ожидания на шине.

2 Бит ВВ позволяет мониторить шину и не допускать формирования ошибочных состояний старта в процессе передачи между другими устройствами на шине.

3 В некоторых случаях шина может «зависать» при активных (с нулевым уровнем) сигналах на линиях SDA и/или SCL. Источниками таких состояний могут быть необнаруженные ошибочные стартовые или стоповые состояния, сформировавшиеся в течение приема ведомых данных. Если считать, что причиной зависания явился модуль I2C, то возможны следующие два варианта развития событий:

а) если зависла линия SCL, ничего не будет происходить, а мастер, захвативший шину, должен освободить ее;

б) если зависла линия SDA, мастер должен освободить шину. Следует помнить, что в нормальном состоянии удерживать линию SCL может только текущий мастер шины. Последовательность действий для выхода из зависания следующая (при условии, что на шине только один мастер):

- выключить и включить модуль I2C для перевода его в режим безадресного ведомого;

- установить бит START для создания состояния старта;

- проверить, удерживается ли линия SDA в «0» (активное состояние) чтением бита TSDA регистра CST. Если линия активна, отправить одиночный импульс по линии SCL, установив бит TGSCS в регистре CST;

- проверить, что в поле MODE записан код 01b (состояние STDONE), который укажет на то, что состояние старта сформировано. Если нет, то повторять предыдущий и этот шаги до тех пор, пока линия SDA не освободится.

## 29 Контроллер интерфейса SPI

В состав микроконтроллера входят четыре идентичных контроллера интерфейса SPI, реализующих интерфейс последовательной синхронной связи в режиме ведущего (мастера) и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из трех протоколов фирм Motorola, National Semiconductor, Texas Instruments.

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает полнодуплексный обмен данными по четырехпроводной линии и программное задание фазы и полярности тактового сигнала.

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием 8-битных управляющих последовательностей.

Интерфейс SSI фирмы Texas Instruments обеспечивает полнодуплексный обмен данными по четырехпроводной линии и возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

Выбор интерфейса осуществляется посредством поля FRF регистра CR0.

В режиме мастера и в режиме ведомого устройства контроллер SPI обеспечивает:

- передачу данных, размещенных в буфере передатчика (восемь 16-разрядных ячеек);

- прием данных и размещение их в буфере приемника (восемь 16-разрядных ячеек).

Контроллер SPI формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов приемника и/или передатчика;
- переполнение буфера приемника;
- наличие данных в буфере приемника по истечении времени таймаута.

### Основные характеристики контроллера SPI:

- программное управление скоростью обмена;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буферов передатчика и приемника;
- поддержка прямого доступа к памяти (блок DMA).

### 29.1 Структура контроллера SPI

Упрощенная функциональная схема контроллера SPI с блоком синхронизации показана на рисунке 29.1.

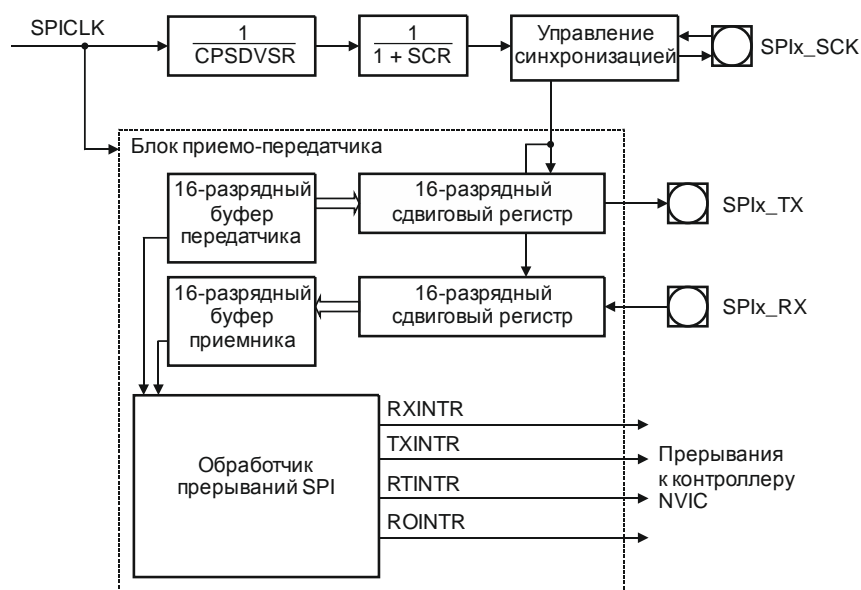


Рисунок 29.1 – Упрощенная функциональная схема контроллера SPI

### Синхронизация

Тактирование контроллера SPI осуществляется тактовым сигналом SPICLK, который формируется на основе одного из нескольких базовых синхросигналов (см. раздел 4).

Существует ограничение на соотношение между частотами тактовых сигналов PCLK и SPICLK:

$$f_{SPICLK} \leq f_{PCLK}. \quad (29.1)$$

В режиме мастера на основе сигнала SPICLK посредством двух последовательно стоящих делителей формируется сигнал тактирования передачи и приема данных с частотой  $F_{SPIx\_SCK}$ , которую можно вычислить по формуле

$$f_{SPIx\_SCK} = f_{SPICLK} / (CPSDVSR \times (1 + SCR)), \quad (29.2)$$

где  $F_{SPICLK}$  – частота входного синхросигнала SPICLK;

CPSDVSR – коэффициент первого делителя частоты (задается в регистре CPSR);

SCR – коэффициент второго делителя частоты (задается в регистре CR0).

Сформированный синхросигнал подается на вывод SPIx\_SCK (где x - номер контроллера 0-3) микроконтроллера и далее к подключенным внешним ведомым устройствам.

В режиме ведомого значения коэффициентов делителей не важны. Внешний синхросигнал подается на вывод SPIx\_SCK и тактирует прием и передачу данных.

Для корректной работы всегда должны соблюдаться условия:

- в режиме мастера для формируемого синхросигнала

$$f_{SPIx\_SCK} \leq f_{PCLK} / 2; \quad (29.3)$$

- в режиме ведомого для входящего внешнего синхросигнала

$$F_{SPIx\_SCK} \leq f_{PCLK} / 12. \quad (29.4)$$

### Буферы приема и передачи

Для хранения передаваемых и принятых данных в контроллере SPI имеются два 16-разрядных буфера, организованных по типу FIFO. Каждый буфер может хранить до восьми слов данных. Буфер для передаваемых данных доступен только для записи, а буфер принятых данных – только для чтения.

Данные для передачи записываются в буфер через регистр DR. Допускается заранее заполнить буфер или записывать в него данные в течение работы контроллера. Состояние буфера можно контролировать с помощью битов TNF и TFE регистра SR. Если контроллер выключен (сброшен бит SSE регистра CR1), то запись в регистр DR приведет к тому, что данные будут размещены в буфере и будут переданы после включения контроллера. Если контроллер включен и выбран режим мастера, то в случае отсутствия данных в буфере запись в регистр DR приведет к немедленному началу передачи. Если запись данных в регистр DR происходит во время текущей передачи, то данные размещаются в буфере.

Полученные данные автоматически сохраняются в буфере принятых данных. Извлечь данные из буфера возможно чтением регистра DR. Состояние буфера можно контролировать с помощью битов RFFF и RNE регистра SR.

Размер передаваемого кадра данных может быть от 4 до 16 бит, что задается полем DSS регистра CR0. Если выбран размер кадра менее 16 бит, данные выравниваются по правой границе; неиспользуемые биты игнорируются.

## 29.2 Интерфейс прямого доступа к памяти

Контроллер SPI имеет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром DMACR.

Сигналы для приема:

- RXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер приемника содержит, по меньшей мере, одно слово;

- RXDMABREQ – запрос блочного обмена данными, инициируется приемопередатчиком. Сигнал переходит в активное состояние в случае, если заполнение буфера приемника превысило заданный порог. Порог программируется индивидуально посредством поля RXIFLSEL регистра CR1;

- RXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен пакетный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в пакете.

Сигналы для передачи:

- TXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер передатчика содержит, по меньшей мере, одну свободную ячейку;

- TXDMABREQ – запрос блочного обмена данными, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если заполнение буфера передатчика ниже заданного порога. Порог программируется индивидуально посредством поля TXIFLSEL регистра CR1;

- TXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен пакетный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в пакете.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключаящими, 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.

Например, нужно принять 19 слов, а порог заполнения буфера установлен равным четырем. Тогда контроллер DMA осуществит четыре пакетных передачи блоков по четыре слова, а оставшиеся три слова – в ходе трех одиночных обменов, поскольку для них контроллер SPI не инициирует процедуру пакетного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом от контроллера DMA.

После снятия сигнала сброса приемопередатчик 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.

## 29.3 Функционирование

Приемопередающая логика модуля, управляющие регистры и FIFO по умолчанию находятся в сбросе. Снять сброс можно путём установки бита RSTDIS в соответствующем регистре SPICFG блока RCU.

Прежде чем разрешить работу битом SSE регистра CR1, следует сконфигурировать контроллер SPI посредством регистров CR0 и CR1, а также, если это необходимо, запрограммировать маски прерываний.

Динамическое изменение конфигурации устройства не допускается.

Для протокола SPI дополнительно задаются полярность и фаза сигнала (биты SPH и SPO регистра CR0).

После разрешения работы приемопередатчик готов к обмену данными с внешними устройствами по линиям SPIx\_TX (передача данных к внешнему устройству) и SPIx\_RX (прием данных от внешнего устройства).

В зависимости от режима работы сигнал на линии SPIx\_FSS используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора устройства в режиме ведомого (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

Во всех трех режимах SPI, Microwire и SSI синхросигнал SPIx\_SCK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SPIx\_SCK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

Установка бита MS регистра CR1 включает режим ведомого устройства. В этом режиме разрешение или запрещение передачи данных через выход SPIx\_TX контролируется битом SOD. На прием синхросигнала и данных состояние этого бита влияния не оказывает.

### Интерфейс SPI

Интерфейс SPI реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPIx\_SCK, две линии приема и передачи данных SPIx\_RX и SPIx\_TX, а также линию выбора устройства (для режима ведомого) SPIx\_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPIx\_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиной от 4 до 16 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Особенностью интерфейса SPI является то, что в нем реализована возможность задания полярности и фазы тактового сигнала. Бит SPO регистра CR0 задает полярность тактового сигнала, т. е. определяет, какой уровень сигнала будет удерживаться на линии SPIx\_SCK в то время, когда линия не активна.

Бит SPH задает фазу тактового сигнала. Фактически, он задает порядок считывания и выставления данных. По умолчанию, бит SPH сброшен и выставление данных на линиях SPIx\_TX и SPIx\_RX происходит по переднему фронту сигнала синхронизации, а выборка – по заднему.

Передним всегда считается тот фронт сигнала, который является началом передачи первого бита, см. рисунок 29.2.

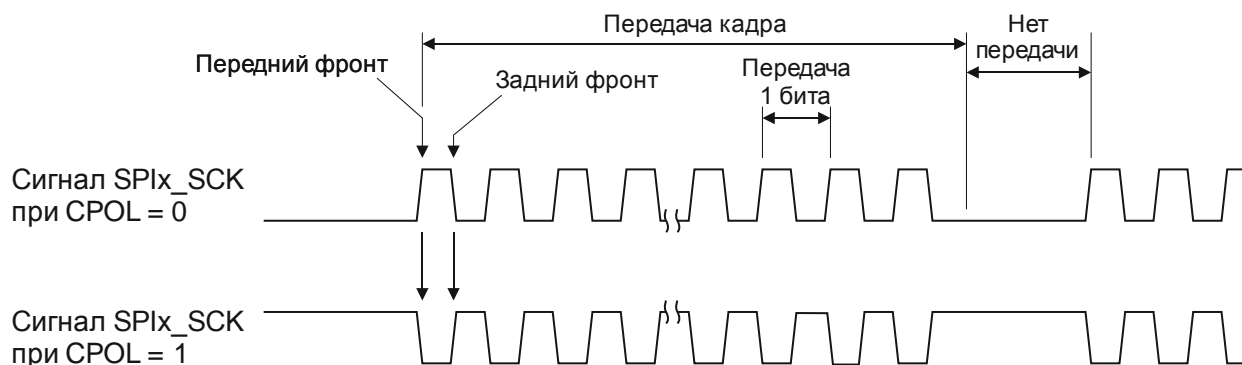


Рисунок 29.2 – Сигнал синхронизации SPIx\_SCK при разных состояниях бита CPOL



Комбинации битов  $SPO$  и  $SPH$  задают четыре режима обмена данными, см. рисунок 29.3.

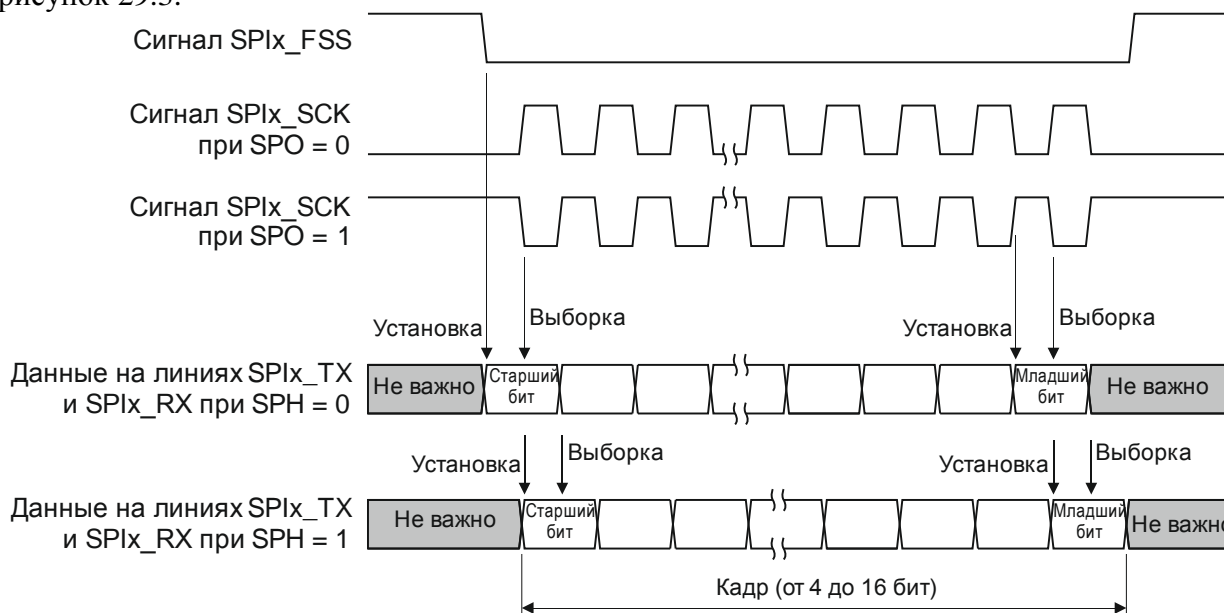


Рисунок 29.3 – Передача кадров данных в интерфейсе SPI

На рисунке 29.4 показано поведение сигналов при непрерывной передаче кадров данных.

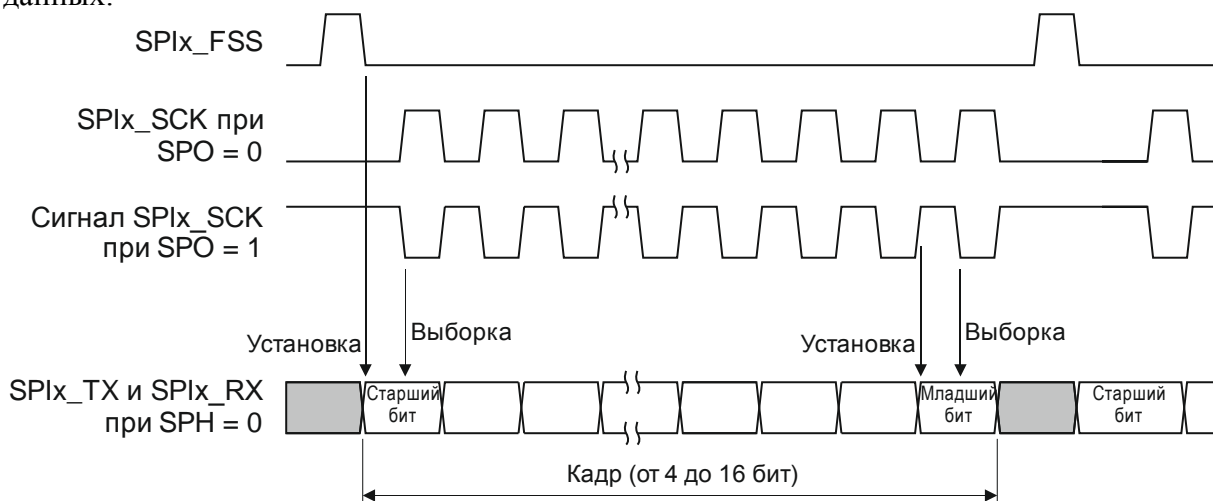


Рисунок 29.4 – Поведение сигналов при непрерывной передаче кадров данных

В режиме непрерывной передачи данных при условии  $SPH = 0$  на линии  $SPIx\_FSS$  должны формироваться импульсы между передачами кадров данных. Это связано с тем, что в этом режиме низкий уровень сигнала на линии  $SPIx\_FSS$  ведомого устройства блокирует запись в сдвиговый регистр. Поэтому мастер должен переводить линию  $SPIx\_FSS$  в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита кадра линия  $SPIx\_FSS$  переводится в состояние логической единицы по истечении одного такта сигнала  $SPIx\_SCK$ .

В режиме непрерывной передачи данных при условии  $SPH = 1$  низкий уровень сигнала на линии  $SPIx\_FSS$  не блокирует запись в сдвиговый регистр. Линия  $SPIx\_FSS$  может оставаться в состоянии нуля в течение передачи всех кадров. Только по окончании передачи линия может быть переведена в состояние логической единицы.

## Интерфейс Microwire

Интерфейс Microwire реализует полудуплексный режим передачи данных.

Включает линию синхронизации SPIx\_SCK, две линии приема и передачи данных SPIx\_RX и SPIx\_TX, а также линию выбора устройства (для режима ведомого) SPIx\_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPIx\_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиной от 4 до 16 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Перед началом передачи линия SPIx\_FSS переводится в низкое состояние.

Каждая передача начинается с передачи от мастера к ведомой 8-битной управляющей последовательности. В течение передачи этой последовательности приемник мастера не обрабатывает входящие данные. После того как управляющая последовательность передана и декодирована одним из ведомых устройств, этот ведомый выдерживает паузу в один такт синхросигнала и начинает передавать мастеру кадр данных, см. рисунок 29.5.

Выставление данных происходит по заднему фронту сигнала SPIx\_SCK, а считывание – по переднему.

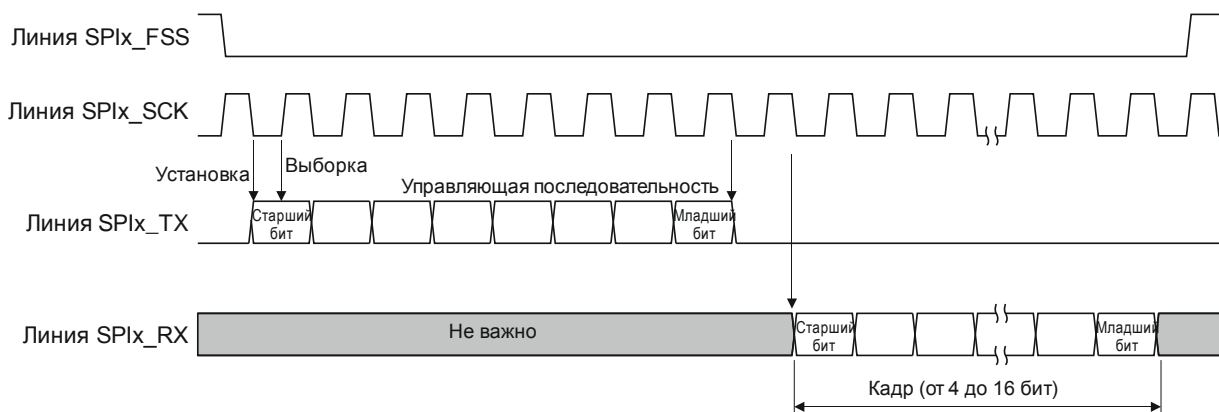


Рисунок 29.5 – Передача кадра данных в интерфейсе Microwire

По окончании приема данных линия SPIx\_FSS переводится в высокое состояние.

Примечание – В течение времени, когда передается управляющая последовательность и между передачами, линия SPIx\_RX может находиться в третьем состоянии.

В режиме непрерывной передачи начало и завершение передачи нескольких кадров данных аналогично передаче одного кадра. Линия SPIx\_FSS удерживается в нуле в течение всего сеанса передачи. По окончании передачи одного кадра данных начинается передача управляющей последовательности без паузы, см. рисунок 29.6.

Примечание – Буферы FIFO приема и передачи данных не очищаются автоматически, даже в случае запрещения работы сбросом бита SSE.

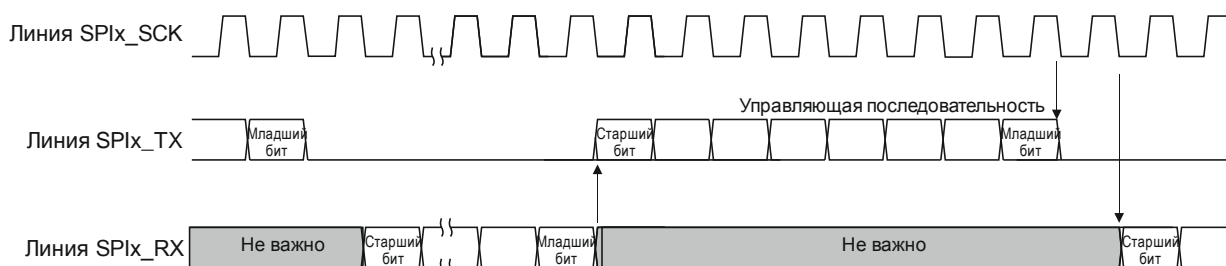


Рисунок 29.6 – Передача кадров данных в интерфейсе Microwire

## Интерфейс SSI

Интерфейс SSI реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPIx\_SCK, две линии приема и передачи данных SPIx\_RX и SPIx\_TX, а также линию выбора устройства SPIx\_FSS.

Перед началом передачи каждого кадра на линии SPIx\_FSS формируется импульс длительностью в один период сигнала SPIx\_SCK. Далее мастер и ведомый передают данные. Установка данных производится по переднему фронту синхросигнала, а выборка – по заднему, см. рисунок 29.7. Весь цикл передачи начинается сразу же после появления хотя бы одного элемента в буфере FIFO передатчика.

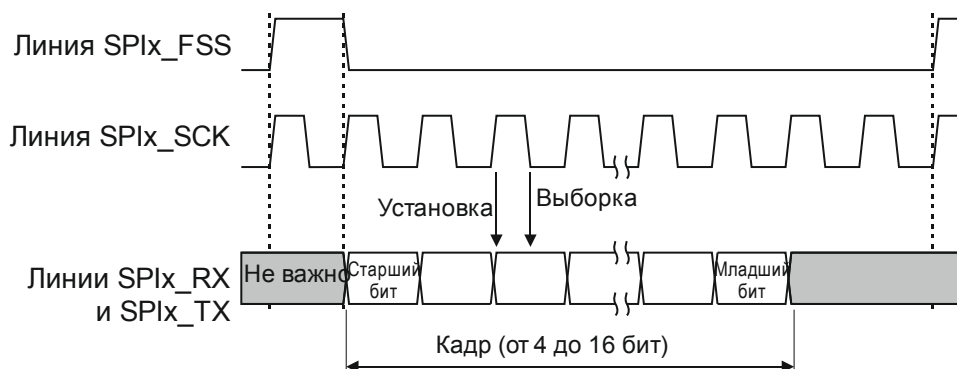


Рисунок 29.7 – Передача кадра данных в интерфейсе SSI

Режим непрерывной передачи кадров данных показан на рисунке 29.8.

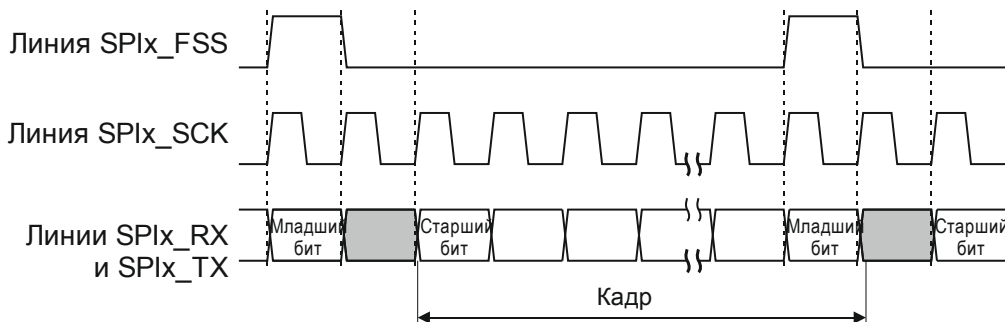


Рисунок 29.8 – Непрерывная передача кадров данных в интерфейсе SSI

## 29.4 Прерывания

Буфер приема и буфер передачи могут генерировать 4 независимых маскируемых запроса на прерывания:

- TXINTR – запрос на обслуживание буфера передатчика (опустошение буфера равно или ниже порога);
- RXINTR – запрос на обслуживание буфера приемника (заполнение буфера равно или выше порога);
- RTINTR – таймаут ожидания чтения данных из буфера приемника (буфер приемника не пуст и не было попыток обращения к нему в течение времени равного передаче 32 бит);
- RORINTR – переполнение буфера приемника.

Пороги программируются индивидуально для каждого буфера посредством полей TXIFLSEL и RXIFLSEL регистра CR1.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IMSC.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания). Сброс прерывания осуществляется программно путём установки соответствующего бита в регистре ICR.

На контроллере прерываний NVIC заведено 5 линий прерываний:

- SPI\_RX\_INT – запрос RXINTR;
- SPI\_TX\_INT – запрос TXINTR;
- SPI\_RO\_INT – запрос ROINTR;
- SPI\_RT\_INT – запрос RTINTR.
- SPI\_INT - логическое ИЛИ вышеперечисленных сигналов прерываний.

## 30 Контроллер интерфейса SpaceWire

### 30.1 Общие сведения

Устройство SpaceWire обеспечивает интерфейс между шиной АНВ и сетью SpaceWire. Данное устройство реализует стандарт SpaceWire (ECSS-E-ST-50-12C).

В состав микроконтроллера входят два идентичных контроллера интерфейса SpaceWire: SPWR<sub>x</sub> (где x равен 0 или 1).

Интерфейс SpaceWire настраивается через регистры, которые доступны через интерфейс APB. Данные передаются через канал DMA (Direct Memory Access), используя ведущий интерфейс АНВ.

Используется три тактовых домена: для интерфейса АНВ (HCLK), для передатчика (SPWRCLK) и для приемника.

### 30.2 Принцип работы

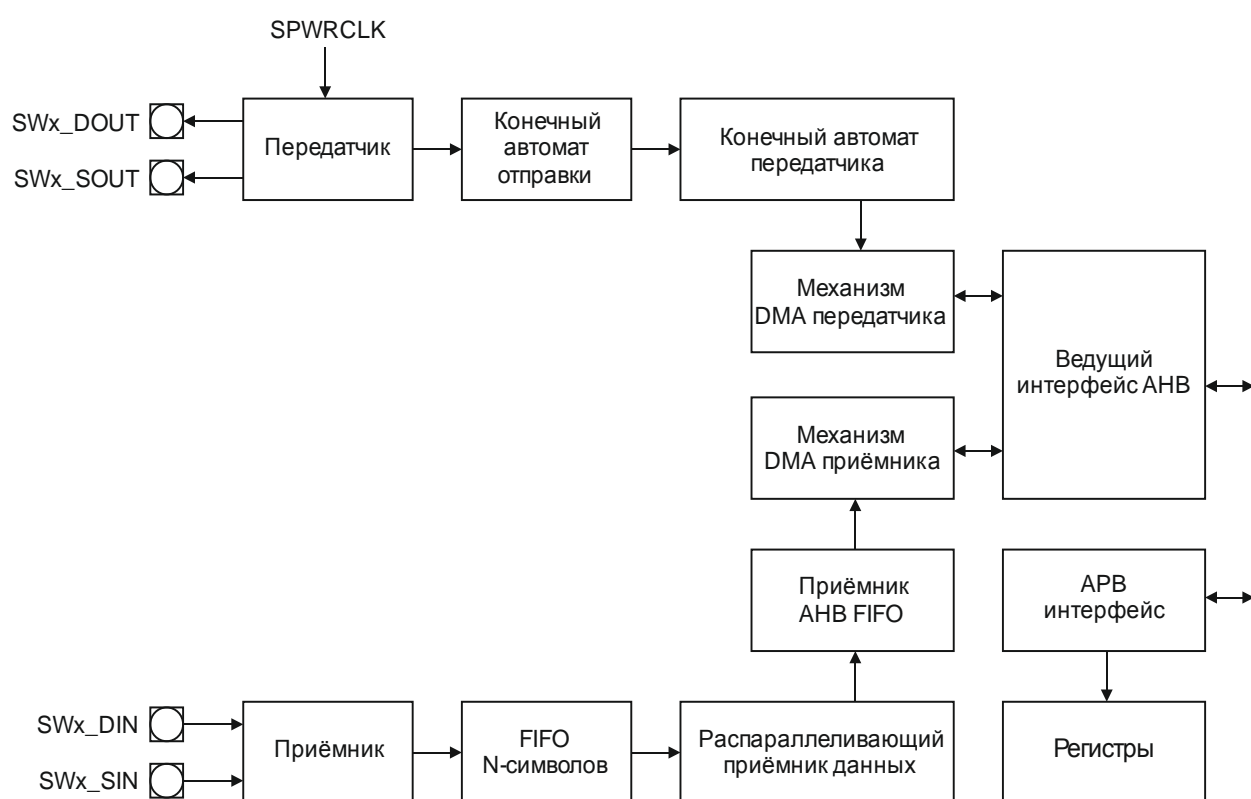


Рисунок 30.1 – Блок-схема интерфейса SpaceWire

Интерфейс связи состоит из приемника, передатчика и конечного автомата (КА) интерфейса связи. Они обеспечивают связь по сети SpaceWire. Интерфейс АМВА состоит из механизма DMA, интерфейса ведущего устройства АНВ и интерфейса APB. Интерфейс связи обеспечивает связь с механизмом DMA через FIFO. Данные буферы используются для передачи N-символов (N-Char, normal-characters, информационные символы) между доменами шины АМВА и SpaceWire в ходе приема и передачи.

Устройство принимает пакеты только с действительным адресом назначения в первом получаемом байте. Пакеты с другими адресами без каких-либо уведомлений будут проигнорированы (за исключением «неразборчивого» режима, который описан в конце подраздела «Приемный канал DMA»).

### 30.3 Интерфейс связи

Интерфейс связи обеспечивает связь по сети SpaceWire и состоит из передатчика, приемника, интерфейсов КА и FIFO.

#### **Конечный автомат интерфейса связи**

Конечный автомат (КА) контролирует интерфейс связи (для получения более подробной информации – см. стандарт SpaceWire). Обработка протокола нижнего уровня (сигнальный и символьный уровень стандарта SpaceWire) выполняется передатчиком и приёмником, в то время как КА занимается обработкой уровня обмена.

Интерфейс конечного автомата управляется через регистр управления CTRL. Линия связи может быть отключена с помощью бита запрета соединения (LD), который, в зависимости от текущего статуса, либо блокирует интерфейс связи, либо принудительно переводит его в состояние сброса после сбоя (ErrorReset). Если соединение разрешено, КА интерфейса связи запускается при условии, что установлен бит старта соединения (LS) или получен NULL-символ и установлен бит автозапуска (AS).

Текущий статус интерфейса связи определяет тип символов, доступных для передачи, который вместе с запросами от хост-интерфейсов определяет, что будет отправлено.

Передача системного времени (тайм-кодов) осуществляется, если КА находится в рабочем режиме (Run) и был получен соответствующий запрос от интерфейса синхронизации (подробнее об этом – в разделе «Распространение системного времени»)

Если интерфейс связи имеет статус соединение (Connecting) или рабочий режим (Run), то разрешена отправка символов управления потоком (FCT). FCT отправляются интерфейсом связи автоматически, когда это возможно. Данные действия производятся на основании текущего свободного места в приёмном буфере N-символов (normal characters, информационных символов) и определенного в стандарте максимального значения счетчика разрешений ожидающих передач данных – 56. Отправка FCT осуществляется до тех пор, пока в FIFO имеется место как минимум под 8 записей и значение счетчика разрешений, ожидающих передачи данных, меньше или равно 48.

N-символы отправляются в рабочем режиме (Run) при наличии разрешения на передачу. NULL-символы отправляются при отсутствии запросов на передачу других символов или если КА имеет статус, который не разрешает выполнения других передач.

Счетчик разрешений на передачу (входящие разрешения на передачу) автоматически увеличивается на 8 при получении FCT и уменьшается на 1 при передаче каждого N-символа. Полученные N-символы сохраняются в FIFO-буфере приёмника для последующей обработки интерфейсом DMA. Полученные коды синхронизации (тайм-коды) обрабатываются интерфейсом синхронизации.

#### **Передатчик**

Состояние КА, счетчики разрешений на передачу, запросы интерфейса синхронизации и интерфейса DMA используются для определения следующего передаваемого символа. Тип символа и сам символ (для N-символов и тайм-кодов) для передачи находятся в передатчике нижнего уровня, который расположен в отдельном тактовом домене.

Это необходимо в связи с тем, что обычно связь SpaceWire запускается с частотой, не соответствующей частоте тактирования системы. Поскольку передатчик нередко работает с большой частотой тактирования, для минимизации потребления энергии и обеспечения синхронизации вся логика, которая это позволяет, размещена в тактовом домене системы.

Логика передатчика в тактовом домене системы определяет, какой символ послать следующим, устанавливает надлежащие управляющие сигналы, а также обеспечивает низкоуровневую передачу (рисунок 30.2). Передатчик отправляет запрошенные символы и генерирует соответствующие биты четности и управления. До тех пор, пока передача разрешена, при отсутствии запросов домена хост-системы отправляются NULL. Большинство сигнальных и символьных уровней стандарта SpaceWire обрабатывается в передатчике. Для сигналов данных и строба необходим внешний LVDS драйвер. Выходы сконфигурированы с одинарной скоростью передачи данных (SDR).

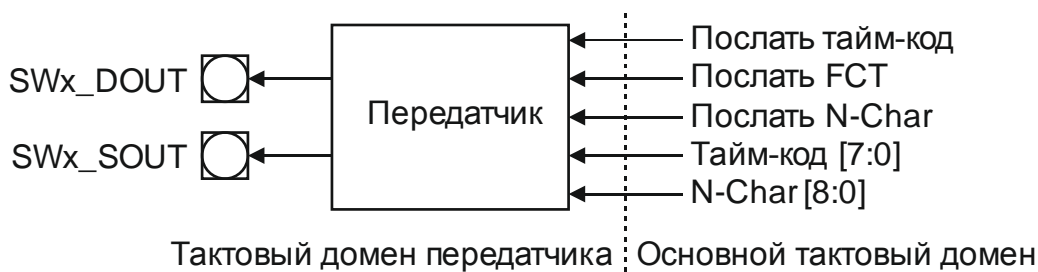


Рисунок 30.2 – Схема передатчика с интерфейсом связи

КА передатчика считывает N-символы для передачи из FIFO передатчика. Интерфейс DMA передает сведения о длине пакета, при этом КА передатчика при необходимости добавляет значения EOP/EEP. По окончании обработки пакета интерфейс DMA получает соответствующее уведомление и выдается новое значение длины пакета.

### Приемник

Приёмник обнаруживает подключения других узлов и получает символы в виде битового потока, восстановленного из сигналов данных и строба. Приёмник расположен в отдельном тактовом домене, который работает от тактового сигнала, генерируемого из сигналов данных и строба.

Приёмник включается при выходе интерфейса связи из состояния сброса после сбоя (ErrorReset). После приема NULL-символа приёмник может начать принимать любые символы. Он обнаруживает ошибки по четности, ошибки перехода и ошибки счетчика разрешений на передачу, которые приводят к тому, что интерфейс связи входит в состояние сброса. Рассоединения обрабатываются в интерфейсе связи в тактовом домене передатчика, так как тактирование приёмника при такой ситуации не доступно.

Принятые символы отмечаются как принадлежащие хост-домену, данные представлены в параллельном виде. Интерфейс хост-домена показан на рисунке 30.3.

L-символы (L-Char, link-символы, каналные символы) обрабатываются автоматически интерфейсом связи хост-домена, все N-символы сохраняются в FIFO приёмника для последующей обработки. При приеме двух или более последовательных EOP/EEP все, кроме первого из них, отбрасываются.

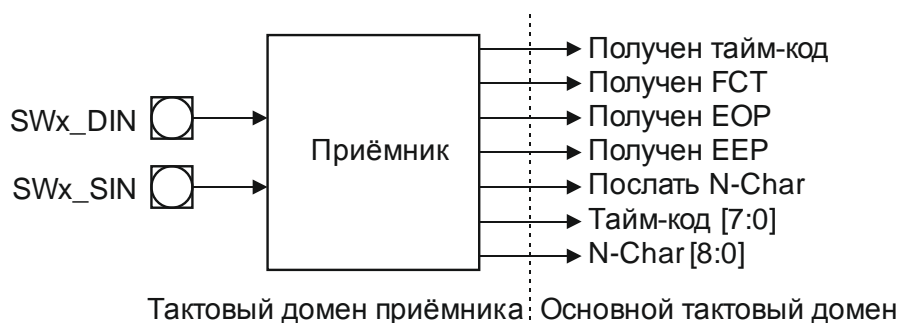


Рисунок 30.3 – Схема приемника с интерфейсом связи

### **Установка скорости канала связи**

Поле CKDSTRT регистра CLKDIV определяет скоростной режим во время инициализации (все состояния вплоть до состояния «Connecting» включительно). Этот регистр также используется для вычисления тайм-аута интерфейса КА (6,4 мкс и 12,8 мкс, как определено в стандарте SpaceWire). Значение поля CKDSTRT всегда должно быть установлено так, чтобы во время инициализации выдерживалась частота в 10 МГц. При выполнении данного условия величина тайм-аута рассоединения также будет рассчитана корректно. Частота  $F_{SPWR}$ , изначально поступающая на блок SpaceWire, выбирается в регистре SPWRCFG блока RCU.

Для того чтобы рассчитать требуемое значение CKDSTRT, следует воспользоваться формулой

$$CKDSTRT = (F_{SPWR} / 10) - 1$$

Скорость передачи в рабочем режиме  $F_{RUN}$  задается с помощью соответствующего коэффициента делителя CKDRUN(поле регистра CLKDIV ) и рассчитывается по следующей формуле:

$$F_{RUN} = F_{SPWR} / (CKDRUN + 1)$$

### **30.4 Распространение системного времени (тайм-кодов)**

Тайм-коды являются кодами управления, которые состоят из двух флагов контроля (биты 7-6) и значений системного времени (биты 5-0). Они используются для распространения времени в сети SpaceWire. Текущее значение времени (значение последних полученных или переданных тайм-кодов) и управляющие флаги могут быть прочитаны из регистра TIMECODE.

#### **Прием тайм-кодов**

Когда управляющий код принят, а также флаги контроля (биты 7-6) имеют значение «00» или сброшен бит фильтра флагов контроля тайм-кодов (TF) регистра управления CTRL, тогда принятый управляющий код считается тайм-кодом.

Если разрешен прием тайм-кодов (бит TF установлен в 1), то полученное значение времени сохраняется в поле TCNT регистра системного времени TIMECODE. Если полученное значение времени равно TCNT+1 (по модулю 64), тогда тайм-код считается действительным.

Когда принят действительный тайм-код, в дополнение к обновлению значения времени, полученные контрольные флаги сохраняются в поле TCTRL. Так же, когда принят действительный тайм-код, устанавливается бит TO регистра статуса STAT. Если при этом установлены биты разрешения прерываний (IE) и прерывания при получении тайм-кода (TQ) регистра управления CTRL установлены, то генерируется прерывание.

#### **Передача тайм-кодов**

Для того, чтобы передать тайм-код, требуется установить бит входного отсчета (TI) регистра управления CTRL. Когда бит установлен, текущее значение времени (поле TCNT регистра системного времени TIMECODE) инкрементируется (по модулю 64) и тайм-код, состоящий из нового значения времени и текущих флагов контроля (поле TC регистра системного времени TIMECODE), отсылается. Бит TI будет оставаться в установленном состоянии до тех пор, пока тайм-код будет передаваться. Данная операция предварительно требует установки бита разрешения передачи тайм-кодов (TT) регистра управления CTRL. Если передача тайм-кодов запрещена, то запись в бит TI не возымеет эффекта.



Обратите внимание, что интерфейс связи должен быть в состоянии «Run» для того, чтобы иметь возможность отправить тайм-код.

### **Аппаратный таймер временных отсчетов**

Для упрощения процесса формирования входных отсчетов предусмотрены два внешних аппаратных таймера, которые работают на системной частоте АНВ шины. Каждый из таймеров жестко связан со своим блоком SpaceWire. Режим счета – вниз, биты TOEN и T1EN разрешают работу соответствующего таймера и регистры RELO и REL1, которые соответственно задают значение, с которого счетчик начнет считать после переполнения. Сигналы переполнения идут на вход блоков SpaceWire и выполняют ту же функцию что и программная установка бита входного отсчета TI.

## **30.5 Приёмный канал DMA**

### **Сравнение адреса**

Путь полученного пакета зависит от его адреса и разрешения/запрещения канала. Когда FIFO N-символов приёмника содержит один или более символов, они считываются с помощью механизма DMA приёмника. Первый символ интерпретируется как логический адрес, пакет будет сохранен с использованием DMA, если его адрес будет соответствовать адресу канала DMA. Указатель на область памяти, в которой сохранен полный пакет, включая адрес и идентификатор протокола, но исключая EOP/EER, и хранится в дескрипторе канала.

Регистр адреса DEFADDR состоит из поля масок и поля адреса. Только биты адреса, содержащие значение 0 в соответствующем поле маски сравниваются. Таким образом, канал DMA может принять ряд адресов.

### **Основные функциональные возможности канала**

Работа приёмника базируется на последовательно расположенных в памяти дескрипторах, которые содержат указатели на буферы, в которые должны быть сохранены пакеты. Дескриптор считывается из области дескрипторов канала DMA, и принятый пакет сохраняется в область памяти, указываемую в дескрипторе. В заключении статус сохраняется в текущий дескриптор, и производится автоматический переход к очередному дескриптору.

### **Подготовка устройства для приёма**

Перед приемом некоторые регистры должны быть инициализированы. Первоначально, перед отправкой данных, необходимо перевести интерфейс связи в рабочий режим (Run). Канал DMA имеет регистр максимального размера принимаемого пакета - DMARXLEN, который задает максимальный размер пакета для получения каналом. Большие пакеты усекаются, и данные, превысившие обозначенный предел, сбрасываются. Если это произошло, то данный факт будет отражен в поле статуса дескриптора. Минимальное значение поля максимального размера приемника – 4, при этом значение можно увеличивать с шагом в четыре байта до максимального значения 33 554 428. Если максимальный размер установлен в «0», нормальная работа приемника не гарантируется.

Также должен быть установлен либо регистр адреса по умолчанию - DEFADDR, либо регистр адреса канала DMA - DMAADDR. Бит разрешения адреса (EN) регистра управления/статуса канала DMA (DMACSR) определяет, какая из пар адрес/маска адреса будет использована для сравнения с адресом принимаемого пакета. Таблица дескрипторов и регистр управления DMACSR также должны быть инициализированы.

### **Установка адреса таблицы дескрипторов**

Устройство считывает дескрипторы из области в памяти, на которую указывает регистр адреса таблицы дескрипторов приёмника DMARDTADDR. Регистр состоит из поля базового адреса и селектора дескрипторов. Базовый адрес указывает на начало области и должен быть выровнен относительно размера таблицы дескрипторов. Таблица рассчитана на 128 записей.

Поле селектора дескрипторов указывает на отдельные дескрипторы и увеличивается на единицу после использования текущего дескриптора. При достижении значением поля верхней границы, он автоматически переводится на начало. Так же подобный переход может быть осуществлен до достижения верхнего предела, путем установки бита перехода на начало в дескрипторе. Идея заключается в том, что поле селектора должно быть инициализировано значением «0» (начало области дескрипторов), но в него может быть записано и другое значение, выровненное до 8 байт для запуска где-либо в середине области. При достижении селектором верхней границы он все равно будет переходить в начало области дескрипторов.

При необходимости использования новой таблицы дескрипторов в первую очередь следует сбросить бит разрешения приёмника (RE) регистра управления/статуса канала DMA (DMACSR). Если бит RX (осуществляется приём) данного регистра сброшен, то обновление регистра адреса таблицы дескрипторов приёмника (DMARDTADDR) безопасно. После завершения обновления и установки бита доступности нового дескриптора приёмника (RD) можно снова установить бит разрешения приёмника (RE).

### **Разрешение на использование дескрипторов**

Как указывалось выше, для обеспечения приема необходимо разрешить использование одного или более дескрипторов. Размер каждого дескриптора – 8 байт (см. в таблицах ниже). Дескрипторы должны быть записаны в область памяти, на которую указывает регистр адреса таблицы дескрипторов приёмника (DMARDTADDR). В случае добавления новых указателей, они должны размещаться после предыдущего, записанного в область. В противном случае они не учитываются.

Активация дескриптора осуществляется путем установки адреса указателя на область памяти, отведенную для сохранения данных, после чего устанавливается бит разрешения на использование (EN). Бит WR устанавливается для сброса поля селектора дескриптора (SEL) регистра адреса таблицы дескрипторов приёмника (DMARDTADDR) после завершения приема по данному дескриптору. Бит IE должен быть установлен, если после завершения приема требуется прерывание. Бит разрешения прерывания (RI) регистра управления/статуса канала DMA (DMACSR) так же должен быть установлен для генерации прерывания.

Адрес пакета дескриптора не обязан быть выровнен по границе слова. Любое количество байт может быть получено по любому возможному адресу пакета без какой-либо избыточной перезаписи. Отдельно взятый дескриптор представлен в таблицах 30.1 и 30.2.

Таблица 30.1 – Слово «0» дескриптора приема (смещение адреса 0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TR	0	0	EP	IE	WR	EN	PKTLEN								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKTLEN															
Поле		Биты	Описание												
TR		31	Усечённый – пакет усечен ввиду превышения заданного максимального размера												
EP		28	Завершение EEP – данный пакет завершен с символом «пакет завершен с ошибкой» (Error End of Packet)												
IE		27	Разрешение прерывания – если установлено, после приема пакета генерируется прерывание, если в регистре управления/статуса канала DMA (DMACSR) установлен бит разрешения прерывания при приеме (RI)												
WR		26	Переход на начало – если установлено, следующий используемый дескриптор будет первым в таблице дескрипторов (располагаться по базовому адресу). В противном случае указатель на дескриптор увеличивается на 0x08 для обеспечения использования следующего дескриптора, расположенного далее в памяти. При достижении границы таблицы дескрипторов, указанной в предыдущем подпункте, указатель автоматически переходит обратно на базовый адрес												
EN		25	Разрешение на использование – устанавливается для активации дескриптора. Это означает, что дескриптор содержит действительные управляющие значения и область памяти, на которую ссылается поле адреса пакета и может использоваться для сохранения пакета												
PKTLEN		24-0	Размер пакета – количество байт, принятых в данный буфер												
–		30,29	Зарезервировано												

Таблица 30.2 – Слово «1» дескриптора приема (смещение адреса 0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKTADDR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKTADDR															
Поле		Биты	Описание												
PKTADDR		31-0	Адрес, указывающий на буфер для сохранения принятого пакета												

### Установка регистра управления DMA

Последний этап для приема пакетов – установка регистра управления: необходимо разрешить приёмник установкой бита RE в регистре управления/статуса канала DMA (DMACSR). Это можно сделать в любой момент: пока бит не установлен, операция не запускается. Бит доступности дескриптора приёмника (RD) также устанавливается для информирования о наличии новых активированных дескрипторов. Это действие всегда должно выполняться после активации дескрипторов; в противном случае система может не обнаружить новые дескрипторы. Остальные дескрипторы могут быть активированы после начала приема при разрешении дескриптора на использование и записи бита RD регистра управления/статуса канала DMA (DMACSR). После установки этих бит приём начинается непосредственно при поступлении данных.

### Влияние управляющих бит во время приема

Когда приёмник не разрешен, все пакеты, отправленные в канал DMA, игнорируются. Если приёмник разрешен и адрес попадает в диапазон принимаемых адресов, то осуществляется переход в следующее состояние с проверкой бита доступности дескриптора приёмника (RD). Данный бит указывает на наличие активированных дескрипторов и должен устанавливаться внешним приложением, использующим канал DMA, при каждом разрешении на использование дескрипторов, как обозначено выше. Если бит RD и бит NS регистра управления/статуса канала DMA (DMACSR) сброшены, то принимаемые пакеты игнорируются. Если бит NS установлен, система ожидает установки бита RD и символы остаются в FIFO N-символов в течение этого времени. Если FIFO заполняется, дальнейшая передача N-символов запрещается остановкой передачи FCT.

После установки бита RD считывается следующий дескриптор; если он активирован, пакет принимается в буфер. Если считанный указатель не разрешен, то бит RD сбрасывается, далее в зависимости от значения NS пакет может быть сброшен или оставлен в буфере.

Приёмник можно отключить в любой момент, при этом прием текущего пакета будет завершён. Бит доступности дескриптора приёмника (RD) также может быть сброшен в любой момент. При этом текущие приемы не затрагиваются, но новые дескрипторы не будут считаны до повторной установки данного бита. RD сбрасывается устройством при считывании не активированного дескриптора.

### Биты статуса

После завершения приема пакета бит разрешения на использование в текущем дескрипторе сбрасывается и при сброшенном разрешении биты статуса остаются действительными, а количество принятых байт указывается в поле размера. **Ошибка! источник ссылки не найден.** (DMACSR) содержит бит статуса (PR), который устанавливается при каждом приеме пакета. Также устройство может генерировать прерывание по наступлению данного события.

### Обработка ошибок

Если прием пакета должен быть прерван из-за перегрузки в сети, рекомендуется установить бит запрещения соединения (LD) регистра управления (CTRL). К сожалению, это приведет к усечению передаваемого пакета, но это единственное безопасное решение, поскольку прием пакета – пассивная операция, зависящая от передатчика на другом конце. Также можно установить бит сброса узла (RS), но это не совсем подходящий путь, так как непереданные символы остаются в узле передатчика. Следующий символ (расположенный где-то в середине пакета) будет интерпретирован как адрес узла, что может привести к тому, что пакет будет сброшен (не со 100 % уверенностью). Обычно это действие выполняется в случае приостановки приема, вызванной тем, что передатчик

больше не предоставляет данные. Сброс соединения не устраняет подобную перегруженность.

При возникновении ошибки АНВ в ходе приема текущий пакет сбрасывается вплоть до следующего символа EEP/EOP, после чего активный канал отключается и приёмник переходит в режим ожидания. Для обозначения данного состояния в регистре управления/статуса канала DMA (DMACSR) устанавливается бит RA.

#### **«Неразборчивый» режим**

В устройстве поддерживается «неразборчивый» режим, в котором все принятые данные сохраняются в канале DMA, независимо от адреса узла и возможных преждевременных EOP/EEP. Это означает, что все принятые N-символы, исключая EOP/EEP, сохраняются через канал DMA. Тем не менее, проверка длины принимаемой последовательности данных остается, при этом пакеты, превышающие размер, указанный в регистре максимального размера принимаемого пакета (DMARXLEN), отсекаются.

### **30.6 Передающий канал DMA**

#### **Базовая функциональность**

DMA передатчика считывает данные с шины АНВ и сохраняет их в FIFO передатчика для передачи по сети SpaceWire. Данная функциональность базируется на дескрипторах такого же типа, что и для приёмника; таблица дескрипторов имеет такое же выравнивание по адресу и ограничения по размеру (64 дескриптора). При активациях новых дескрипторов система считывает их и передает данные предписанного размера.

#### **Подготовка устройства для передачи**

Перед передачей должны быть выполнены четыре этапа. Сначала необходимо разрешить и запустить интерфейс связи записью соответствующего значения в регистр управления (CTRL). После этого адрес таблицы дескрипторов следует записать в регистр адреса таблицы дескрипторов передатчика (DMATDTADDR), при этом в таблице должны быть разрешены на использование один или более дескрипторов. В заключение в регистр управления/статуса канала DMA (DMACSR) следует записать «1» в бит TE – после этого запускается передача. Более подробное описание – в последующих подпунктах.

#### **Разрешение на использование дескрипторов**

Устройство считывает дескрипторы из области в памяти, на которую указывает регистр адреса таблицы дескрипторов передатчика (DMATDTADDR). Регистр состоит из поля базового адреса и селектора дескрипторов. Базовый адрес указывает на начало области и должен быть выровнен относительно размера таблицы дескрипторов. Каждый дескриптор занимает 16 байт. Всего таблица рассчитана на 64 дескриптора.

Для передачи пакетов один или несколько дескрипторов должны быть инициализированы в памяти. Для этого следует установить количество байт для передачи и указатель на область данных. Две различных длины и два поля адреса в дескрипторах передатчика используются потому, что указатели на заголовок и данные разнесены. Если длина поля равна нулю, то соответствующая часть пакета будет пропущена, если оба поля равны нулю, то пакет не отправляется. Максимальная длина заголовка – 255 байт, максимальная длина данных 16 Мбайт – 1. Когда указатель и поле размера установлены, для активации дескриптора необходимо установить бит разрешения на использование дескриптора (EN) в самом дескрипторе. Данный бит устанавливается в последнюю очередь. Другие биты управления также должны быть установлены перед активацией дескриптора.

Размер дескриптора передатчика – 16 байт, максимальное количество в одной таблице – 64. Различные поля дескриптора и смещение памяти представлены в таблицах ниже.

### Запуск передач

Чтобы система запустила передачу, после инициализации дескрипторов в регистре управления/статуса канала DMA (DMACSR) должен быть установлен бит разрешения передачи (TE). Новые дескрипторы можно активировать в таблице уже в процессе работы (при запущенной передаче). При каждом добавлении набора дескрипторов должен быть установлен бит разрешения передачи, т.к. при каждом обнаружении устройством не активированного дескриптора, бит TE сбрасывается. Отдельно взятый дескриптор представлен в таблицах 30.3-30.6.

Таблица 30.3 – Слово «0» дескриптора передатчика (смещение адреса 0x0)

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															
LE	IE	WR	EN	-				HEADERLEN							
Поле	Биты	Описание													
LE	15	Ошибка соединения – Ошибка соединения в ходе передачи пакета													
IE	14	Разрешение прерывания – Если установлено, генерируется прерывание после передачи пакета при условии, что бит разрешения прерывания передатчика (RI) регистра управления/статуса канала DMA (DMACSR) также установлен													
WR	13	Переход на начало – Если установлено, селектор дескриптора сбрасывается, соответственно следующий считанный дескриптор будет первым в таблице дескрипторов (располагаться по базовому адресу). В противном случае, указатель на дескриптор увеличивается на 0x10 для обеспечения использования следующего дескриптора, расположенного далее в памяти													
EN	12	Разрешение – Разрешение дескриптора передатчика. Если установлены все поля управления (адреса, размера, перехода на начало и CRC), данный бит следует установить. При установленном бите дескриптор не должен быть изменён во избежание нарушения передачи. SpaceWire сбрасывает данный бит после завершения передачи													
HEADERLEN	7-0	Размер заголовка – Размер заголовка в байтах. Если сброшено, заголовок пропускается													
-	31-16, 11-8	Зарезервировано													

Таблица 30.4 – Слово «1» дескриптора передатчика (смещение адреса 0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEADERADDR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEADERADDR															
Поле		Биты	Описание												
HEADERADDR		31–0	Адрес заголовка – Адрес, по которому расположен заголовок пакета. Выравнивание до границы слова не обязательно												

Таблица 30.5 – Слово «2» дескриптора передатчика (смещение адреса 0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-								DATALEN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATALEN															
Поле		Биты	Описание												
DATALEN		23–0	Размер данных – Размер данных пакета. Если сброшено, данные не отправляются. Если размеры данных и заголовка сброшены, пакет не отправляется												
-		31-24	Зарезервировано												

Таблица 30.6 – Слово «3» дескриптора передатчика (смещение адреса 0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAADDR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAADDR															
Поле		Биты	Описание												
DATAADDR		31–0	Адрес данных – Адрес, по которому считываются данные. Выравнивание до границы слова необязательно												

### **Процесс передачи**

После установки бита TE регистра управления/статуса канала DMA (DMACSR) сразу же начинается считывание дескриптора. Количество обозначенных байтов считывается и передается. После завершения передачи статус записывается в первое слово дескриптора, при этом в регистре управления/статуса канала DMA устанавливается бит отправленного пакета (PS). Если было запрошено прерывание, то оно генерируется. После этого считывается новый дескриптор; если он активирован, запускается новая передача. В противном случае, бит разрешения передачи (TE) сбрасывается (процесс останавливается до повторной установки данного бита).

### **Регистр адреса таблицы дескрипторов**

Внутренний указатель, который используется для хранения текущего положения в таблице дескрипторов, можно считывать и записывать через интерфейс APB. Этот указатель устанавливается в ноль при выполнении сброса и инкрементируется каждый раз при использовании очередного дескриптора. Переход на начало выполняется автоматически при достижении границы таблицы дескрипторов или раньше, если в текущем дескрипторе установлен соответствующий бит.

Регистр адреса таблицы дескрипторов передатчика (DMATDTADDR) может быть обновлен новой таблицей в любое время, если не ведется передача. Передача не является активной, если бит разрешения передачи (TE) равен нулю и полная таблица была отправлена, или если таблица прервана. Если таблица прервана, необходимо подождать сброса бита TE перед тем, как обновлять указатель на таблицу.

### **Обработка ошибок: прерванная передача**

Регистр управления/статуса канала DMA (DMACSR) содержит бит «прервать передачу» (AT), установка которого вызывает прекращение текущей передачи, при этом пакет усекается и вставляется EEP. Это может потребоваться, если только требуется прерывание пакета в случае перегрузки сети SpaceWire. В случае прерывания пакета в дескрипторе устанавливается бит LE. Бит регистра разрешения передачи (TE) регистра управления/статуса канала DMA (DMACSR) также сбрасывается; новые передачи не выполняются до следующего разрешения передатчика.

### **Обработка ошибок: ошибка АНВ**

При обнаружении ошибки АНВ в ходе передачи активный канал DMA отключается и передатчик переходит в режим ожидания. В регистр управления/статуса канала DMA (DMACSR) устанавливается соответствующий бит, характеризующий ошибку, и если разрешено, генерируется прерывание. Дальнейшая обработка ошибок зависит от статуса передатчика механизма DMA на момент возникновения ошибки АНВ. Если дескриптор был считан, а передача пакета еще не началась, то никаких дополнительных действий больше не требуется.

Если ошибка АНВ возникла в ходе передачи пакета, пакет усекается и вставляется EEP. Если ошибка возникла в ходе записи статуса в дескриптор, пакет успешно передан, но дескриптор не перезаписывается и остается активированным (это также означает, что биты ошибки в дескрипторе не устанавливаются для случая ошибок на шине АНВ).

Пользователь канала должен исправить ошибку АНВ и повторно активировать канал. Другие передачи АНВ из блока (передатчика или приемника), который был задействован в момент обнаружения ошибки АНВ, не выполняются до сброса состояния ошибки и повторного разрешения данного блока.

### **Обработка ошибок: ошибка соединения**

При обнаружении ошибки связи в ходе передачи оставшаяся часть пакета отбраковывается вплоть до следующего EOP/EEP включительно. После этого статус



незамедлительно записывается в дескриптор (бит LE установлен) и указатель на дескриптор инкрементируется. Соединение разрывается при возникновении ошибки соединения, но система автоматически пытается снова восстановить связь при условии, что бит запуска связи (LS) регистра управления (CTRL) установлен и бит отключения связи (LD) сброшен. Если бит LE в регистре управления/статуса канала DMA (DMACSR) не установлен, механизм DMA передатчика будет ожидать осуществления перехода в рабочий режим (Run), после чего немедленно начинается новая передача при условии, что пакеты находятся в режиме ожидания. Если бит LE в регистре управления/статуса канала DMA (DMACSR) установлен, то передатчик будет отключен при возникновении ошибки соединения, произошедшей в ходе передачи текущего пакета. В этом случае пакеты не будут передаваться до повторного разрешения передатчика.

## **30.7 Аппаратные особенности**

### **Особенности системы тактирования**

Тактовая частота передатчика генерируется на основе сигнала SPWRCLK с использованием делителя. Отдельный тактовый вход используется для того, чтобы можно было вести передачу на частотах значительно превышающих системную. Частота передатчика должна составлять 10 МГц при установлении соединения, в рабочем режиме может быть использовано любое значение, превышающее 2 МГц.

Регистр тактового делителя (CLKDIV) содержит два поля. Одно из них определяет коэффициент деления в процессе инициализации соединения, второе задает тактовую частоту в рабочем режиме. Значение по сбросу для обоих полей – 4, что соответствует делению входной тактовой частоты в 5 раз. Поскольку для коэффициентов делителей частоты доступны только целочисленные значения и частота при инициализации должна быть 10 МГц, на тактовый вход передатчика должен поступать сигнал с частотой, кратной 10 МГц.

### **Синхронизация**

Скорости передачи бит приёмника и передатчика могут быть в восемь раз выше, чем основная тактовая частота процессора. Такое соотношение приводится с учетом неустойчивости фазы и искажения сигналов, поэтому соединение может запуститься даже и при большей разнице частот. Однако следует принимать во внимание тот факт, что между частотой выборки и скоростью передачи бит при приёме нет прямой зависимости.

Требования по синхронизации тактирования не единственный фактор, который может препятствовать достижению предельных значений частот, могут присутствовать технологические ограничения.

### **Конфигурация буферов FIFO**

FIFO АНВ приемника состоит из 64 записей разрядностью в 32 бита. Организация используемой памяти 64×32 бит.

FIFO АНВ передатчика состоит из 64 записей разрядностью в 32 бита. Организация используемой памяти 64×32 бит.

FIFO N-символов приемника состоит из 64 записей разрядностью в 9 бит. Организация используемой памяти 64×9 бит.

## 31 Контроллер интерфейса OWI

В состав микроконтроллера входят два контроллера интерфейса однопроводной линии типа 1-wire, применяемой для связи с устройствами с низкой скоростью передачи данных (до 125 Кбит/с). Контроллер позволяет организовать полудуплексную двунаправленную связь, и обладает возможностью аппаратного распознавания подключенных устройств к линии связи. Интерфейс может работать только в режиме мастера.

Контроллер интерфейса OWIx (где x - 0 или 1) состоит из (рисунок 31.1):

- делителя частоты для формирования длительности одного стартового импульса ( $T_{START}$ );
- сдвигового 8-битного регистра для приема/передачи данных;
- выходного буфера приемника;
- схемы формирования стартовой последовательности.

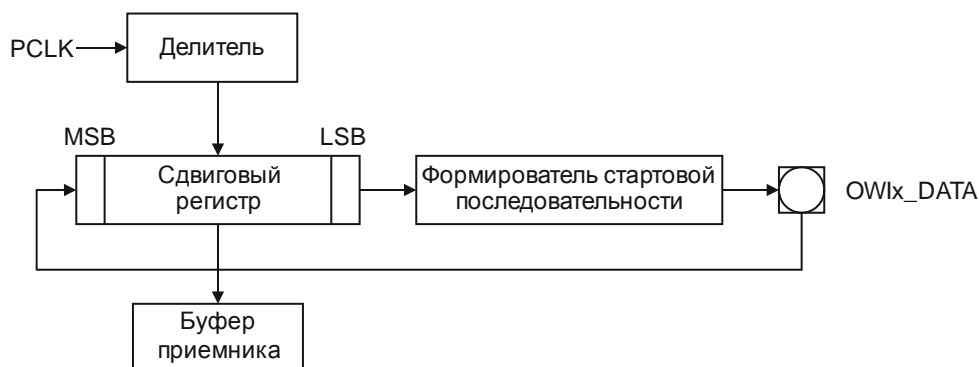


Рисунок 31.1 – Структурная схема контроллера OWI

### 31.1 Функционирование интерфейса

Для включения работы контроллера следует установить бит EN в регистре CTRL0.

Интерфейс выдает и принимает данные с младшего информационного бита. Все транзакции инициируются мастером. Данные передаются побитно.

Начало каждого бита сопровождается стартовой последовательностью (формирование мастером низкого уровня на линии) на время  $T_{START}$ . Время  $T_{START}$  задается через регистр CTRL0 в поле DIV. Значение DIV вычисляется как  $T_{START} * F_{PCLK}$ . Типичное значение  $T_{START}$  5-6 мкс.

Длительность принимаемого или передаваемого бита равна  $T_{БИТ} = T_{START} * ОБРЕ$ . ОБРЕ задается в регистре CTRL1. Типичное значение  $T_{БИТ}$  60-80 мкс. Нули передаются низким уровнем, удерживаемым мастером или устройством, единицы - высоким уровнем, формируемым схемой подтяжки к питанию.

Биты разделяются межбитовым интервалом высокого уровня, формируемого схемой подтяжки к питанию,  $T_{INTRБИТ} = 6-8$  мкс.

#### Операция записи

Процесс записи запускается записью значения в регистр DATA.

При записи, после формирования стартовой последовательности, мастер формирует на линии низкий или высокий уровень в зависимости от значения передаваемого бита.

Данная последовательность действий происходит до тех пор, пока не будут переданы все 8 бит. Временные диаграммы работы контроллера интерфейса OWI при выполнении операции записи представлены на рисунке 31.2. Жирные черные линии соответствуют удержанию мастером низкого уровня, жирные серые - работа схемы подтяжки.

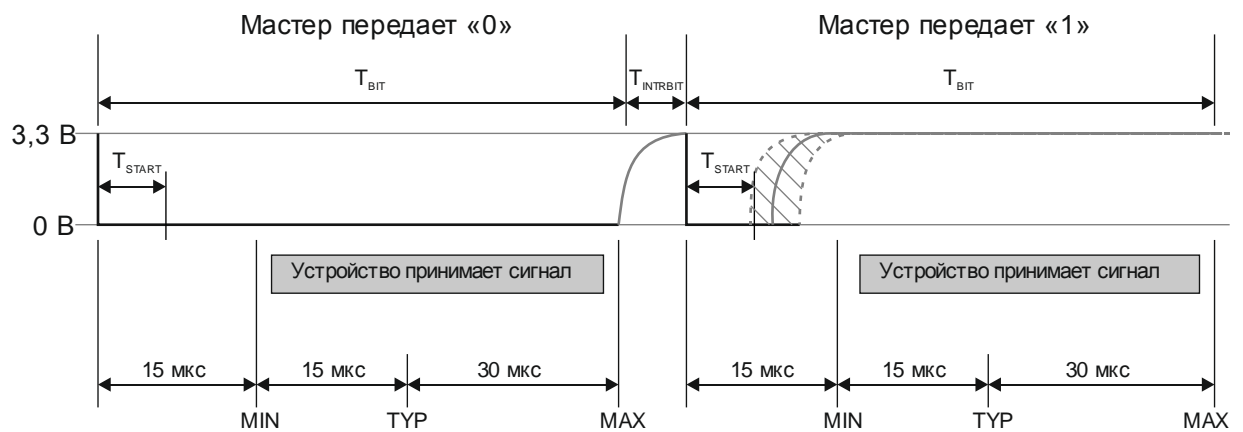


Рисунок 31.2 – Операция записи

### Операция чтения

Процесс чтения запускается установкой бита RDCMD регистра CMD. Результат чтения считывается из регистра DATA после завершения транзакции.

При чтении, после формирования стартовой последовательности, мастер снимает низкий уровень с линии. Ведомое устройство, в зависимости от значения передаваемого бита, формирует на линии низкий или высокий уровень, который принимается мастером через время  $T_{START}$  после стартового бита, т. е. через время  $T_{READ} = 2 \times T_{START}$ . Данная последовательность действий происходит до тех пор, пока не будут прочитаны все 8 бит.

Временные диаграммы работы контроллера интерфейса OWI при выполнении операции чтения представлены на рисунке 31.3. Жирные черные линии соответствуют удержанию мастером низкого уровня, тонкие черные - удержание устройством низкого уровня, жирные серые - работа схемы подтяжки.

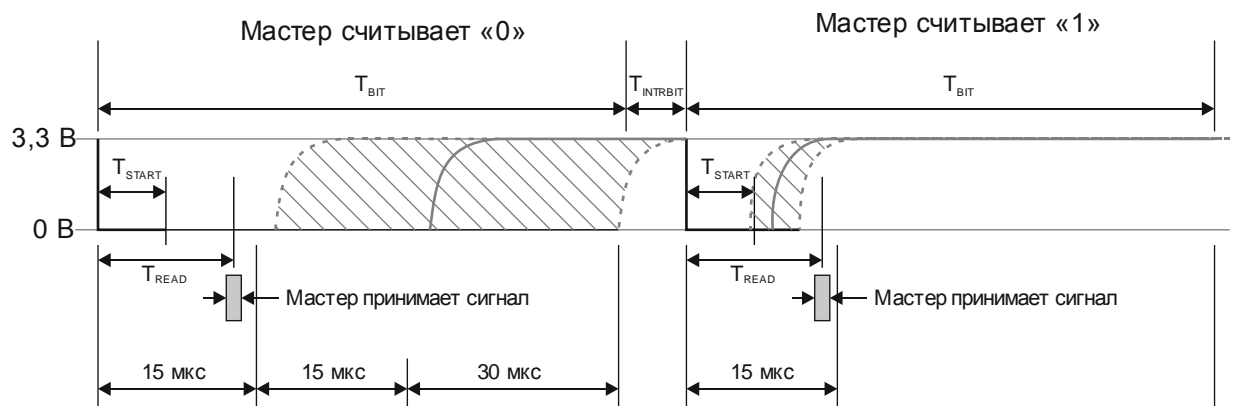


Рисунок 31.3 – Структурная схема контроллера OWI

### Сброс

Для посылки последовательности сброса линии в регистре CMD следует установить бит RSTCMD. При этом мастер формирует логический ноль на линии на время  $T_{START} * RSTPER$ . Значение RSTPER задается в регистре CTRL1.

Определение присутствия на линии устройств определяется по наличию нуля на линии после формирования последовательности сброса через время  $T_{START} * PRESER$ . Значение PRESER задается в регистре CTRL1. Если ведомое устройство на линии отвечает, то выставляется бит PRES в регистре STAT.

### **Прерывание**

На время транзакции выставляется бит **BUSY** в регистре **STAT**, который снимается автоматически после завершения работы контроллера интерфейса. После транзакции формируется сигнал **DONE** в регистре **STAT**. Если в регистре **IRQ** установлен бит **IRQDONEEN**, то после завершения транзакции выставляется сигнал прерывания. Для снятия сигнала прерывания следует записать «1» в бит **DONE** регистра **STAT**.

## 32 Программно-аппаратные средства отладки

Для освоения и изучения 32-разрядного микроконтроллера 1921BK028, а также для макетирования и отладки систем пользователя на его основе, следует использовать макетно-отладочную плату, которая позволяет подключать внешние элементы к портам микроконтроллера, работать с внешними интерфейсами, а также программировать встроенную Flash-память и выполнять отладку и оценку работы прикладных программ.

Для создания программного обеспечения рекомендуется использовать программный продукт CodeMaster++[ARM], который представляет собой набор программно-аппаратных средств для разработки и отладки систем на базе микроконтроллера 1921BK028.

Среда разработки CodeMaster++[ARM] включает в себя менеджер проектов, редактор исходных кодов, компилятор (C, C++), средства отладки и симуляции микроконтроллера 1921BK028. Среда позволяет осуществлять отладку программ, а также программирование микроконтроллера посредством JTAG эмулятора JEM-NT-СМ4.

Адаптер JEM-NT-СМ4 обеспечивает взаимодействие между интегрированной средой разработки CodeMaster++[ARM], установленной на персональном компьютере, и отладочными ресурсами, встроенными в микроконтроллер 1921BK028, а также выполнение отладочных функций. Информационный обмен с микроконтроллером осуществляется по одному из отладочных портов JTAG или SWD.

Отладка пользовательской программы предполагает два основных режима работы: выполнение программы в режиме реального времени RUN и останов программы на определенном адресе или в определенный момент выполнения HALT. Большинство отладочных функций доступно исключительно в режиме останова. В этом режиме отладчик JEM-NT-СМ4 во взаимодействии с CodeMaster++[ARM] позволяет анализировать и изменять ход исполнения пользовательской программы между отдельными участками программы, исполняемыми в режиме RUN.

При работе с пользовательской программой адаптер JEM-NT-СМ4 обеспечивает выполнение следующих отладочных действий:

- сброс микроконтроллера с остановом пользовательской программы на начальном адресе;
- запуск на выполнение программы в режиме реального времени RUN;
- останов программы в произвольный момент времени STOP;
- определение и изменение адреса выполнения программы (чтение и запись счетчика команд);
- чтение и запись доступных ресурсов микроконтроллера (ОЗУ, Flash, SFR и т. д.);
- установку и снятие точек останова по адресу выполнения программы;
- запуск на выполнение программы до определенного места в исходном коде (до курсора, до адреса);
- пошаговое исполнение программы: шаги низкого и высокого уровней, с заходом и без захода в подпрограммы.

## **Заключение**

В настоящем руководстве пользователя представлено описание архитектуры, функционального построения и периферии микроконтроллера 1921BK028. Техническое описание может служить практическим руководством по применению микроконтроллера для разработчиков систем на его основе и программистов.

## Приложение А (обязательное) Регистры микроконтроллера

Для каждого блока отдельно приведены абсолютные базовые адреса, а в описаниях регистров показаны смещения относительно его базового адреса, если не указано иное. Абсолютный адрес регистра вычисляется путем сложения абсолютного базового адреса блока и смещения этого регистра.

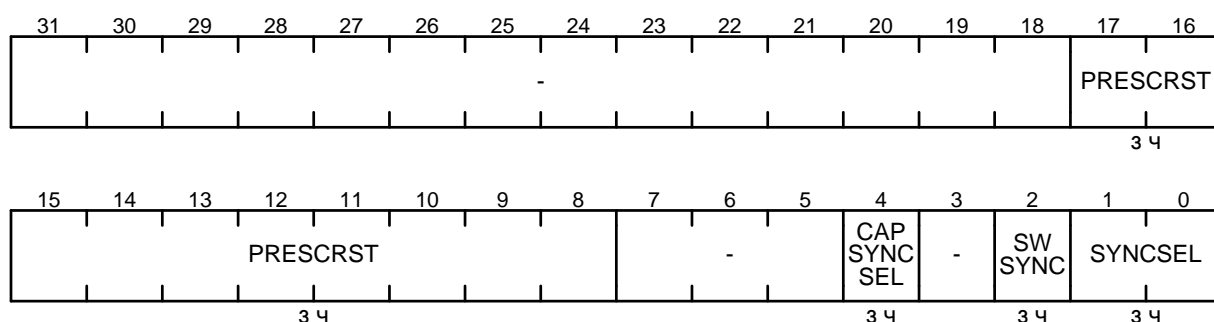
### А.1 Регистры блока управления системой SIU

Базовый адрес: 4008\_0000h

#### PWMSYNC – регистр настройки синхронизации PWM

Смещение: + 10h

Сброс: 0h

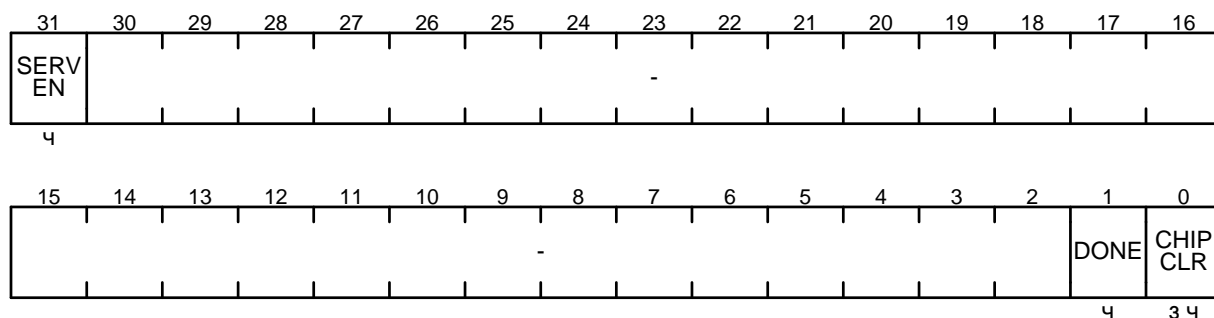


Поле	Биты	Описание
PRESCRST	17-8	Биты сброса счетчиков предварительных делителей блоков ШИМ. Запись нуля в младший бит поля сбрасывает счетчик блока ШИМ0, в первый бит – счетчик блока ШИМ1, в старший бит – счетчик блока ШИМ9. Запись единицы разрешает счет
CAPSYNCSEL	4	Бит выбора источника синхронизации для блоков захвата 3, 4 и 5, см. рисунок 19.3
WSYNC	2	Генерация импульса синхронизации ШИМ
		0   Нет реакции
	1	Генерируется импульс синхронизации на входе SYNC1 блока ШИМ0 (имитация внешней синхронизации)
SYNCSEL	1-0	Поле выбора схемы синхронизации блоков ШИМ, см. рисунок 20.3
–	31-18, 7-5, 3	Зарезервировано

## SERVCTL – регистр настройки сервисного режима

Смещение: + 14h

Сброс: 0h

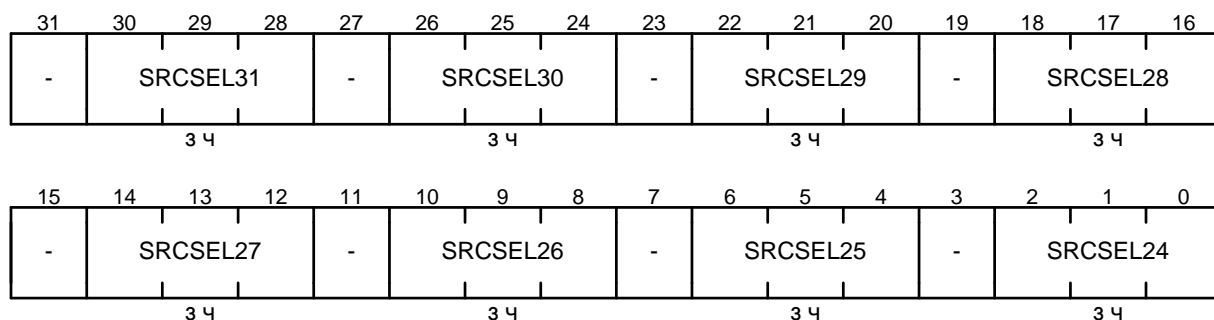


Поле	Биты	Описание
SERVEN	31	0   Обычный режим работы
		1   Сервисный режим. Во время сброса на выводе SERVEN была единица
DONE	1	Флаг завершения команды сервисного стирания
		0   Команда не завершена
		1   Команда завершена
CHIPCLR	0	Бит стирания. Запись единицы активирует полное стирание всей Flash-памяти микроконтроллера
–	30-2	Зарезервировано

## DMAMUX – регистр настройки конфигурируемых каналов DMA

Смещение: + 20h

Сброс: 0h



Поле	Биты	Описание
SRCSEL31	30-28	Выбор источника для канала 31 контроллера DMA
		0h   Запрос от порта H
		1h   Запрос от блока ETMR3
		2h   Запрос от блока PWM6 канал B
		3h   Запрос от блока PWM7 канал B
		4h   Запрос от порта L
		5h   Запрос от порта M
		6h, 7h   Нет аппаратного источника



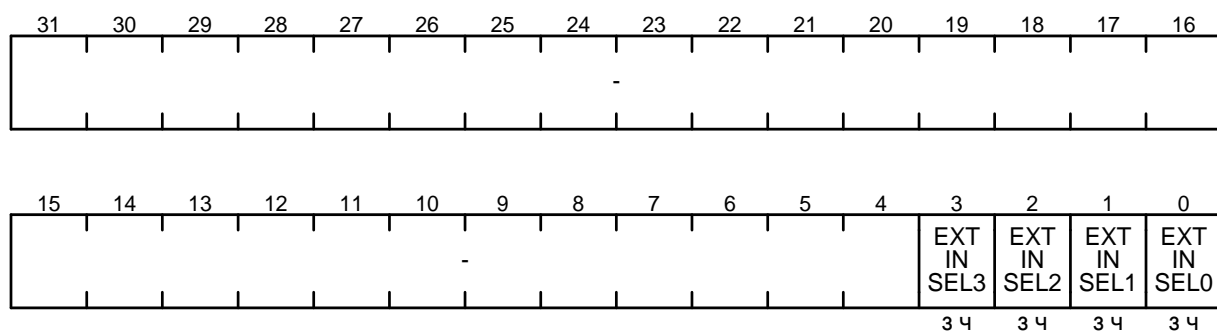
Поле	Биты	Описание
SRCSEL30	26-24	Выбор источника для канала 30 контроллера DMA
		0h   Запрос от порта G
		1h   Запрос от блока ETMR2
		2h   Запрос от блока PWM4 канал B
		3h   Запрос от блока PWM5 канал B
		4h   Запрос от блока QEP3
		5h   Запрос от канала SDFM3
		6h, 7h   Нет аппаратного источника
SRCSEL29	22-20	Выбор источника для канала 29 контроллера DMA
		0h   Запрос от порта F
		1h   Запрос от блока ETMR1
		2h   Запрос от блока PWM2 канал B
		3h   Запрос от блока PWM3 канал B
		4h   Запрос от блока QEP2
		5h   Запрос от канала SDFM2
		6h, 7h   Нет аппаратного источника
SRCSEL28	18-16	Выбор источника для канала 28 контроллера DMA
		0h   Запрос от порта E
		1h   Запрос от блока ETMR0
		2h   Запрос от блока PWM0 канал B
		3h   Запрос от блока PWM1 канал B
		4h   Запрос от блока PWM8 канал B
		5h   Запрос от блока PWM9 канал B
		6h, 7h   Нет аппаратного источника
SRCSEL27	14-12	Выбор источника для канала 27 контроллера DMA
		0h   Запрос от порта D
		1h   Запрос от блока TMR3
		2h   Запрос от блока PWM6 канал A
		3h   Запрос от блока PWM7 канал A
		4h   Запрос от порта J
		5h   Запрос от порта K
		6h, 7h   Нет аппаратного источника
SRCSEL26	10-8	Выбор источника для канала 26 контроллера DMA
		0h   Запрос от порта C
		1h   Запрос от блока TMR2
		2h   Запрос от блока PWM4 канал A
		3h   Запрос от блока PWM5 канал A
		4h   Запрос от блока QEP1
		5h   Запрос от канала SDFM1
		6h, 7h   Нет аппаратного источника

Поле	Биты	Описание
SRCSEL25	6-4	Выбор источника для канала 25 контроллера DMA
		0h   Запрос от порта B
		1h   Запрос от блока TMR1
		2h   Запрос от блока PWM2 канал A
		3h   Запрос от блока PWM3 канал A
		4h   Запрос от блока QEP0
		5h   Запрос от канала SDFM0
		6h, 7h   Нет аппаратного источника
SRCSEL24	18-16	Выбор источника для канала 24 контроллера DMA
		0h   Запрос от порта A
		1h   Запрос от блока TMR0
		2h   Запрос от блока PWM0 канал A
		3h   Запрос от блока PWM1 канал A
		4h   Запрос от блока PWM8 канал A
		5h   Запрос от блока PWM9 канал A
		6h, 7h   Нет аппаратного источника
–	31, 27, 23, 19, 15, 11, 7, 3	Зарезервировано

## TMRMUX – регистр настройки входа синхронизации таймеров TMR

Смещение: + 1Ch

Сброс: 0h

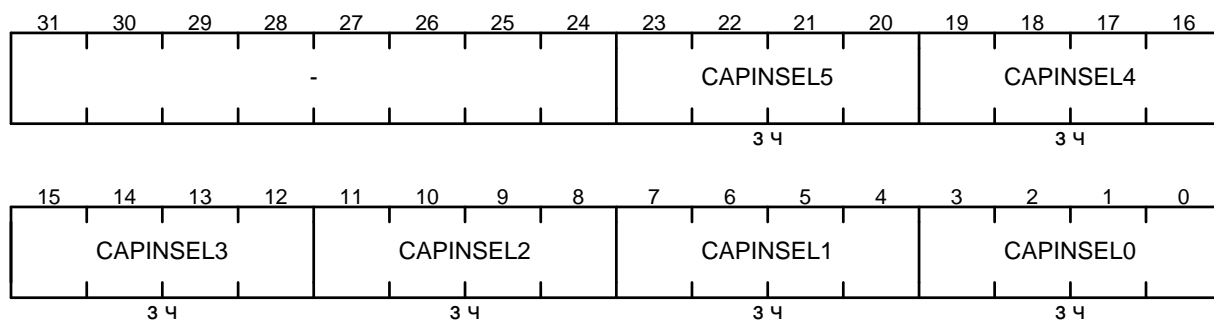


Поле	Биты	Описание
EXTINSEL3	3	Выбор источника для входа внешней синхронизации таймера TMR3
		0   Вывод TMR3_IN
		1   Сигнал RTC1K
EXTINSEL2	2	Выбор источника для входа внешней синхронизации таймера TMR2
		0   Вывод TMR2_IN
		1   Сигнал RTC1K
EXTINSEL0	1	Выбор источника для входа внешней синхронизации таймера TMR1
		0   Вывод TMR1_IN
		1   Сигнал RTC1K
EXTINSEL0	0	Выбор источника для входа внешней синхронизации таймера TMR0
		0   Вывод TMR0_IN
		1   Сигнал RTC1K
–	31-4	Зарезервировано

## CAPMUX – регистр настройки входов блоков захвата

Смещение: + 20h

Сброс: 0h



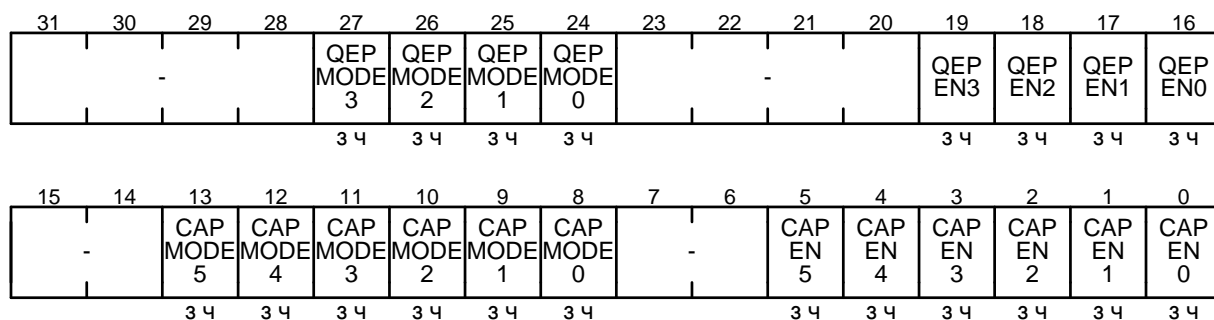
Поле	Биты	Описание	
CAPINSEL5	23-20	Выбор источника для входа блока захвата ECAP5	
		0h-7h	Сигнал CAP_PWM5_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_I_F
		Dh	Сигнал QEP1_I_F
		Eh	Сигнал QEP2_I_F
		Fh	Сигнал QEP3_I_F
CAPINSEL4	19-16	Выбор источника для входа блока захвата ECAP4	
		0h-7h	Сигнал CAP_PWM4_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_B_F
		Dh	Сигнал QEP1_B_F
		Eh	Сигнал QEP2_B_F
		Fh	Сигнал QEP3_B_F
CAPINSEL3	15-12	Выбор источника для входа блока захвата ECAP3	
		0h-7h	Сигнал CAP_PWM3_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_A_F
		Dh	Сигнал QEP1_A_F
		Eh	Сигнал QEP2_A_F
		Fh	Сигнал QEP3_A_F

Поле	Биты	Описание	
CAPINSEL2	11-8	Выбор источника для входа блока захвата ECAP2	
		0h-7h	Сигнал CAP_PWM2_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_I_F
		Dh	Сигнал QEP1_I_F
		Eh	Сигнал QEP2_I_F
		Fh	Сигнал QEP3_I_F
CAPINSEL1	7-4	Выбор источника для входа блока захвата ECAP1	
		0h-7h	Сигнал CAP_PWM1_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_B_F
		Dh	Сигнал QEP1_B_F
		Eh	Сигнал QEP2_B_F
		Fh	Сигнал QEP3_B_F
CAPINSEL0	3-0	Выбор источника для входа блока захвата ECAP0	
		0h-7h	Сигнал CAP_PWM0_F
		8h	Сигнал SDFM_CMPHZ0
		9h	Сигнал SDFM_CMPHZ1
		Ah	Сигнал SDFM_CMPHZ2
		Bh	Сигнал SDFM_CMPHZ3
		Ch	Сигнал QEP0_A_F
		Dh	Сигнал QEP1_A_F
		Eh	Сигнал QEP2_A_F
		Fh	Сигнал QEP3_A_F
	31-24	Зарезервировано	

## QCQUALCTL – регистр настройки фильтрации входов ЕСАР и QEP

Смещение: + 24h

Сброс: 0h

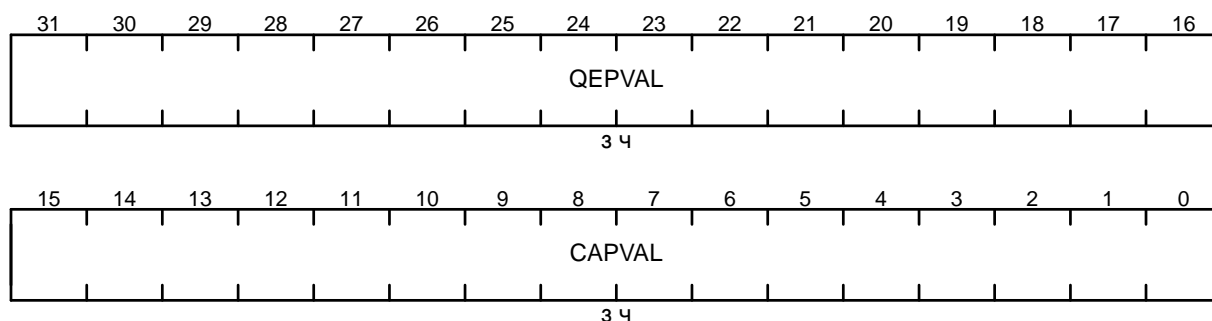


Поле	Биты	Описание
QEPMODE <sub>n</sub>	27-24	Выбор режима фильтрации входов QEP <sub>n</sub> (n – номер QEP от 0 до 3)
		0   По трём одинаковым отсчетам
		1   По шести одинаковым отсчетам
QEPEN <sub>n</sub>	19-16	Включение дополнительной фильтрации входов QEP <sub>n</sub> (n – номер QEP от 0 до 3)
		0   Дополнительный фильтр отключен
		1   Дополнительный фильтр включен
CAPMODE <sub>m</sub>	13-8	Выбор режима фильтрации входов ЕСАР <sub>m</sub> (m – номер ЕСАР от 0 до 5)
		0   По трём одинаковым отсчетам
		1   По шести одинаковым отсчетам
CAPEN <sub>m</sub>	5-0	Включение дополнительной фильтрации входов ЕСАР <sub>m</sub> (m – номер ЕСАР от 0 до 5)
		0   Дополнительный фильтр отключен
		1   Дополнительный фильтр включен
	31-28, 23-20, 15-14, 7-6	Зарезервировано

## QCQUALSAMPLE – регистр настройки периода фильтрации входов ЕСАР и QEP

Смещение: + 28h

Сброс: 0h

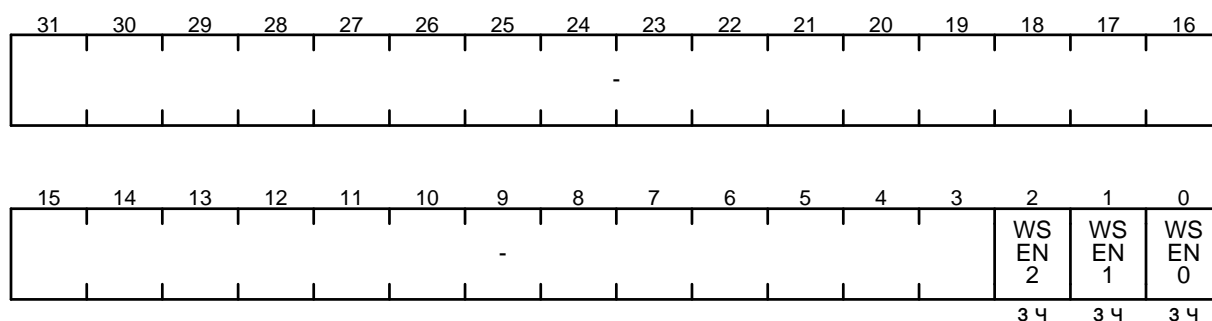


Поле	Биты	Описание
QEPVAL	31-16	Временной интервал (в тактах FCLK) между отсчетами фильтров. Заданное значение является единым для всех входных фильтров QEP.
CAPVAL	15-0	Временной интервал (в тактах FCLK) между отсчетами фильтров. Заданное значение является единым для всех входных фильтров ЕСАР.

## RAMWSCTL – регистр настройки тактов ожидания ОЗУ

Смещение: + 2Ch

Сброс: 0h

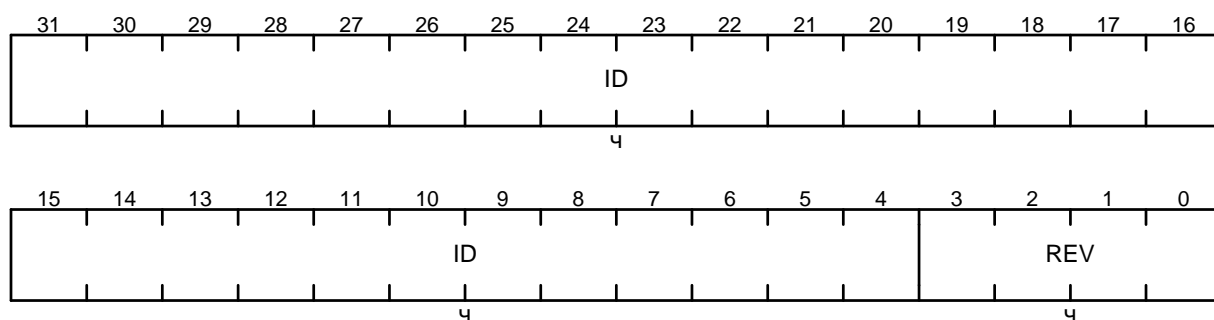


Поле	Биты	Описание
WSENn	2-0	Включение дополнительного такта ожидания при доступе к ОЗУn (где n от 0 до 2)
		0   Без такта ожидания
		1   С дополнительным тактом ожидания
	31-3	Зарезервировано

## CHIPID – регистр идентификации системы

Смещение: + FFCh

Сброс: 3ABF\_2FD1h



Поле	Биты	Описание
ID	31-4	Номер модели
REV	3-0	Номер ревизии



## A.2 Регистры блока управления сбросом и синхронизацией RCU

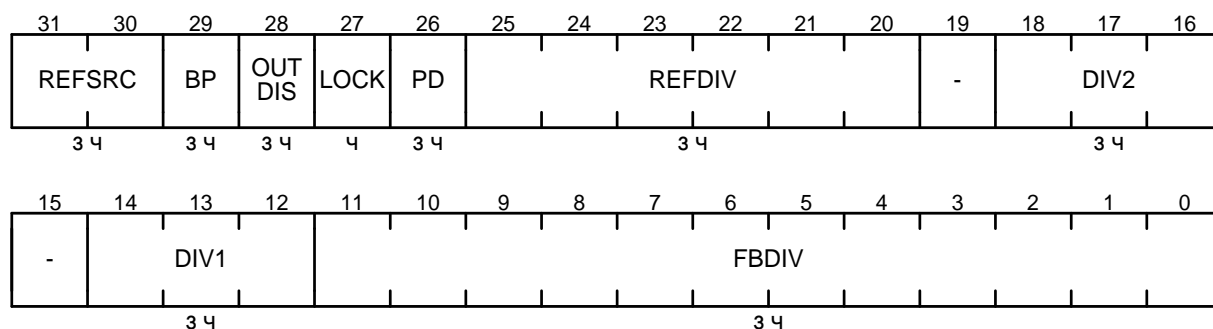
Базовый адрес: 4008\_1000h

Смещение: + 40h (MILSTDCFG) Регистры настройки тактирования MILSTD  
 + 50h (SPWRCFG) Регистры настройки тактирования SPWR  
 + 60h (UARTCFG) Регистры настройки тактирования UART  
 + 80h (SPICFG) Регистры настройки тактирования SPI

### PLLCFG – регистр конфигурации PLL

Смещение: + 08h

Сброс: 3411\_1101h

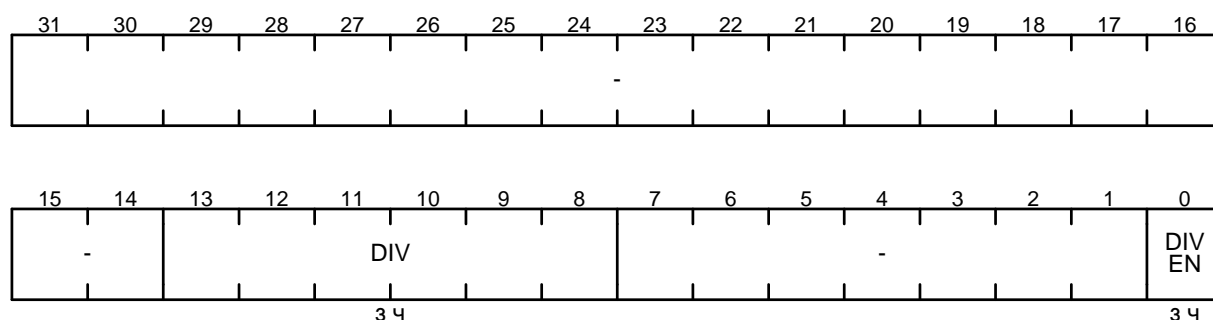


Поле	Биты	Описание	
REFSRC	31-30	Выбор источника входной частоты $F_{ref}$	
		0h	Внешний тактовый сигнал OSECLK
		1h	Внутренний тактовый сигнал OSICLK
		2h	Внешний сигнал с вывода PLL_EXTSRC
BP	29	Бит разрешения сквозного прохождения $F_{ref}$ на выход PLL	
		0	Запрещено
		1	Разрешено
OUTDIS	28	Бит запрещения генерации выходной частоты с блока PLL	
		0	Разрешена
		1	Запрещена
LOCK	27	Устанавливается, если выходная частота блока PLL стабильна	
PD	26	Бит установки энергосберегающего режима PLL	
		0	Запрещен
		1	Разрешен
REFDIV	25-20	Коэффициент деления входной частоты REFDIV. Значения от 1 до 63.	
DIV2	18-16	Коэффициент деления DIV2. Значения от 1 до 7.	
DIV1	14-12	Коэффициент деления DIV1. Значения от 1 до 7.	
FBDIV	11-0	Коэффициент деления обратной связи FBDIV. Значения от 16 до 200.	
-	19, 15	Зарезервировано	

## PLLDIV – регистр внешнего делителя PLL

Смещение: + 0Ch

Сброс: 0h

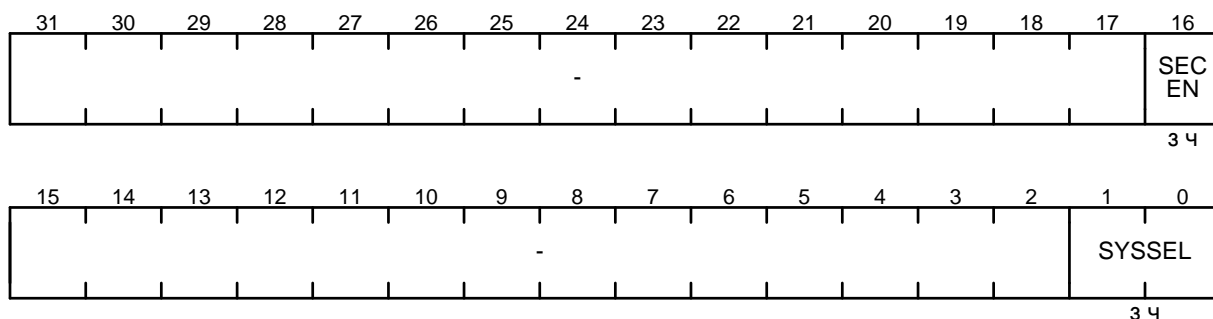


Поле	Биты	Описание
DIV	13-8	Коэффициент выходного делителя PLL, вычисляемый как DIV+1
DIVEN	0	Бит разрешения выходного делителя частоты блока PLL
		0   Запрещен
		1   Разрешен
–	31-14, 7-1	Зарезервировано

## SYSCCLKCFG – регистр конфигурации системного тактового сигнала

Смещение: + 10h

Сброс: 0h

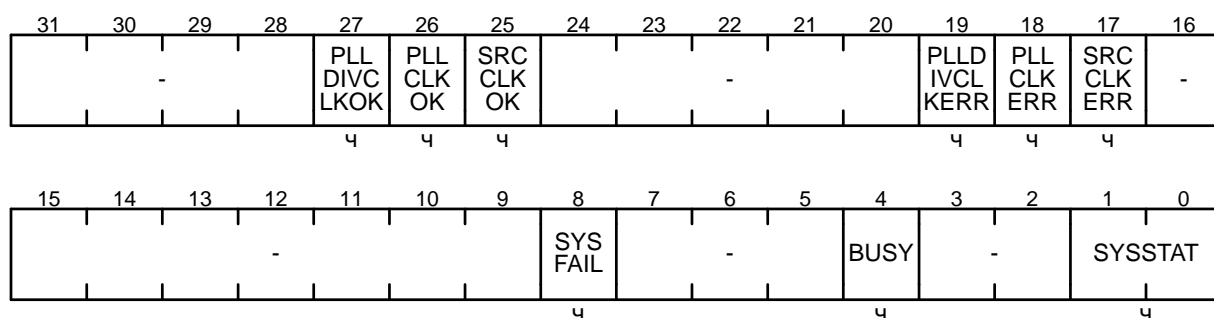


Поле	Биты	Описание
SECEN	16	Бит включения системы слежения за тактовым сигналом
		0   Выключена
		1   Включена
SYSSSEL	1-0	Выбор источника системного тактового сигнала
		00b   Сигнал REFCLK
		01b   Сигнал SRCCLK
		10b   Сигнал PLLCLK с выхода блока PLL
		11b   Сигнал PLLDIVCLK после внешнего делителя PLL
–	31-17, 15-2	Зарезервировано

## SYSCLKSTAT – регистр статуса системного тактового сигнала

Смещение: + 14h

Сброс: C100\_0000h

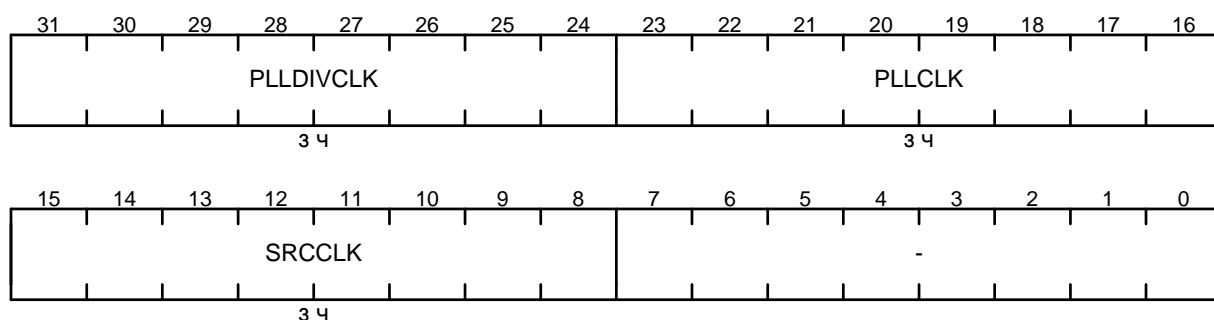


Поле	Биты	Описание
PLLDIVCLKOK	27	Флаг наличия тактового сигнала PLLDIVCLK
PLLCLKOK	26	Флаг наличия тактового сигнала PLLCLK
SRCCLKOK	25	Флаг наличия тактового сигнала SRCCLK
PLLDIVCLKERR	19	Флаг отсутствия тактового сигнала PLLDIVCLK
PLLCLKERR	18	Флаг отсутствия тактового сигнала PLLCLK
SRCCLKERR	17	Флаг отсутствия тактового сигнала SRCCLK
SYSFAIL	8	Флаг ошибки при пропадании тактового сигнала SYSCLK
BUSY	4	Флаг занятости блока RCU (например, во время смены тактового сигнала)
SYSSTAT	1-0	Текущий источник системного тактового сигнала
		00b   Сигнал REFCLK
		01b   Сигнал SRCCLK
		10b   Сигнал PLLCLK с выхода блока PLL
		11b   Сигнал PLLDIVCLK после внешнего делителя PLL
–	31-28, 24-20, 16-9, 7-5, 3-2	Зарезервировано

## SECPRD – регистр слежения за состоянием системного тактового сигнала

Смещение: + 18h

Сброс: FFFF\_FF00h

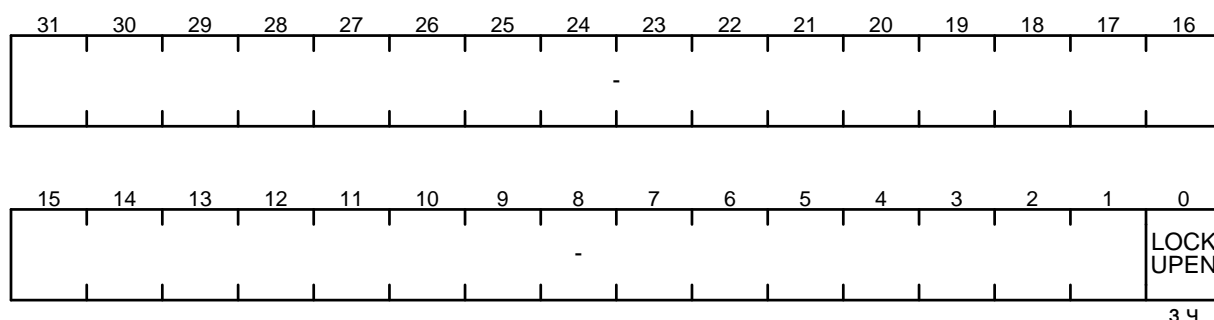


Поле	Биты	Описание
PLLDIVCLK	31-24	Максимальное значение счетчика периодов для детектирования пропадания сигнала PLLDIVCLK
PLLCLK	23-16	Максимальное значение счетчика периодов для детектирования пропадания сигнала PLLCLK
SRCCLK	15-8	Максимальное значение счетчика периодов для детектирования пропадания сигнала SRCCLK
–	7-0	Зарезервировано

## SYSRSTCFG – регистр настройки системного сброса

Смещение: + 1Ch

Сброс: 0h

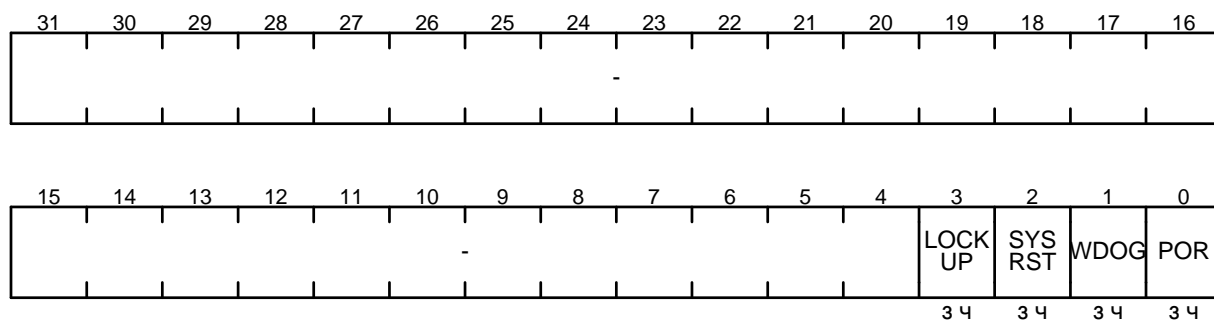


Поле	Биты	Описание
LOCKUPEN	0	Бит разрешения сброса, когда процессор в состоянии LOCKUP
		0   Запрещен
		1   Разрешен
–	31-1	Зарезервировано

## SYSRSTSTAT – регистр статуса системного сброса

Смещение: + 20h

Сброс: 1h

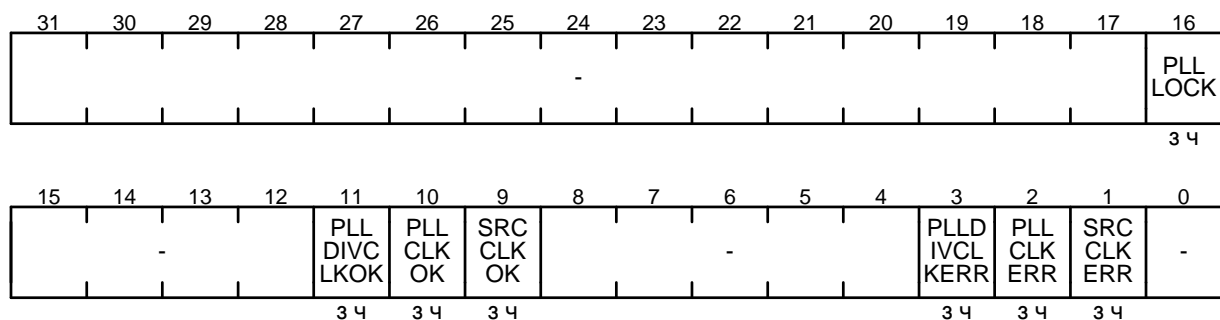


Поле	Биты	Описание
LOCKUP	3	Флаг устанавливается, если последний сброс произошел из-за входа процессора в состояние LOCKUP
SYSRST	2	Флаг устанавливается, если последний сброс произошел в результате подачи сигнала системного сброса
WDOG	1	Флаг устанавливается, если последний сброс был активирован сторожевым таймером
POR	0	Флаг устанавливается, если последний сброс произошел из-за срабатывания схемы POR
–	31-4	Зарезервировано

## INTEN – регистр разрешения прерываний

Смещение: + 24h

Сброс: 0h

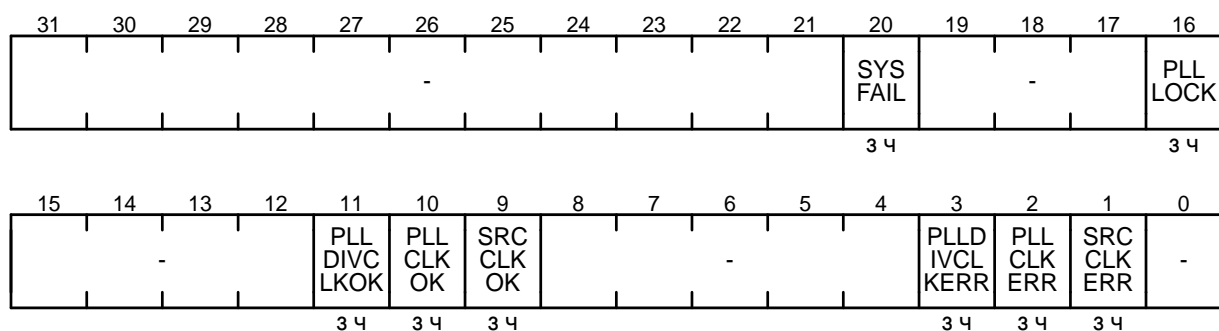


Поле	Биты	Описание
PLLLOCK	16	Разрешение прерывания по установке сигнала LOCK
PLLDIVCLKOK	11	Разрешение прерывания при появлении стабильного тактового сигнала PLLDIVCLK
PLLCLKOK	10	Разрешение прерывания при появлении стабильного тактового сигнала PLLCLK
SRCCLKOK	9	Разрешение прерывания при появлении стабильного тактового сигнала SRCCLK
PLLDIVCLKERR	3	Разрешение прерывания при пропадании тактового сигнала PLLDIVCLK
PLLCLKERR	2	Разрешение прерывания при пропадании тактового сигнала PLLCLK
SRCCLKERR	1	Разрешение прерывания при пропадании тактового сигнала SRCCLK
–	31-17, 15-12, 8-4, 0	Зарезервировано

## INTSTAT – регистр статуса прерываний

Смещение: + 28h

Сброс: 0h



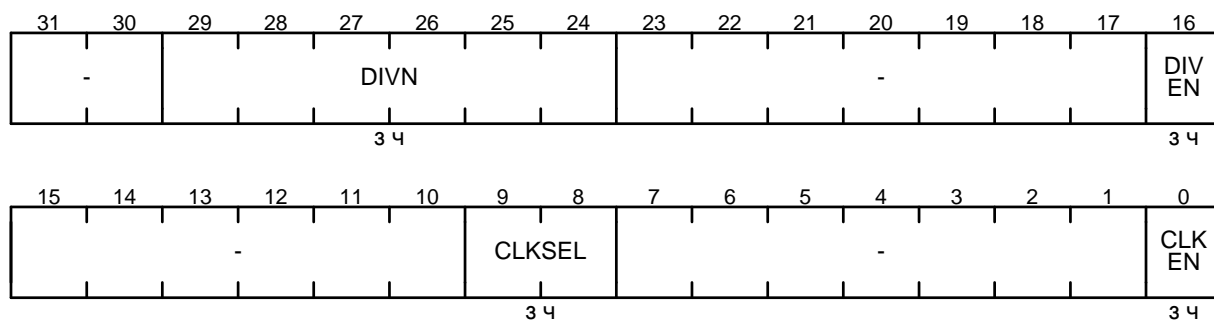
Поле	Биты	Описание
SYSFAIL	20	Флаг устанавливается при попытке перейти на отсутствующий источник тактирования (прерывание, выставляющее данный флаг, соответствует вектору NMI)
PLLLOCK	16	Флаг прерывания по сигналу LOCK
PLLDIVCLKOK	11	Флаг прерывания при появлении стабильного тактового сигнала PLLDIVCLK
PLLCLKOK	10	Флаг прерывания при появлении стабильного тактового сигнала PLLCLK
SRCCLKOK	9	Флаг прерывания при появлении стабильного выбранного тактового сигнала SRCCLK
PLLDIVCLKERR	3	Флаг прерывания при пропадании стабильного тактового сигнала PLLDIVCLK
PLLCLKERR	2	Флаг прерывания при пропадании стабильного тактового сигнала PLLCLK
SRCCLKERR	1	Флаг прерывания при пропадании стабильного выбранного тактового сигнала SRCCLK
–	31-21, 19-17, 15-12, 8-4, 0	Зарезервировано

Примечание – Флаги сбрасываются только программно. Для сброса флага следует записать единицу в соответствующий бит.

## TRACECFG – регистр настройки интерфейса трассировки

Смещение: + 2Ch

Сброс: 1h



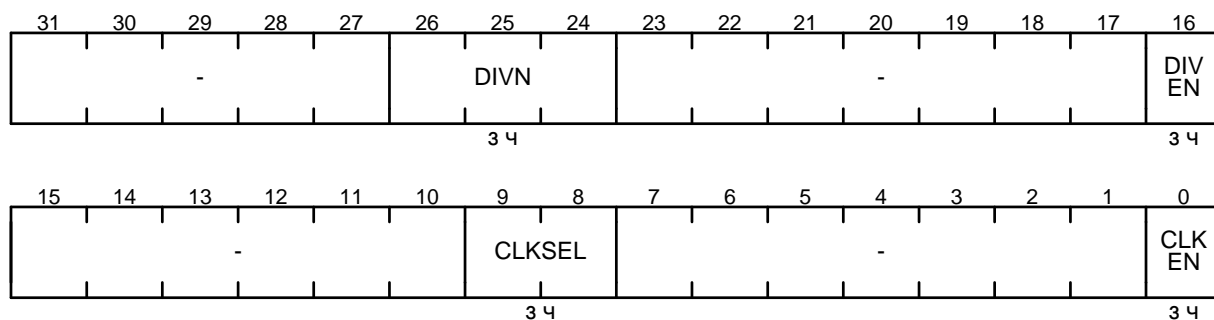
Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входной частоты интерфейса трассировки. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$ .
DIVEN	16	Разрешение делителя входной частоты интерфейса трассировки
CLKSEL	9-8	Выбор источника входной частоты интерфейса трассировки
		00b   Сигнал REFCLK
		01b   Сигнал SRCCLK
		10b   Сигнал SYSCLK
	11b   Сигнал с вывода TRACE_EXTCLK	
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-10, 7-1	Зарезервировано



## CLKOUTCFG – регистр настройки выходного тактового сигнала

Смещение: + 30h

Сброс: 0h

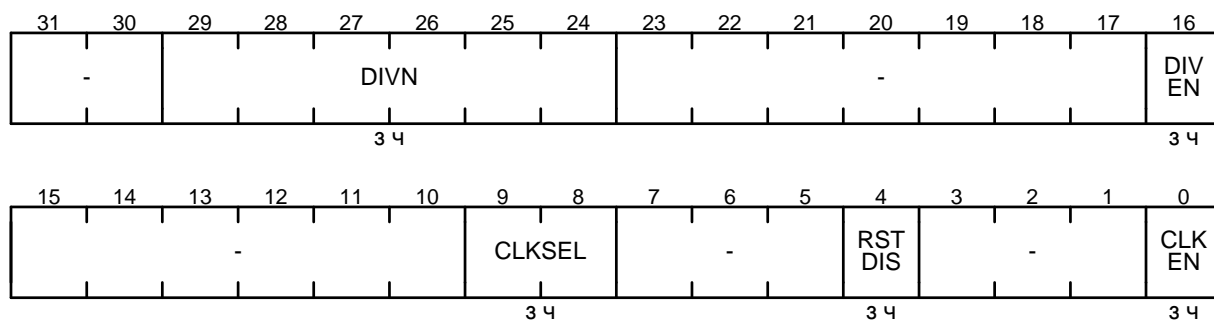


Поле	Биты	Описание
DIVN	26-24	Коэффициент деления выходного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$
DIVEN	16	Разрешение делителя выходного сигнала
CLKSEL	9-8	Выбор источника выходного сигнала
	00b	Сигнал REFCLK
	01b	Сигнал SRCCLK
	10b	Сигнал SYSCLK
	11b	Сигнал PLLCLK с выхода блока PLL
CLKEN	0	Разрешение выходного сигнала
-	31-27, 23-17, 15-10, 7-1	Зарезервировано

## WDTCFG – регистр настройки сторожевого таймера

Смещение: + 34h

Сброс: 0701\_0000h



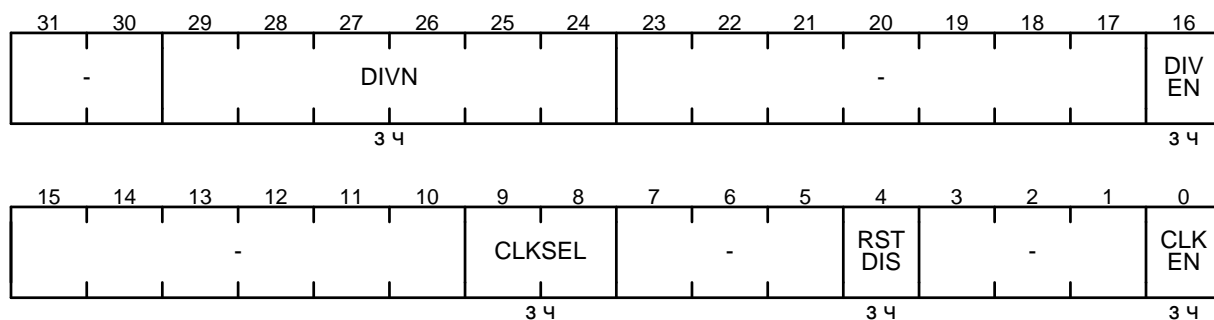
Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$ .
DIVEN	16	Разрешение делителя входного сигнала
CLKSEL	9-8	Выбор источника тактового сигнала
		00b Сигнал REFCLK
		01b Сигнал SRCCLK
		10b Сигнал SYSCLK
11b Сигнал PLLCLK с выхода блока PLL		
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-10, 7-5, 3-1	Зарезервировано

Примечание – Частота тактирования сторожевого таймера должна быть как минимум в 2 раза ниже системной.

## MILSTDCFG – массив регистров настройки MILSTD

Смещение: MILSTDCFG + (4\*n)h, где n – номер блока 0, 1

Сброс: 0h

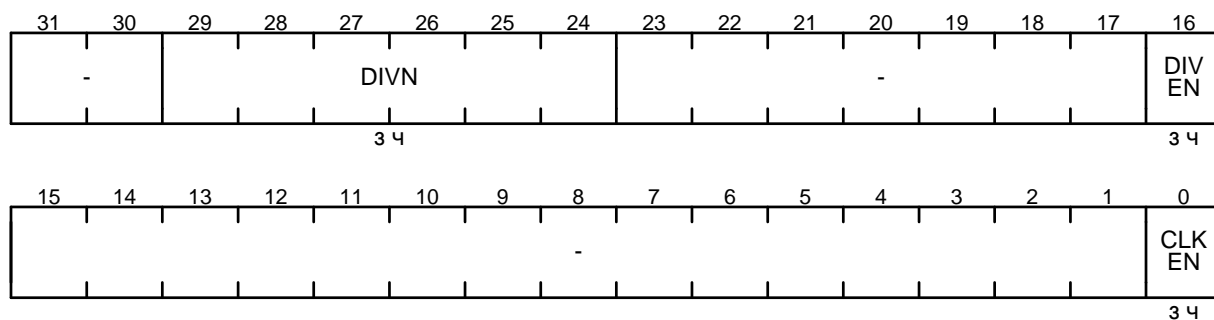


Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$
DIVEN	16	Разрешение делителя входного сигнала
CLKSEL	9-8	Выбор источника тактового сигнала
		00b   Сигнал OSECLK
		01b   Сигнал PLLCLK с выхода блока PLL
		10b   Сигнал PLLDIVCLK после внешнего делителя PLL
11b   Сигнал PLLEXTCLK		
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-10, 7-5, 3-1	Зарезервировано

## SPWRCFG – массив регистров настройки SPWR

Смещение: SPWRCFG + (4\*n)h, где n – номер блока 0, 1

Сброс: 0h

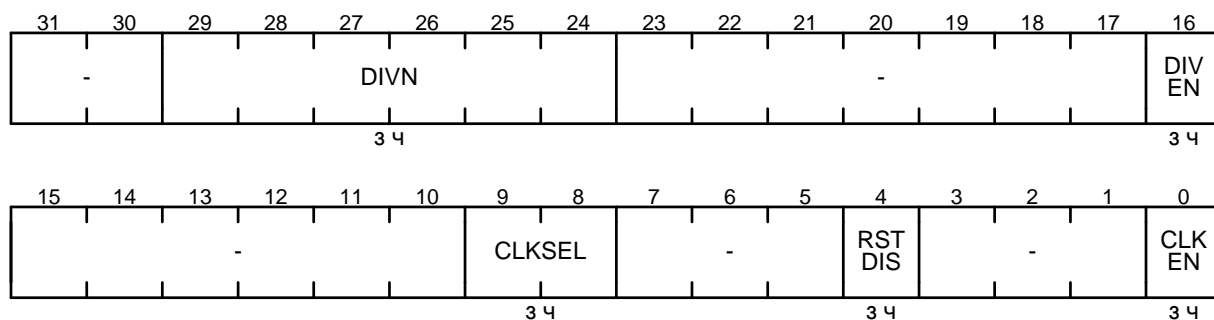


Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$ .
DIVEN	16	Разрешение делителя входного сигнала
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-1	Зарезервировано

## UARTCFG – массив регистров настройки UART

Смещение: UARTCFG + (4\*n)h, где n – номер блока 0-3

Сброс: 0h

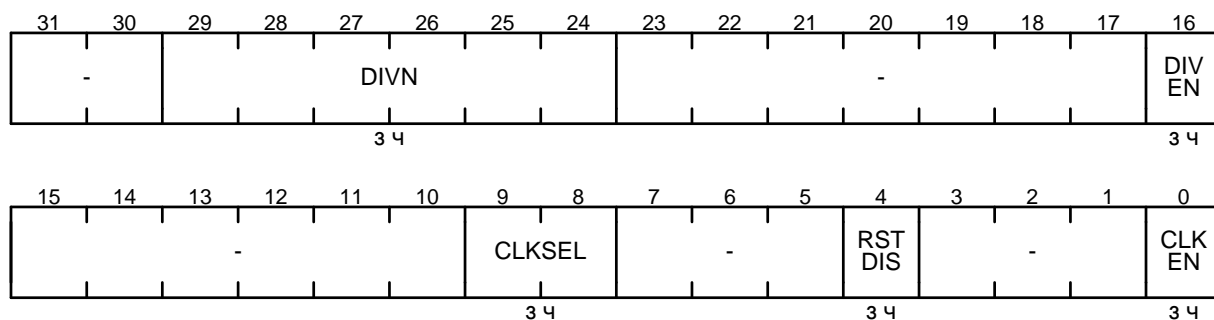


Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$
DIVEN	16	Разрешение делителя входного сигнала
CLKSEL	9-8	Выбор источника тактового сигнала
		00b   Сигнал OSECLK
		01b   Сигнал PLLCLK с выхода блока PLL
		10b   Сигнал PLLDIVCLK после внешнего делителя PLL
11b   Сигнал PLLEXTCLK		
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-10, 7-5, 3-1	Зарезервировано

## SPICFG – регистр настройки SPI

Смещение: SPICFG+ (4\*n)h, где n – номер блока 0-3

Сброс: 0h

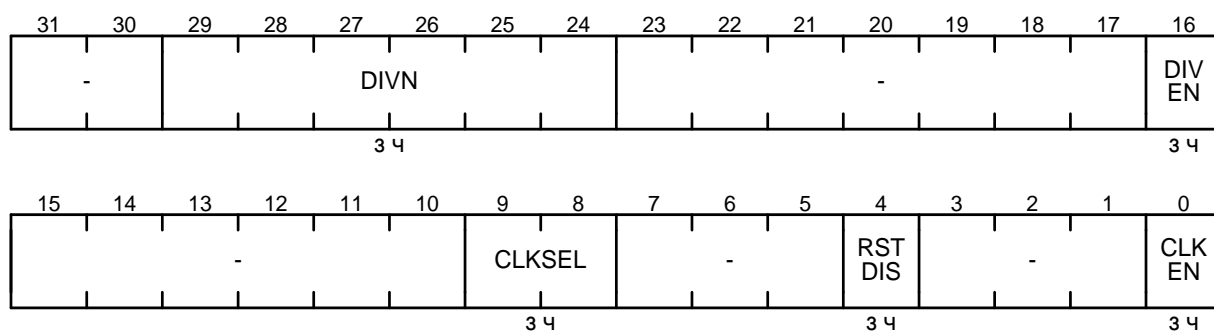


Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$ .
DIVEN	16	Разрешение делителя входного сигнала
CLKSEL	9-8	Выбор источника тактового сигнала
		00b   Сигнал OSECLK
		01b   Сигнал PLLCLK с выхода блока PLL
		10b   Сигнал PLLDIVCLK после внешнего делителя PLL
11b   Сигнал PLLEXTCLK		
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса.
CLKEN	0	Разрешение тактирования
-	31-30, 23-17, 15-10, 7-5, 3-1	Зарезервировано

## ADCCFG – регистр настройки блока АЦП

Смещение: + A0h

Сброс: 0h

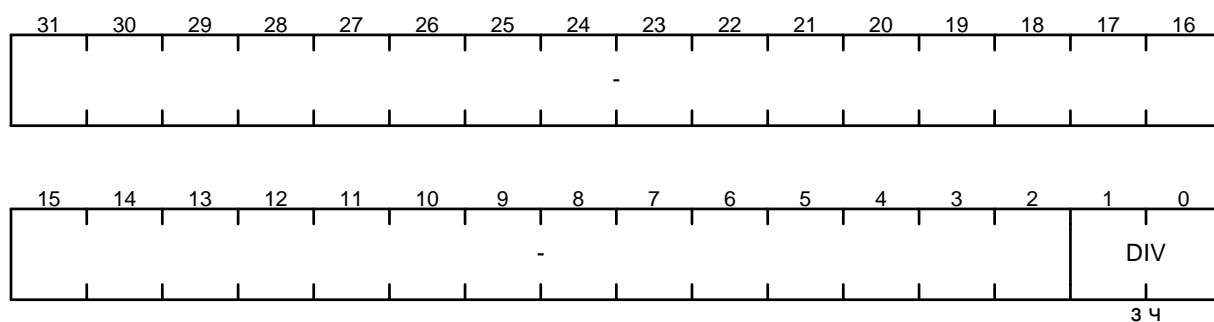


Поле	Биты	Описание
DIVN	29-24	Коэффициент деления входного сигнала. Результирующий коэффициент деления $N = 2 \times (DIVN + 1)$ .
DIVEN	16	Разрешение делителя входного сигнала
CLKSEL	9-8	Выбор источника тактового сигнала
		00b   Сигнал OSECLK
		01b   Сигнал PLLCLK с выхода блока PLL
		10b   Сигнал PLLDIVCLK после внешнего делителя PLL
11b   Сигнал PLEXTCLK		
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса.
CLKEN	0	Разрешение тактирования
–	31-30, 23-17, 15-10, 7-5, 3-1	Зарезервировано

## APBCFG – регистр настройки тактирования шины APB

Смещение: + C0h

Сброс: 0h



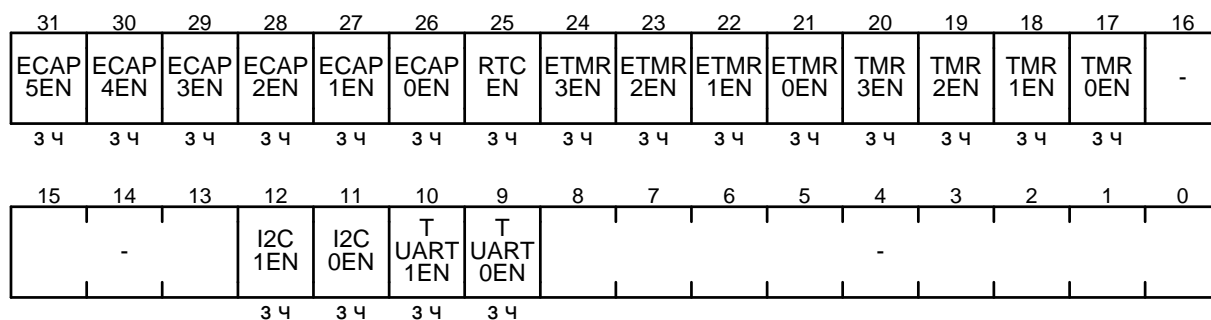
Поле	Биты	Описание	
DIV	1-0	Коэффициент деления SYSCLK для получения тактового сигнала PCLK	
		00 b	DIV = 0 (нет деления)
		01 b	DIV = 2
		10 b	DIV = 4
		11 b	DIV = 8
–	31-2	Зарезервировано	



## PCLKCFG0 – регистр разрешения тактовых сигналов периферийных блоков шины APB

Смещение: + E0h

Сброс: 0h

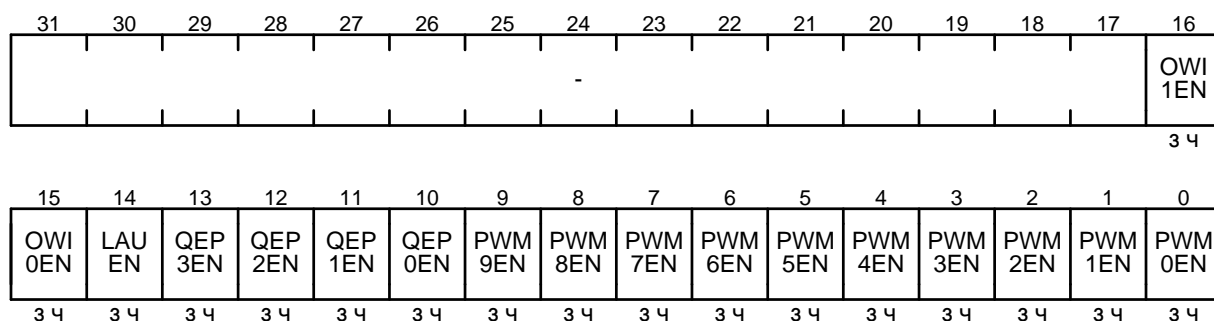


Поле	Биты	Описание
ECAP5EN	31	Бит разрешения тактирования блока ECAP5
ECAP4EN	30	Бит разрешения тактирования блока ECAP4
ECAP3EN	29	Бит разрешения тактирования блока ECAP3
ECAP2EN	28	Бит разрешения тактирования блока ECAP2
ECAP1EN	27	Бит разрешения тактирования блока ECAP1
ECAP0EN	26	Бит разрешения тактирования блока ECAP0
RTCEN	25	Бит разрешения тактирования блока RTC
ETMR3EN	24	Бит разрешения тактирования блока ETMR3
ETMR2EN	23	Бит разрешения тактирования блока ETMR2
ETMR1EN	22	Бит разрешения тактирования блока ETMR1
ETMR0EN	21	Бит разрешения тактирования блока ETMR0
TMR3EN	20	Бит разрешения тактирования блока TMR3
TMR2EN	19	Бит разрешения тактирования блока TMR2
TMR1EN	18	Бит разрешения тактирования блока TMR1
TMR0EN	17	Бит разрешения тактирования блока TMR0
I2C1EN	12	Бит разрешения тактирования контроллера I2C1
I2C0EN	11	Бит разрешения тактирования контроллера I2C0
TUART1EN	10	Бит разрешения тактирования блока TUART1
TUART0EN	9	Бит разрешения тактирования блока TUART0
–	16-13, 8-0	Зарезервировано

## PCLKCFG1 – регистр разрешения тактовых сигналов периферийных блоков шины APB

Смещение: + E4h

Сброс: 0h

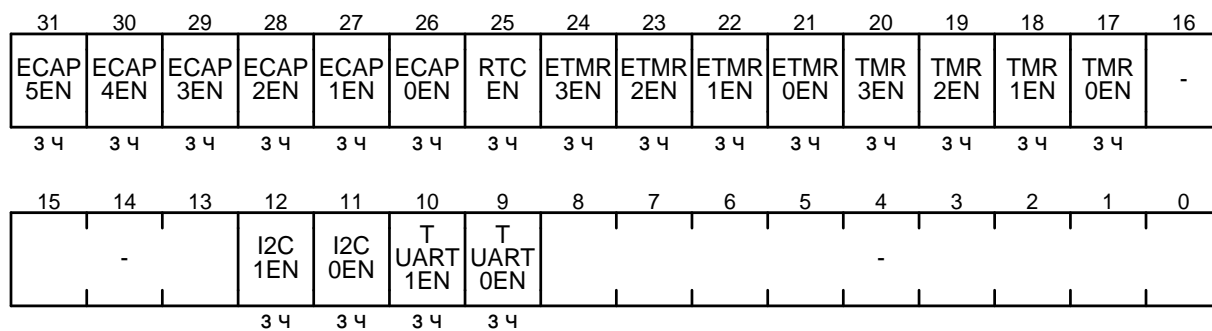


Поле	Биты	Описание
OWI1EN	16	Бит разрешения тактирования блока OWI1
OWI0EN	15	Бит разрешения тактирования блока OWI0
LAUEN	14	Бит разрешения тактирования блока LAU
QEP3EN	13	Бит разрешения тактирования блока QEP3
QEP2EN	12	Бит разрешения тактирования блока QEP2
QEP1EN	11	Бит разрешения тактирования блока QEP1
QEP0EN	10	Бит разрешения тактирования блока QEP0
PWM9EN	9	Бит разрешения тактирования блока PWM9
PWM8EN	8	Бит разрешения тактирования блока PWM8
PWM7EN	7	Бит разрешения тактирования блока PWM7
PWM6EN	6	Бит разрешения тактирования блока PWM6
PWM5EN	5	Бит разрешения тактирования блока PWM5
PWM4EN	4	Бит разрешения тактирования блока PWM4
PWM3EN	3	Бит разрешения тактирования блока PWM3
PWM2EN	2	Бит разрешения тактирования блока PWM2
PWM1EN	1	Бит разрешения тактирования блока PWM1
PWM0EN	0	Бит разрешения тактирования блока PWM0
–	31-17	Зарезервировано

## PRSTCFG0 – регистр включения периферийных блоков шины APB

Смещение: + F0h

Сброс: 0h

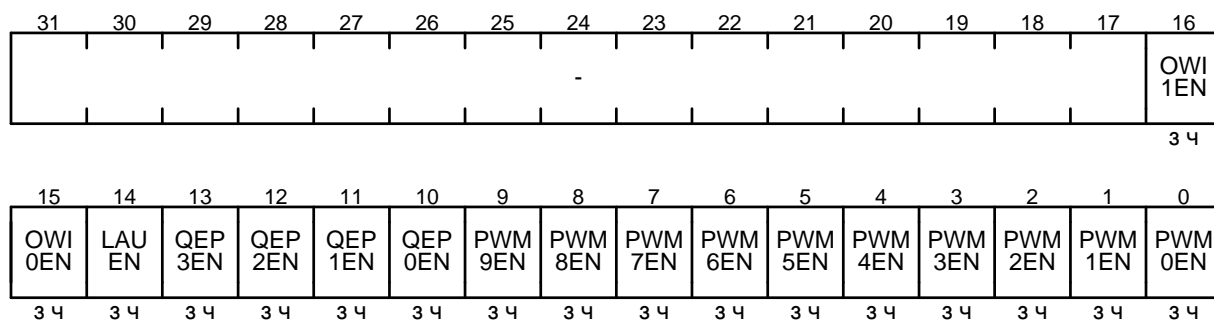


Поле	Биты	Описание
ECAP5EN	31	Бит включения блока ECAP5
ECAP4EN	30	Бит включения блока ECAP4
ECAP3EN	29	Бит включения блока ECAP3
ECAP2EN	28	Бит включения блока ECAP2
ECAP1EN	27	Бит включения блока ECAP1
ECAP0EN	26	Бит включения блока ECAP0
RTCEN	25	Бит включения блока RTC
ETMR3EN	24	Бит включения блока ETMR3
ETMR2EN	23	Бит включения блока ETMR2
ETMR1EN	22	Бит включения блока ETMR1
ETMR0EN	21	Бит включения блока ETMR0
TMR3EN	20	Бит включения блока TMR3
TMR2EN	19	Бит включения блока TMR2
TMR1EN	18	Бит включения блока TMR1
TMR0EN	17	Бит включения блока TMR0
I2C1EN	12	Бит включения блока I2C1
I2C0EN	11	Бит включения блока I2C0
TUART1EN	10	Бит включения блока TUART1
TUART0EN	9	Бит включения блока TUART0
–	16-13, 8-0	Зарезервировано

## PRSTCFG1 – регистр включения периферийных блоков шины APB

Смещение: + F4h

Сброс: 0h

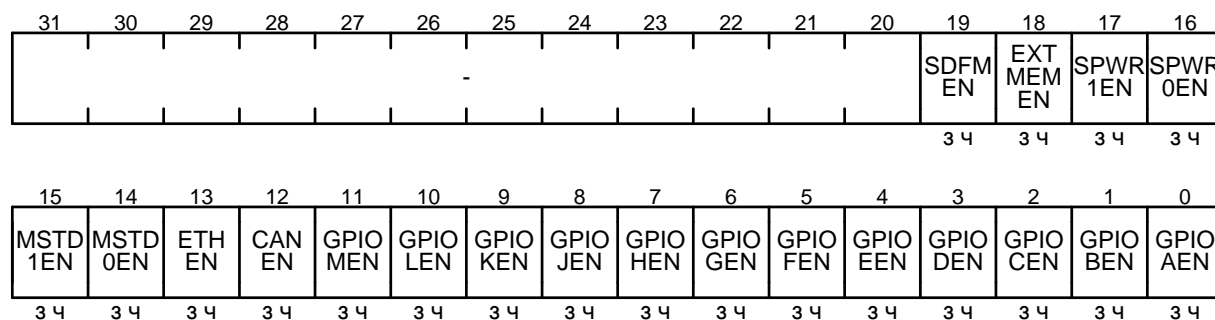


Поле	Биты	Описание
OWI1EN	16	Бит включения блока OWI1
OWI0EN	15	Бит включения блока OWI0
LAUEN	14	Бит включения блока LAU
QEP3EN	13	Бит включения блока QEP3
QEP2EN	12	Бит включения блока QEP2
QEP1EN	11	Бит включения блока QEP1
QEP0EN	10	Бит включения блока QEP0
PWM9EN	9	Бит включения блока PWM9
PWM8EN	8	Бит включения блока PWM8
PWM7EN	7	Бит включения блока PWM7
PWM6EN	6	Бит включения блока PWM6
PWM5EN	5	Бит включения блока PWM5
PWM4EN	4	Бит включения блока PWM4
PWM3EN	3	Бит включения блока PWM3
PWM2EN	2	Бит включения блока PWM2
PWM1EN	1	Бит включения блока PWM1
PWM0EN	0	Бит включения блока PWM0
–	31-17	Зарезервировано

## HCLKCFG – регистр разрешения тактирования периферийных блоков шины АНВ

Смещение: + 100h

Сброс: 0h

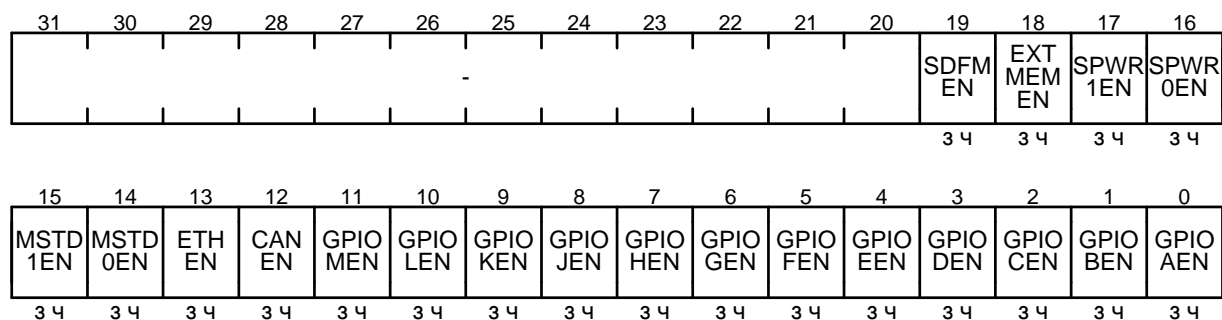


Поле	Биты	Описание
SDFMEN	19	Бит разрешения тактирования SDFM
EXTMEMEN	18	Бит разрешения тактирования EXTMEM
SPWR1EN	17	Бит разрешения тактирования SPWR1
SPWR0EN	16	Бит разрешения тактирования SPWR0
MSTD1EN	15	Бит разрешения тактирования MSTD1
MSTD0EN	14	Бит разрешения тактирования MSTD0
ETHEN	13	Бит разрешения тактирования ETHERNET
CANEN	12	Бит разрешения тактирования CAN
GPIOMEN	11	Бит разрешения тактирования порта M
GPIOLEN	10	Бит разрешения тактирования порта L
GPIOKEN	9	Бит разрешения тактирования порта K
GPIOJEN	8	Бит разрешения тактирования порта J
GPIOHEN	7	Бит разрешения тактирования порта H
GPIOGEN	6	Бит разрешения тактирования порта G
GPIOFEN	5	Бит разрешения тактирования порта F
GPIOEEN	4	Бит разрешения тактирования порта E
GPIODEN	3	Бит разрешения тактирования порта D
GPIOCEN	2	Бит разрешения тактирования порта C
GPIOBEN	1	Бит разрешения тактирования порта B
GPIOAEN	0	Бит разрешения тактирования порта A
–	31-20	Зарезервировано

## HRSTCFG – регистр включения периферийных блоков шины АНВ

Смещение: + 104h

Сброс: 0h



Поле	Биты	Описание
SDFMEN	19	Бит включения SDFM
EXTMEMEN	18	Бит включения EXTMEM
SPWR1EN	17	Бит включения SPWR1
SPWR0EN	16	Бит включения SPWR0
MSTD1EN	15	Бит включения MSTD1
MSTD0EN	14	Бит включения MSTD0
ETHEN	13	Бит включения ETHERNET
CANEN	12	Бит включения CAN
GPIOMEN	11	Бит включения порта M
GPIOLEN	10	Бит включения порта L
GPIOKEN	9	Бит включения порта K
GPIOJEN	8	Бит включения порта J
GPIOHEN	7	Бит включения порта H
GPIOGEN	6	Бит включения порта G
GPIOFEN	5	Бит включения порта F
GPIOEEN	4	Бит включения порта E
GPIODEN	3	Бит включения порта D
GPIOCEN	2	Бит включения порта C
GPIOBEN	1	Бит включения порта B
GPIOAEN	0	Бит включения порта A
–	31-20	Зарезервировано

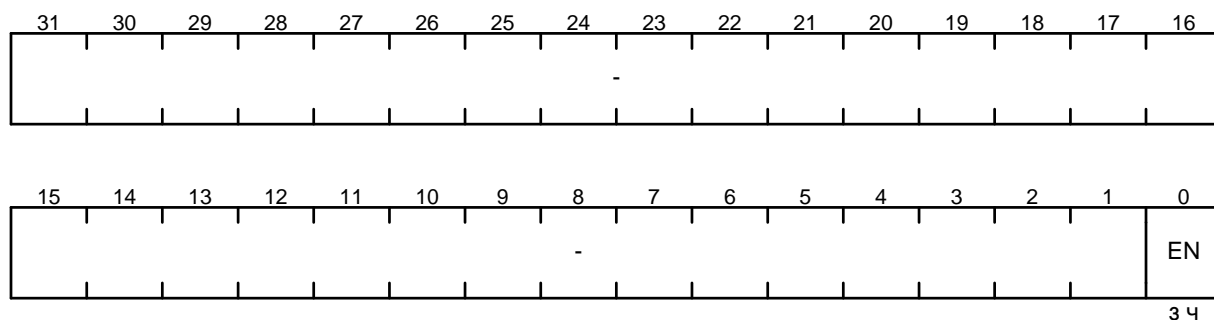
### А.3 Регистры блока управления энергопотреблением PMU

Базовый адрес: 4008\_2000h

#### CFG – регистр настройки PMU

Смещение: + 00h

Сброс: 0h

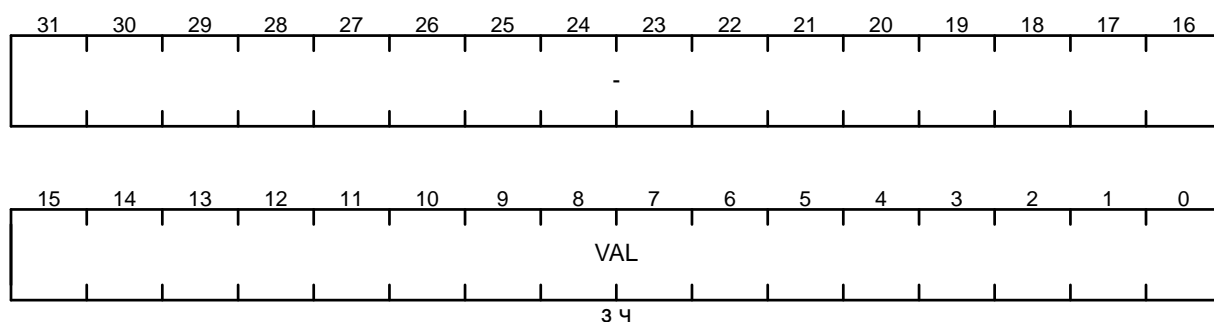


Поле	Биты	Описание
EN	0	Бит включения PMU
		0   Выключено
		1   Включено
–	31-1	Зарезервировано

#### PUDEL – регистр задания задержки пробуждения блоков

Смещение: + 04h

Сброс: 0000\_02C0h

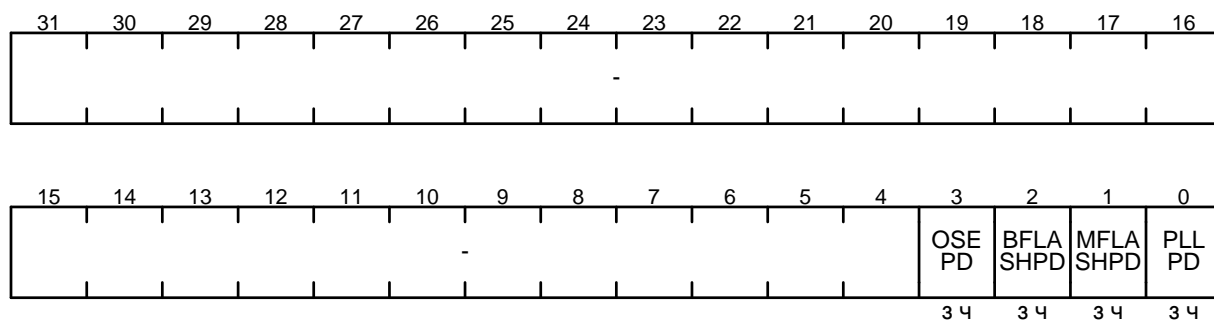


Поле	Биты	Описание
VAL	15-0	Значение задержки пробуждения для периферийных блоков в тактах REFCLK
–	31-16	Зарезервировано

## PDEN – регистр управления режимом Powerdown блоков

Смещение: + 08h

Сброс: 0000\_000Fh



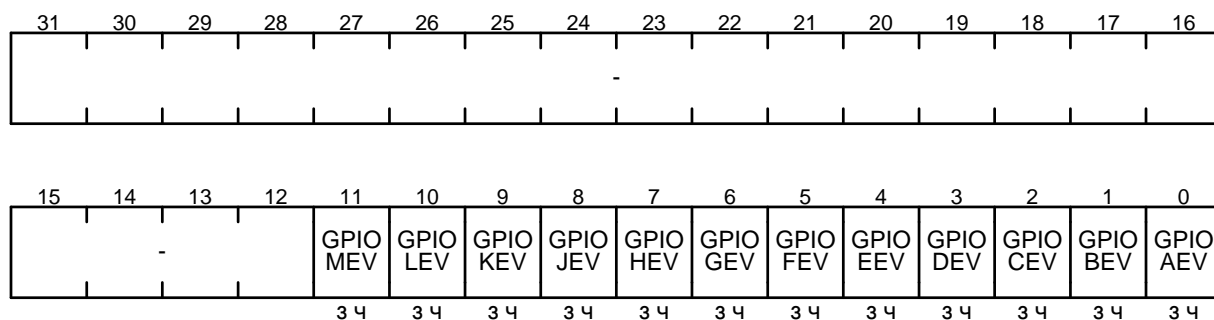
Поле	Биты	Описание
OSEPD	3	Бит включения режима Powerdown для внешнего осциллятора
		0   Выключено
		1   Включено
BFLASHPD	2	Бит включения режима Powerdown для BFLASH
		0   Выключено
		1   Включено
MFLASHPD	1	Бит включения режима Powerdown для MFLASH
		0   Выключено
		1   Включено
PLLPD	0	Бит включения режима Powerdown для PLL
		0   Выключено
		1   Включено
–	31-4	Зарезервировано



## RXEVEN – регистр разрешения событий RXEV

Смещение: + 0Ch

Сброс: 0h



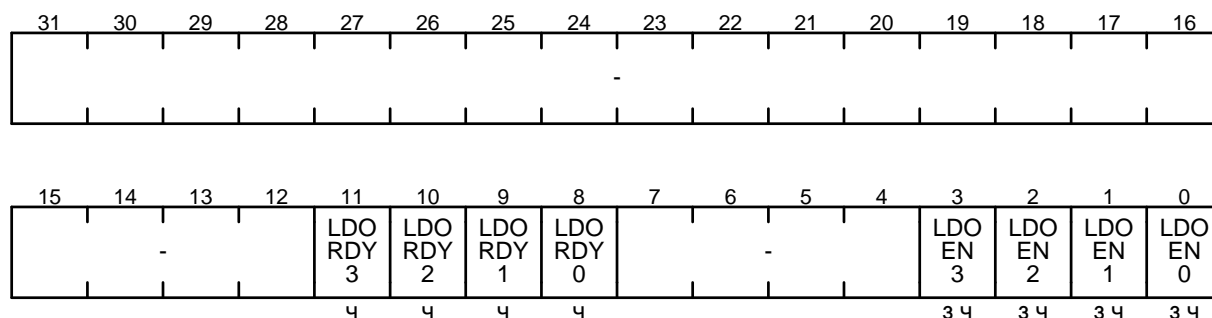
Поле	Биты	Описание
GPIO MEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO M
GPIO LEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO L
GPIO KEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO K
GPIO JEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO J
GPIO HEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO H
GPIO GEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO G
GPIO FEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO F
GPIO EEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO E
GPIO DEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO D
GPIO CEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO C
GPIO BEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO B
GPIO AEV	1	Бит разрешения выхода из режима сна по RXEV от GPIO A
–	31-12	Зарезервировано

Примечание – Установленный бит разрешает обработку события для выхода из режима сна, сброшенный - запрещает.

## ADCPC – регистр управления питанием АЦП

Смещение: + 10h

Сброс: 0h



Поле	Биты	Описание		
LDORDY <sub>m</sub>	3-0	Флаг готовности LDO преобразователя внутри модуля АЦП m (от 0 до 3)		
		<table border="1"> <tr> <td>0</td> <td>LDO выключен</td> </tr> <tr> <td>1</td> <td>LDO включился и стабильно работает</td> </tr> </table>	0	LDO выключен
0	LDO выключен			
1	LDO включился и стабильно работает			
LDOEN <sub>m</sub>	3-0	Бит включения LDO преобразователя внутри модуля АЦП m (от 0 до 3)		
		<table border="1"> <tr> <td>0</td> <td>LDO преобразователь выключен</td> </tr> <tr> <td>1</td> <td>LDO преобразователь включен</td> </tr> </table>	0	LDO преобразователь выключен
0	LDO преобразователь выключен			
1	LDO преобразователь включен			

#### А.4 Регистры контроллера основной Flash-памяти

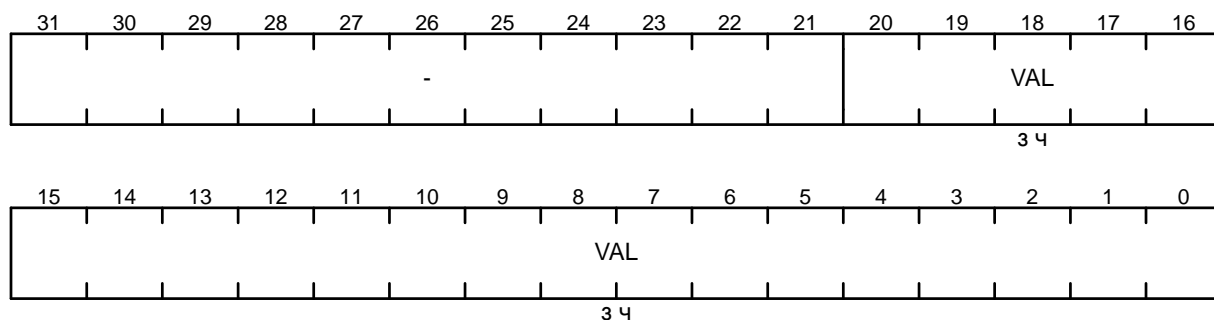
Базовый адрес: 4006\_0000h

Смещение: + 04h (DATA)      Регистры данных

##### ADDR – регистр адреса Flash-памяти

Смещение: + 00h

Сброс: 0h

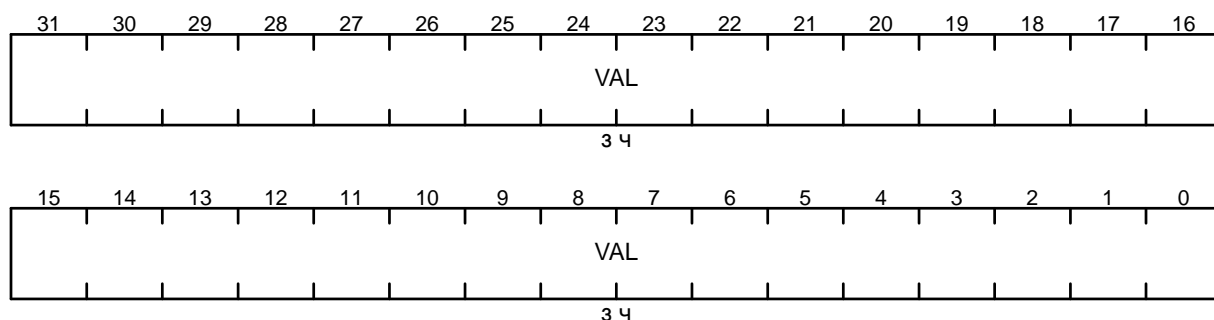


Поле	Биты	Описание
VAL	20-0	Адрес, используемый при командах записи, чтения и постраничного стирания
	31-21	Зарезервировано

##### DATA – массив регистров данных Flash-памяти

Смещение: DATA + (4\*d)h, где d = 0, ..., 15

Сброс: FFFF\_FFFFh

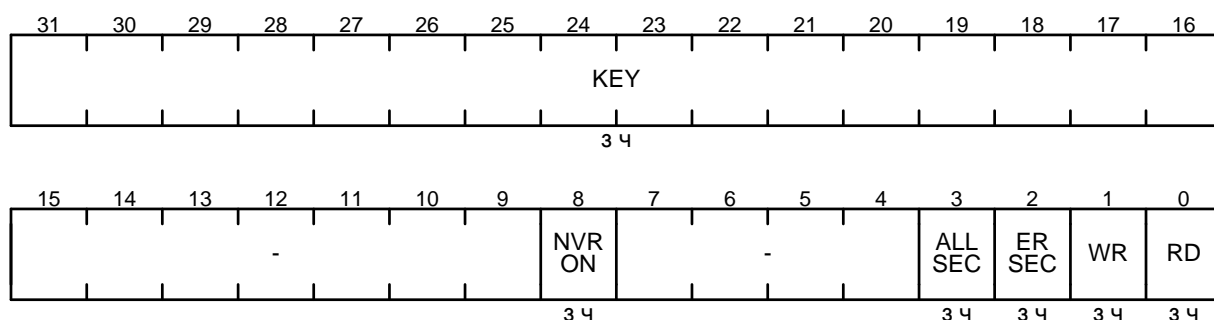


Поле	Биты	Описание
VAL	31-0	32-разрядные регистры слов данных. Все слова данных должны быть загружены в регистры до установки бита команды записи. Читаемые данные будут доступны в регистрах после сброса флага BUSY.

## CMD – регистр команд Flash-памяти

Смещение: + 44h

Сброс: DEC0\_0000h

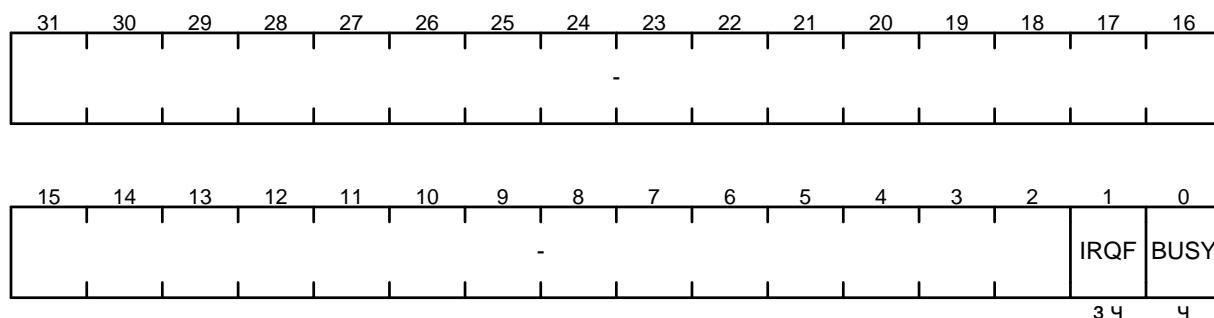


Поле	Биты	Описание
KEY	31-16	Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле KEY значения CODEh. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. При одновременной записи нескольких команд будет выполнена та, номер бита которой меньше. Чтение поля KEY всегда возвращает DEC0h.
NVRON	8	Бит модификации команды для работы с NVR областью
		0   Команда выполняется для основной области Flash-памяти 1   Команда выполняется для NVR области Flash-памяти. Примечание – Исключением является ситуация, когда одновременно с битом NVRON будут установлены биты ALLSEC и ERSEC. Вэтом случае будут стерты полностью как основная область, так и NVR.
ALLSEC	3	Бит модификации команды стирания. При установленных битах ALLSEC и ERSEC происходит полное стирание области.
ERSEC	2	Бит активации команды стирания страницы области. Адрес страницы вычисляется на основе значения регистра ADDR.
WR	1	Бит активации команды записи данных DATA <sub>d</sub> (d от 0 до 15), начиная с адреса ADDR в области
RD	0	Бит активации команды чтения данных в DATA <sub>d</sub> (d от 0 до 15), начиная с адреса ADDR в области
–	15-9, 7-4	Зарезервировано

## STAT – регистр статуса Flash-памяти

Смещение: + 48h

Сброс: 0h

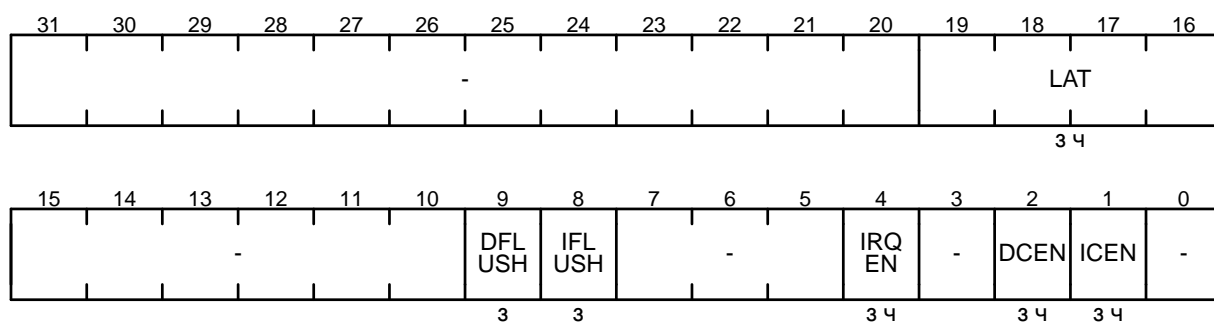


Поле	Биты	Описание
IRQF	1	Флаг прерывания по окончании выполнения команды. Устанавливается только если бит <b>IRQEN</b> установлен.
		0   Нет информации
		1   Команда выполнена
		Сбрасывается записью «1».
BUSY	0	Статус работы контроллера Flash-памяти
		0   Нет активной команды
		1   Выполняется команда
		Примечание – В связи с особенностями пересинхронизации, при работе на высоких частотах ядра необходимо добавлять задержку между записью регистра <b>CMD</b> и чтением флага <b>BUSY</b> , например, 5 <b>NOP</b> команд.
–	31-2	Зарезервировано

## CTRL – регистр настройки контроллера Flash-памяти

Смещение: + 4Ch

Сброс: 1\_0000h

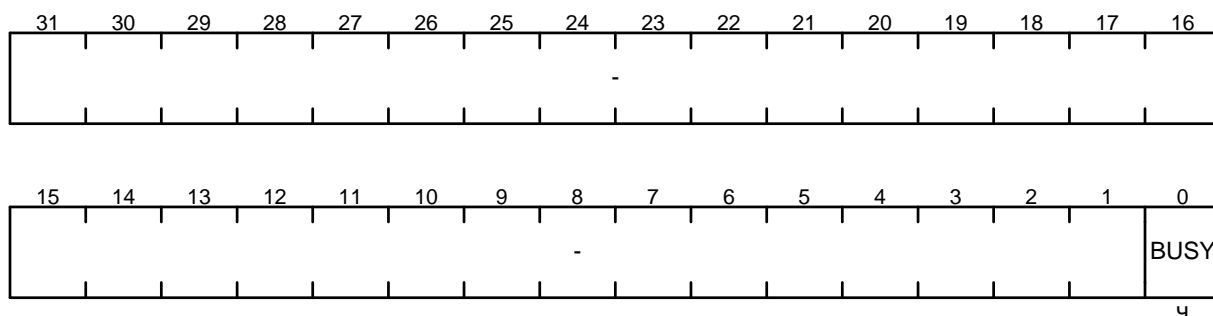


Поле	Биты	Описание
LAT	19-16	Поле задания количества дополнительных тактов ожидания при чтении из Flash
DFLUSH	9	0   Нет реакции
		1   Запуск очистки
IFLUSH	8	0   Нет реакции
		1   Запуск очистки
IRQEN	4	0   Нет реакции
		1   Прерывание разрешено
DCEN	2	0   Выключен
		1   Включен
ICEN	1	0   Выключен
		1   Включен
–	31-20, 15-10, 7-5, 3,0	Зарезервировано

### ICSTAT – регистр статуса кэша инструкций

Смещение: + 50h

Сброс: 0h

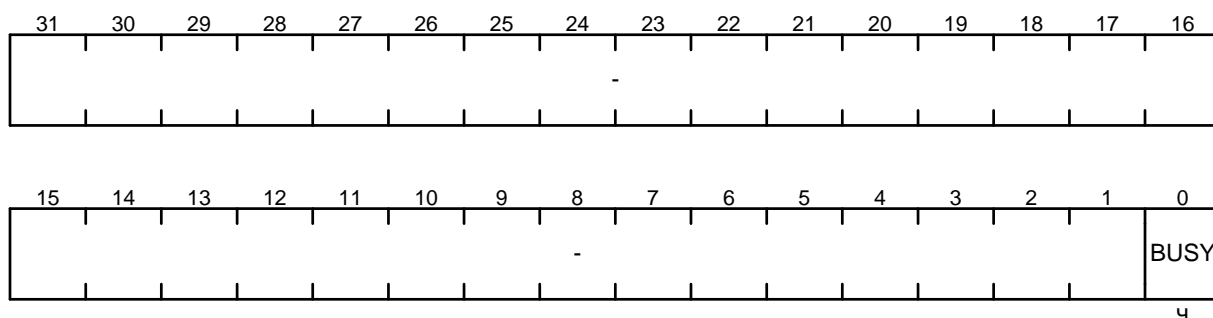


Поле	Биты	Описание
BUSY	0	Флаг устанавливается при запуске команды очистки кэша данных, сбрасывается после её окончания
–	31-1	Зарезервировано

### DCSTAT – регистр статуса кэша данных

Смещение: + 54h

Сброс: 0h



Поле	Биты	Описание
BUSY	0	Флаг устанавливается при запуске команды очистки кэша данных, сбрасывается после её окончания
–	31-1	Зарезервировано

## A.5 Регистры контроллера загрузочной Flash-памяти

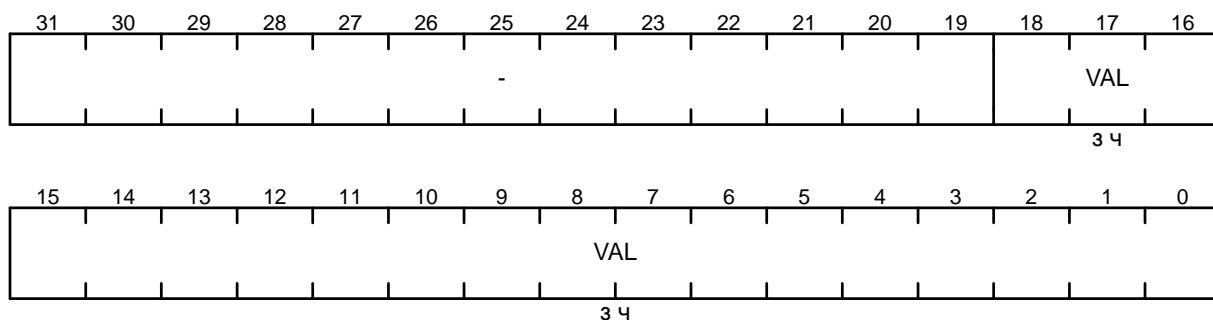
Базовый адрес: 4006\_1000h

Смещение: + 04h (DATA)      Регистры данных

### ADDR – регистр адреса Flash-памяти

Смещение: + 00h

Сброс: 0h

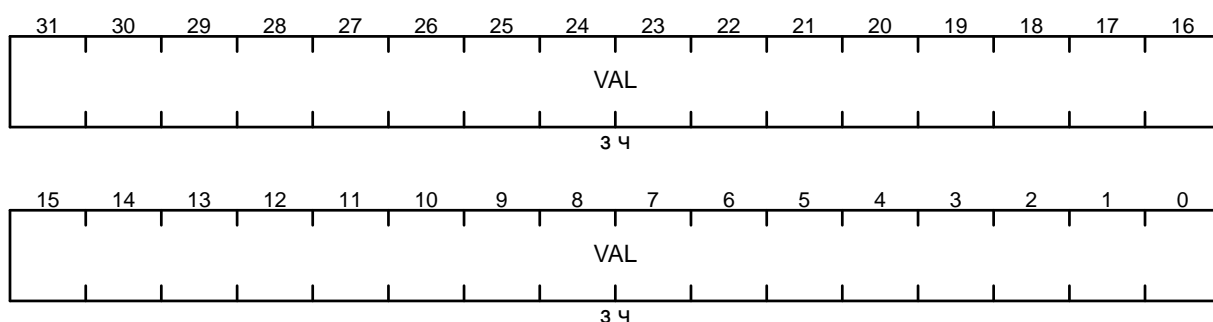


Поле	Биты	Описание
VAL	18-0	Адрес, используемый при командах записи, чтения и постраничного стирания
	31-19	Зарезервировано

### DATA – массив регистров данных Flash-памяти

Смещение: DATA + (4\*d)h, где d = 0, ..., 3

Сброс: FFFF\_FFFFh



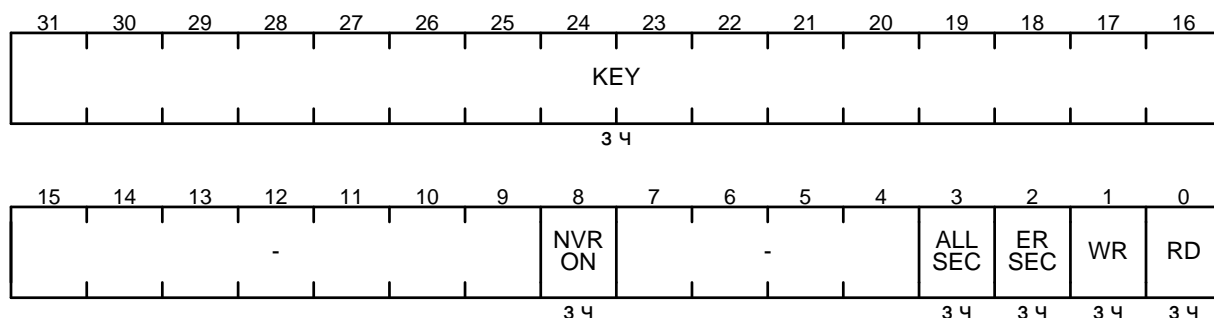
Поле	Биты	Описание
VAL	31-0	32-разрядные регистры слов данных. Все слова данных должны быть загружены в регистры до установки бита команды записи. Читаемые данные будут доступны в регистрах после сброса флага BUSY.



## CMD – регистр команд Flash-памяти

Смещение: + 44h

Сброс: DEC0\_0000h

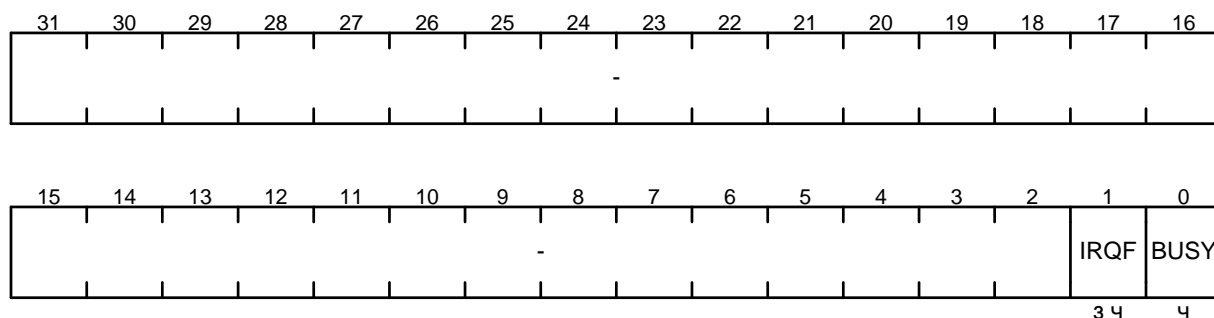


Поле	Биты	Описание
KEY	31-16	Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле KEY значения CODEh. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. При одновременной записи нескольких команд будет выполнена та, номер бита которой меньше. Чтение поля KEY всегда возвращает DEC0h.
NVRON	8	Бит модификации команды для работы с NVR областью
		0   Команда выполняется для основной области Flash-памяти 1   Команда выполняется для NVR области Flash-памяти. Примечание – Исключением является ситуация, когда одновременно с битом NVRON будут установлены биты ALLSEC и ERSEC. В этом случае будут стерты полностью как основная область, так и NVR.
ALLSEC	3	Бит модификации команды стирания. При установленных битах ALLSEC и ERSEC происходит полное стирание области.
ERSEC	2	Бит активации команды стирания страницы области. Адрес страницы вычисляется на основе значения регистра ADDR.
WR	1	Бит активации команды записи данных DATA <sub>d</sub> (d от 0 до 3), начиная с адреса ADDR в области
RD	0	Бит активации команды чтения данных в DATA <sub>d</sub> (d от 0 до 3), начиная с адреса ADDR в области
–	15-9, 7-4	Зарезервировано

## STAT – регистр статуса Flash-памяти

Смещение: + 48h

Сброс: 0h

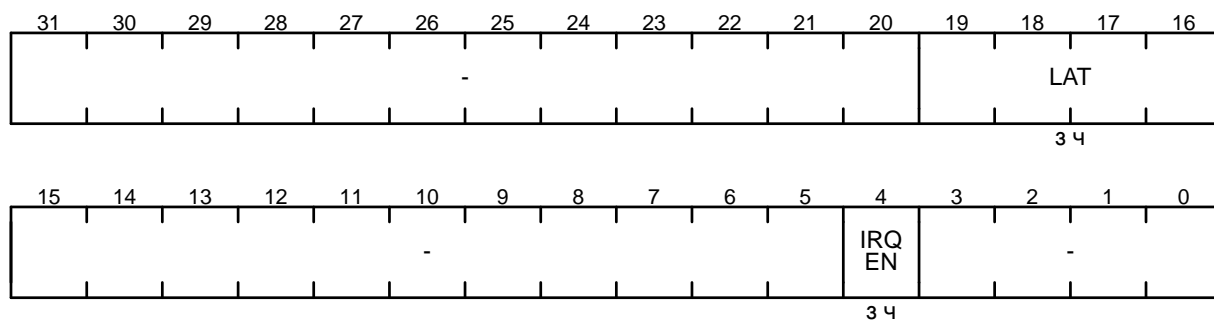


Поле	Биты	Описание
IRQF	1	Флаг прерывания по окончании выполнения команды. Устанавливается только если бит <b>IRQEN</b> установлен.
		0   Нет информации
		1   Команда выполнена Сбрасывается записью «1».
BUSY	0	Статус работы контроллера Flash-памяти
		0   Нет активной команды
		1   Выполняется команда Примечание – В связи с особенностями пересинхронизации, при работе на высоких частотах ядра необходимо добавлять задержку между записью регистра <b>CMD</b> и чтением флага <b>BUSY</b> , например, 5 <b>NOP</b> команд.
–	31-2	Зарезервировано

## CTRL – регистр настройки контроллера Flash-памяти

Смещение: + 4Ch

Сброс: 1\_0000h



Поле	Биты	Описание
LAT	19-16	Поле задания количества дополнительных тактов ожидания при чтении из Flash
IRQEN	4	Бит управления прерыванием по завершению выполнения команды регистра CMD
		0   Нет реакции
		1   Прерывание разрешено
–	31-20, 15-5, 3	Зарезервировано

## А.6 Регистры контроллера внешней памяти

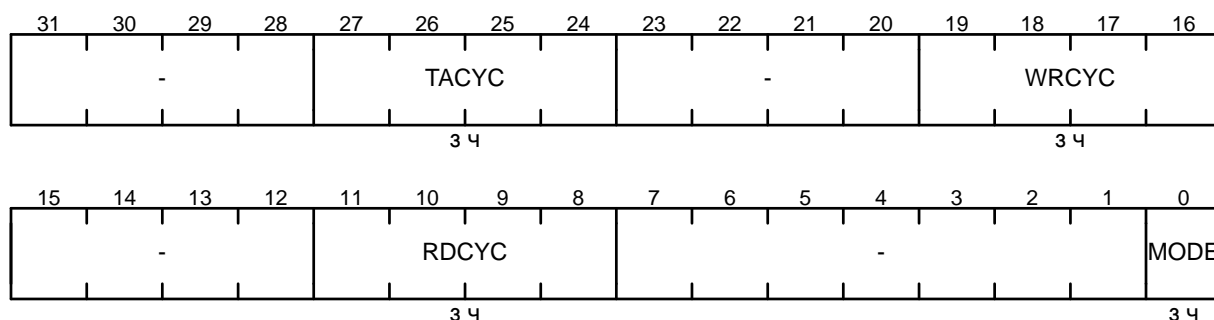
Базовый адрес: 4006\_2000h

Примечание – d –

### WINCFG – массив регистров настройки контроллера внешней памяти

Смещение: (4\*d)h, номер окна от 0 до 7

Сброс: 0707\_0701h



Поле	Биты	Описание
TACYC	27-24	Длительность цикла переключения шины. Поле задает паузу между циклами чтения и записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати
WRCYC	19-16	Длительность цикла записи слова данных. Поле задает длительность цикла записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати
RDCYC	11-8	Длительность цикла чтения слова данных. Поле задает длительность цикла чтения на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати
MODE	0	Бит задания разрядности контроллера внешней памяти
		0   8-разрядный режим работы
		1   16-разрядный режим работы
–	31-28, 23-20, 16-12, 7-1	Зарезервировано

## A.7 Регистры контроллера прямого доступа к памяти DMA

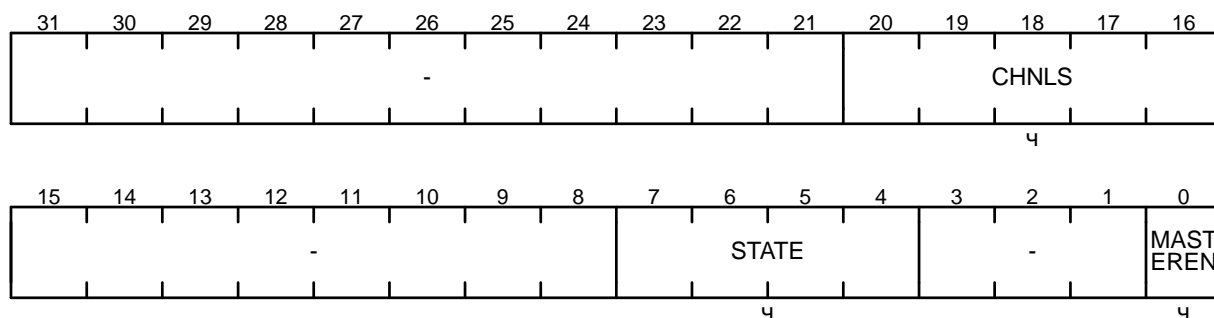
**Базовый адрес:** 4008\_4000h

**Примечание:** i – порядковый номер канала от 0 до 31

### STATUS – регистр статуса DMA

Смещение: + 00h

Сброс: 0000\_0000h



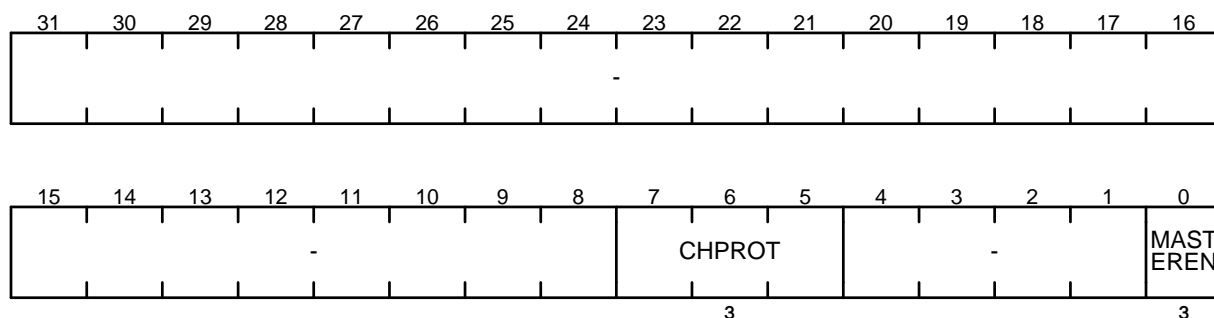
Поле	Биты	Описание
CHNLS	20-16	Количество доступных каналов DMA
		00h   1 канал
		01h   2 канала
		02h   3 канала
		...   ...
		1Fh   32 канала
STATE	7-4	Текущее состояние конечного автомата управления контроллера
		0h   В покое
		1h   Чтение управляющих данных канала
		2h   Чтение указателя конца данных источника
		3h   Чтение указателя конца данных приемника
		4h   Чтение данных источника
		5h   Запись данных в приемник
		6h   Ожидание запроса на выполнение прямого доступа
		7h   Запись управляющих данных канала
		8h   Приостановлен
		9h   Выполнен
		Ah   Режим работы с периферией «разборка-сборка»
Bh-Fh   Зарезервировано		
MASTEREN	0	Состояние контроллера DMA
		0   Работа контроллера запрещена
		1   Работа контроллера разрешена
–	31-21, 15-8, 3-1	Зарезервировано

Примечание – Регистр доступен только для чтения. Возвращает состояние контроллера DMA. Во время сброса чтение регистра запрещено.

## CFG – регистр конфигурации контроллера DMA

Смещение: + 04h

Сброс: 0h

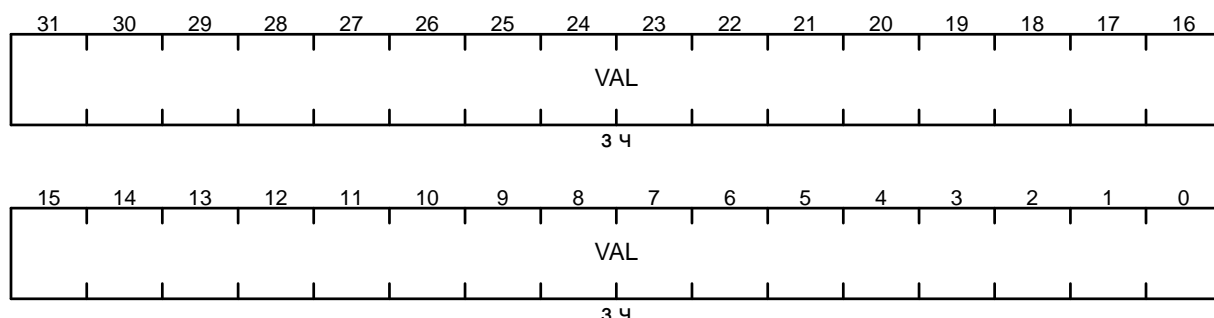


Поле	Биты	Описание			
CHPROT	7-5	Задаёт параметры защиты шины АНВ при обращении контроллера DMA к структурам управляющих данных каналов			
		Биты поля CHPROT			
			7	6	5
		0	Доступ не кэшируется	Доступ не буферизуется	Доступ непривилегированный
1	Доступ кэшируется	Доступ буферизуется	Доступ привилегированный		
MASTEREN	0	Бит разрешения работы контроллера DMA			
		0	Запрещена		
		1	Разрешена		
–	31-8, 4-1	Зарезервировано			

## BASEPTR – регистр базового адреса управляющих данных каналов

Смещение: + 08h

Сброс: 0h



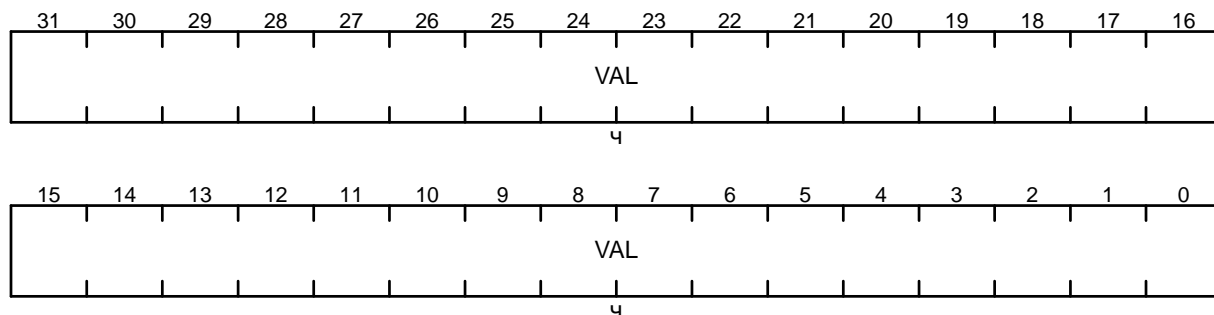
Поле	Биты	Описание
VAL	31-0	Указатель на базовый адрес первичной структуры управляющих данных

Примечание – Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов. Во время сброса чтение регистра запрещено.

## ALTBASEPTR – регистр базового адреса альтернативных управляющих данных каналов

Смещение: + 0Ch

Сброс: 0h



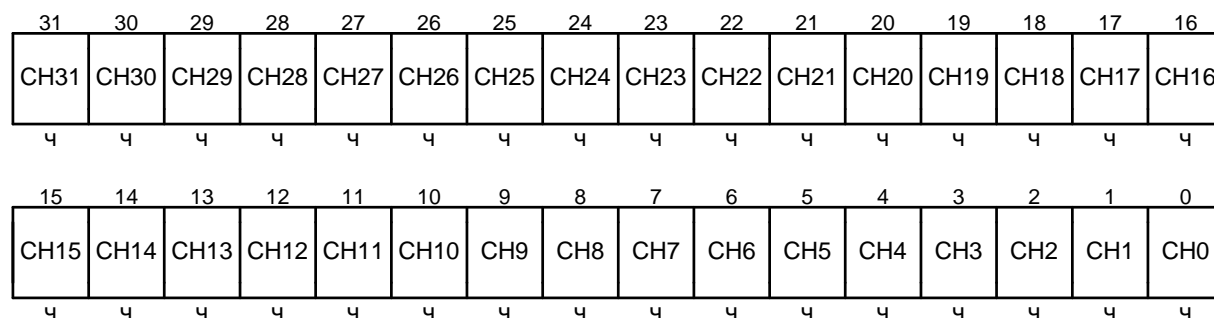
Поле	Биты	Описание
VAL	31-0	Указатель базового адреса альтернативной структуры управляющих данных каналов

Примечание – Регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов. Доступен только для чтения и возвращает указатель базового адреса альтернативных управляющих данных каналов. Во время сброса чтение регистра запрещено.

## WAITONREQ – регистр статуса ожидания запросов для передачи

Смещение: + 10h

Сброс: 0h



Поле	Биты	Описание		
CHi	31-0	Чтение	0	Доступны только BREQ запросы от периферии
			1	Доступны BREQ и SREQ запросы от периферии
		Запись	Не выполняется	

## SWREQ – регистр программного запроса на обработку каналов DMA

Смещение: + 14h

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Поле	Биты	Описание	
CHi	31-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Устанавливает запрос на выполнение цикла DMA по каналу i

## USEBURSTSET – регистр установки пакетного обмена каналов DMA

Смещение: + 18h

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4	3 4

Поле	Биты	Описание		
CHi	31-0	Бит отключения выполнения одиночных запросов SREQ. При активации режима будут обрабатываться и исполняться только пакетные запросы BREQ		
		Чтение	0	Выполнение циклов DMA в ответ на SREQ и BREQ
			1	Выполнение циклов DMA только в ответ на BREQ
		Запись нуля	Не выполняется	
Запись единицы	Отключает выполнение запросов SREQ			



## USEBURSTCLR – регистр сброса пакетного обмена каналов DMA

Смещение: + 1Ch

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Поле	Биты	Описание	
CHi	31-0	Чтение	Читаются нули
		Запись нуля	Не выполняется
		Запись единицы	Разрешает обрабатывать одиночные запросы SREQ на выполнение циклов DMA

## REQMASKSET – регистр маскирования запросов от периферии на обслуживание каналов DMA

Смещение: + 20h

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч

Поле	Биты	Описание		
CHi	31-0	Бит отключения выполнения каналом i циклов DMA в ответ на поступающие запросы SREQ и BREQ периферии		
		Чтение	0	Канал i выполняет циклы
			1	Канал i не выполняет циклы
		Запись нуля	Не выполняется	
Запись единицы	Отключает выполнение циклов			

## REQMASKCLR – регистр сброса маскирования запросов на обслуживание каналов DMA

Смещение: + 24h

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Поле	Биты	Описание	
CHi	31-0	Чтение	Читаются нули
		Запись нуля	Не выполняется
		Запись единицы	Разрешает выполнение циклов DMA по запросам SREQ и BREQ периферии

## ENSET – регистр установки разрешения работы каналов DMA

Смещение: + 28h

Сброс: 0h

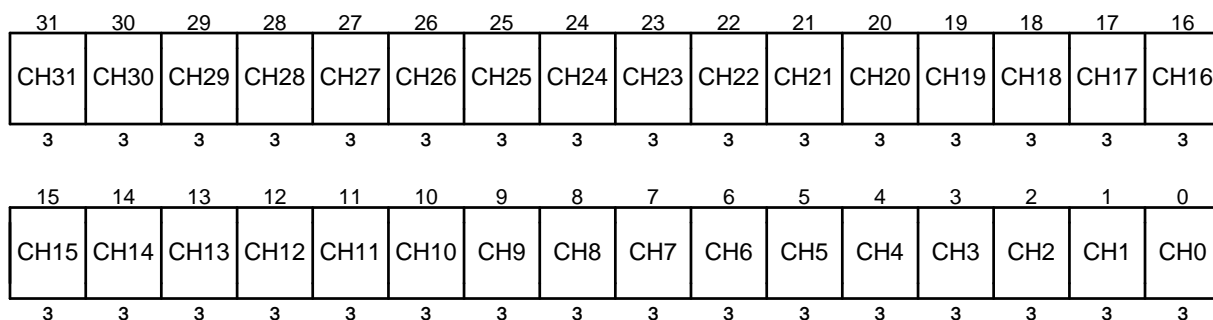
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч

Поле	Биты	Описание		
CHi	31-0	Чтение	0	Канал i отключен
			1	Работа канала i разрешена
		Запись нуля	Не выполняется	
			Запись единицы	Разрешает работу канала i

## ENCLR – регистр сброса разрешения работы каналов DMA

Смещение: + 2Ch

Сброс: 0h



Поле	Биты	Описание	
CHi	31-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Запрещает работу канала i

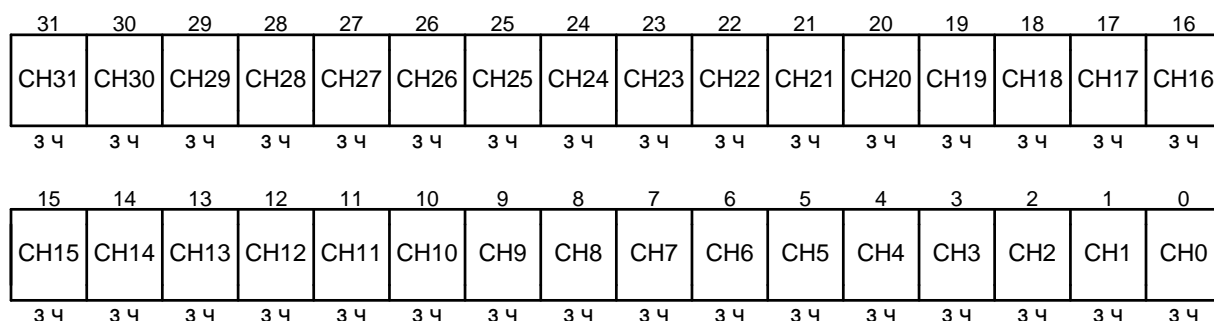
Примечание – Контроллер может отключить канал i в случае:

- завершения цикла DMA;
- чтения CHANNEL\_CFG с полем CYCLE\_CTRL, значение которого 000b;
- появления ошибки на шине АHB.

## PRIALTSET – регистр установки первичной/альтернативной структуры управляющих данных каналов DMA

Смещение: + 30h

Сброс: 0h



Поле	Биты	Описание		
CH <sub>i</sub>	31-0	Бит включения использования каналом <i>i</i> альтернативной структуры управляющих данных		
		Чтение	0	Используется первичная структура
			1	Используется альтернативная структура
		Запись нуля	Не выполняется	
Запись единицы	Включает использование альтернативной структуры управляющих данных			

Примечание – Контроллер может переключить состояние бита CH<sub>i</sub> в случае:

- завершения четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «разборка-сборка»;

- завершения всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»;

- завершения всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах «пинг-понг» и «разборка-сборка».

## PRIALTCLR – регистр сброса первичной/альтернативной структуры управляющих данных каналов DMA

Смещение: + 34h

Сброс: 0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Поле	Биты	Описание	
CHi	31 -0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Включает использование первичной структуры управляющих данных канала i

## PRIORITYSET – регистр установки приоритета каналов DMA

Смещение: + 38h

Сброс: 0h

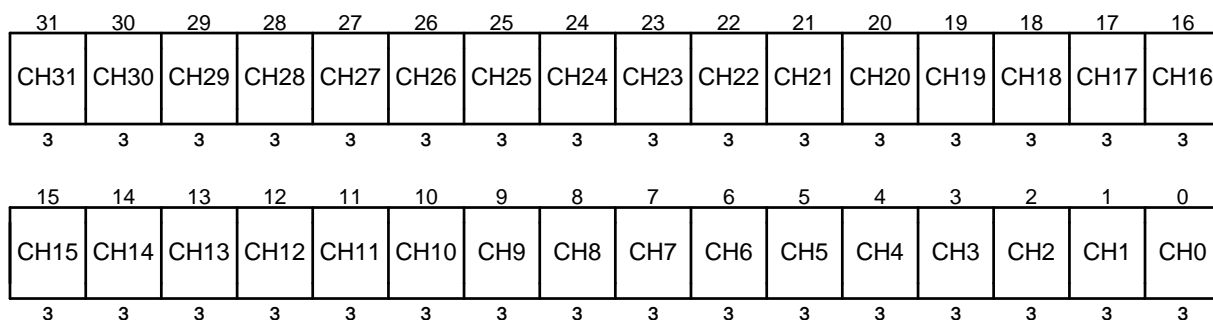
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч

Поле	Биты	Описание		
CHi	31-0	Чтение	0	Каналу i присвоен уровень приоритета по умолчанию
			1	Каналу i присвоен высокий уровень приоритета
		Запись нуля	Не выполняется	
		Запись единицы	Присваивает каналу i высокий уровень приоритета	

## PRIORITYCLR – регистр сброса установок приоритета каналов DMA

Смещение: + 3Ch

Сброс: 0h

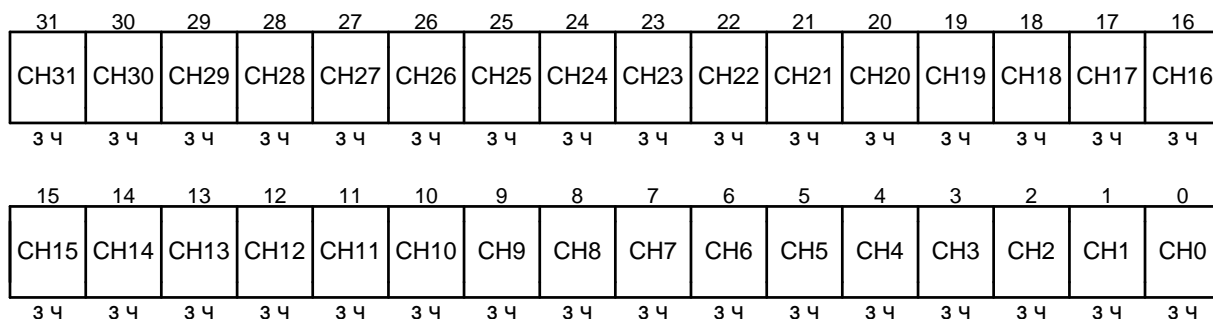


Поле	Биты	Описание	
CHi	31-0	Чтение	Читаются нули
		Запись нуля	Не выполняется
		Запись единицы	Присваивает каналу i уровень приоритета по умолчанию

## CIRCULARSET– регистр установки циклического режима каналов DMA

Смещение: + 40h

Сброс: 0h

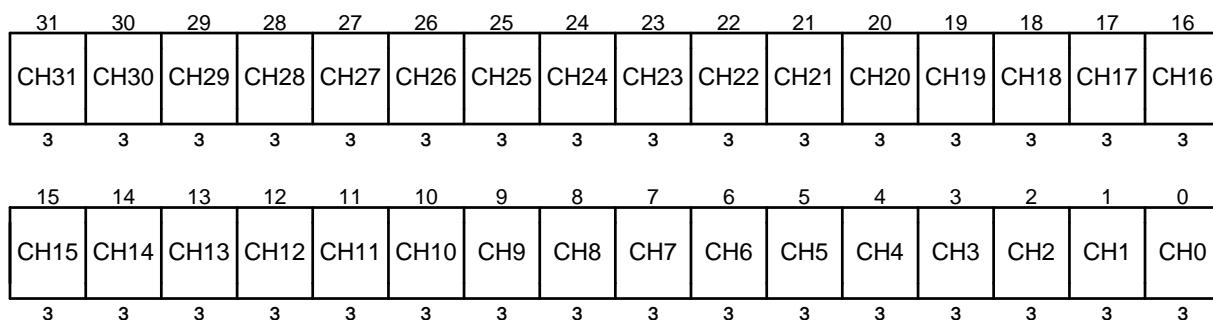


Поле	Биты	Описание		
CHi	31-0	Чтение	0	Канал i функционирует в обычном режиме
			1	Каналу i функционирует в циклическом режиме
		Запись нуля	Не выполняется	
		Запись единицы	Устанавливает циклический режим работы по каналу i	

## CIRCULARCLR – регистр сброса циклического режима каналов DMA

Смещение: + 44h

Сброс: 0h

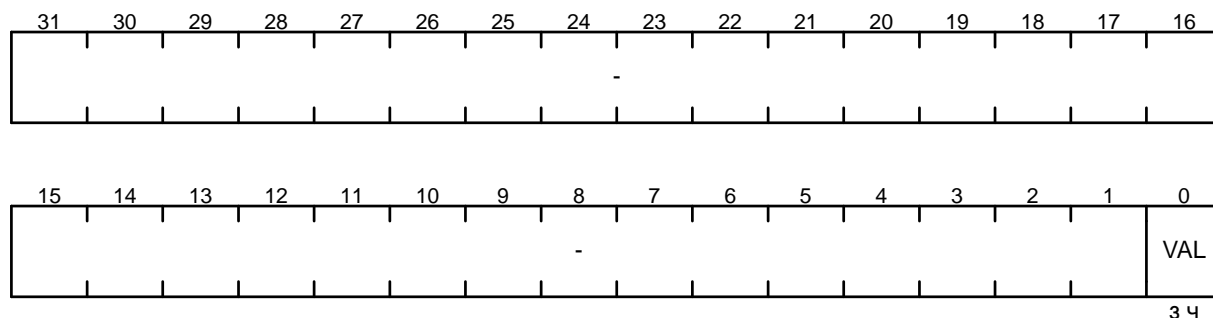


Поле	Биты	Описание	
CHi	31-0	Чтение	Читаются нули
		Запись нуля	Не выполняется
		Запись единицы	Отключает циклический режим работы по каналу i

## ERRCLR – регистр сброса флага ошибки DMA

Смещение: + 4Ch

Сброс: 0h



Поле	Биты	Описание	
VAL	0	Флаг ошибки на шине АHB	
		Чтение	0   Ошибок не обнаружено
			1   Произошла ошибка
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает флаг ошибки VAL
–	31-1	Зарезервировано	

Примечание – При одновременном сбросе флага VAL и появлении ошибки на шине АHB приоритет отдается ошибке, и бит VAL остается установленным

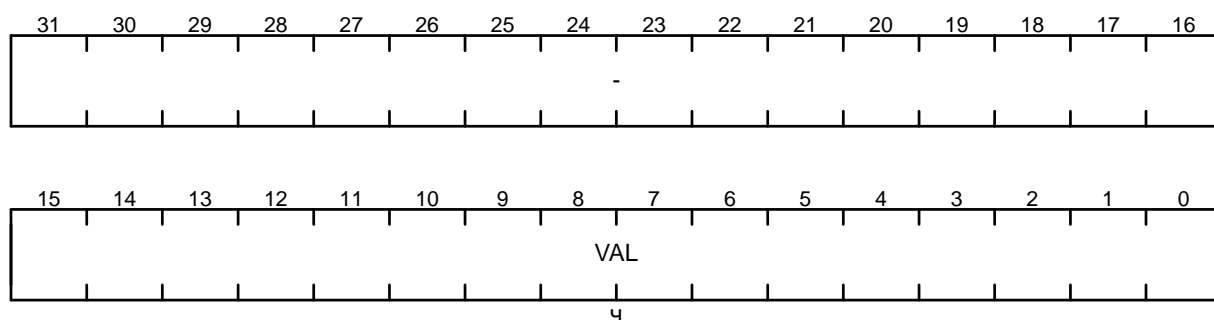
## A.8 Регистры портов ввода-вывода

<b>Базовый адрес:</b>	4001_0000h 4001_1000h 4001_2000h 4001_3000h 4001_4000h 4001_5000h 4001_6000h 4001_7000h 4001_8000h 4001_9000h 4001_A000h 4001_B000h	Регистры порта GPIOA; Регистры порта GPIOB; Регистры порта GPIOC; Регистры порта GPIOD; Регистры порта GPIOE; Регистры порта GPIOF; Регистры порта GPIOG; Регистры порта GPIOH; Регистры порта GPIOJ; Регистры порта GPIOK; Регистры порта GPIOL; Регистры порта GPIOM;
<b>Смещение:</b>	+ 400h (MASKLB) + 800h (MASKHB)	Массив регистров масок младшего байта порта; Массив регистров масок старшего байта порта
<b>Мнемоника:</b>	GPIOp	

Примечание – p – имя порта A, B, C, D, E, F, G, H, J, K, L или M;  
 n – порядковый номер вывода порта от 0 до 15.

### DATA – регистр входных данных порта

Смещение: + 00h  
 Сброс: 0000xxxxh



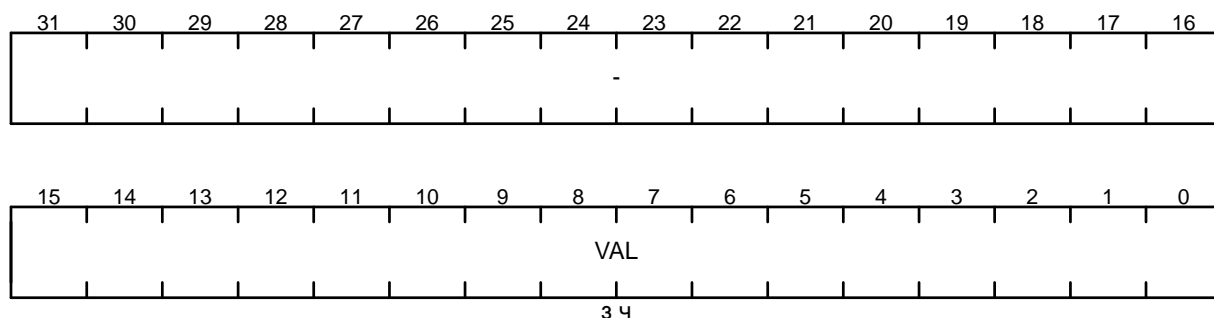
Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает текущее состояние порта
		Запись	Не выполняется
–	31-16	Зарезервировано	



## DATAOUT – регистр выходных данных порта

Смещение: + 04h

Сброс: 0h

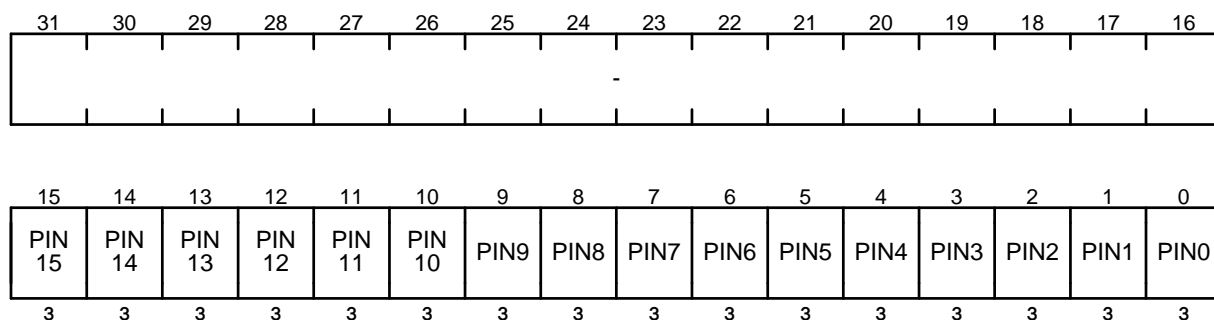


Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние порта
		Запись	Устанавливает на выводах порта уровень сигнала, соответствующий записанному в биты значению
–	31-16	Зарезервировано	

## DATAOUTSET – регистр установки битов порта

Смещение: + 08h

Сброс: 0h

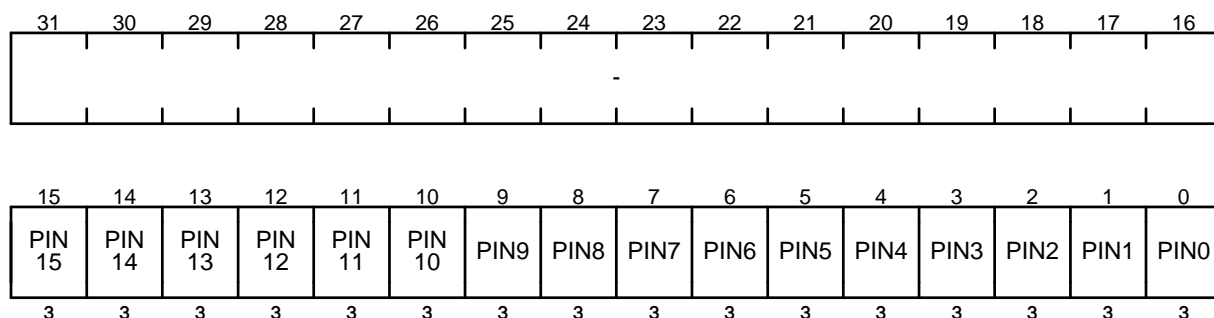


Поле	Биты	Описание	
PINn	15-0	Чтение	Не допускается
		Запись нуля	Не выполняется
		Запись единицы	Устанавливает соответствующий бит n в регистре DATAOUT и, как следствие, высокий уровень сигнала на выводе n порта
–	31-16	Зарезервировано	

## DATAOUTCLR – регистр сброса битов порта

Смещение: + 0Ch

Сброс: 0h

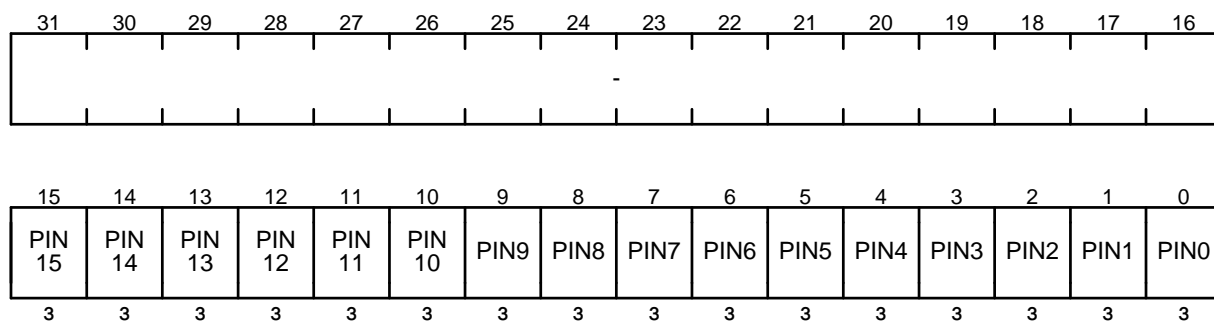


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре DATAOUT и, как следствие, устанавливает низкий уровень сигнала на выводе n порта
–	31-16	Зарезервировано	

## DATAOUTTGL – регистр переключения битов порта

Смещение: + 10h

Сброс: 0h

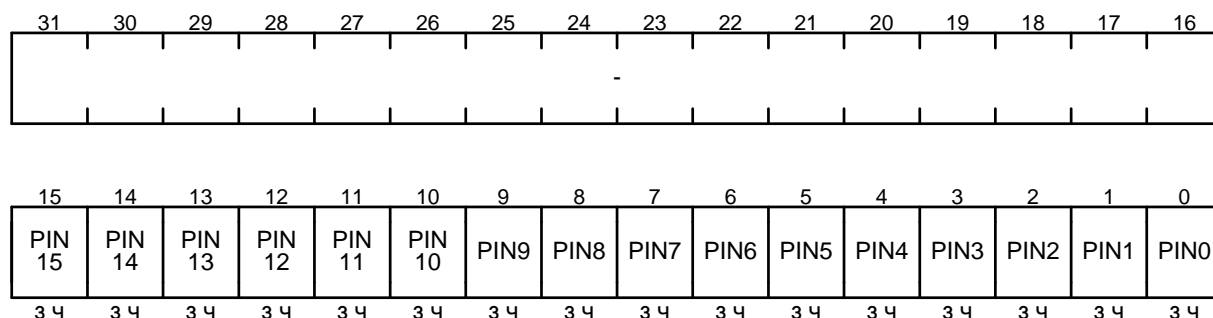


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Изменяет состояние соответствующего бита n в регистре DATAOUT на противоположное и, как следствие, состояние сигнала на выводе n порта
–	31-16	Зарезервировано	

## DENSET – регистр разрешения цифровой функции порта

Смещение: + 14h

Сброс: 0h

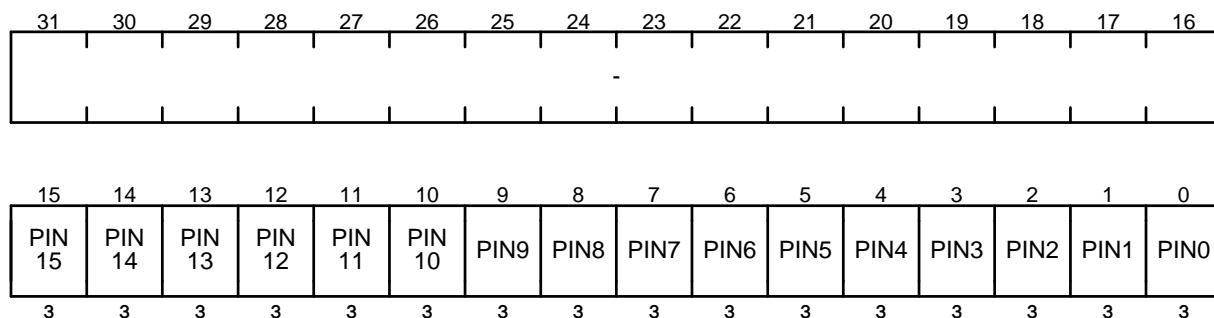


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Вывод n отключен и находится в третьем состоянии
			1	Вывод n подключен
		Запись нуля	Не выполняется	
		Запись единицы	Подключает вывод n к его цифровой схеме управления, т. е. разрешает функционирование вывода как входа/выхода порта	
–	31-16	Зарезервировано		

## DENCLR – регистр запрещения цифровой функции порта

Смещение: + 18h

Сброс: 0h

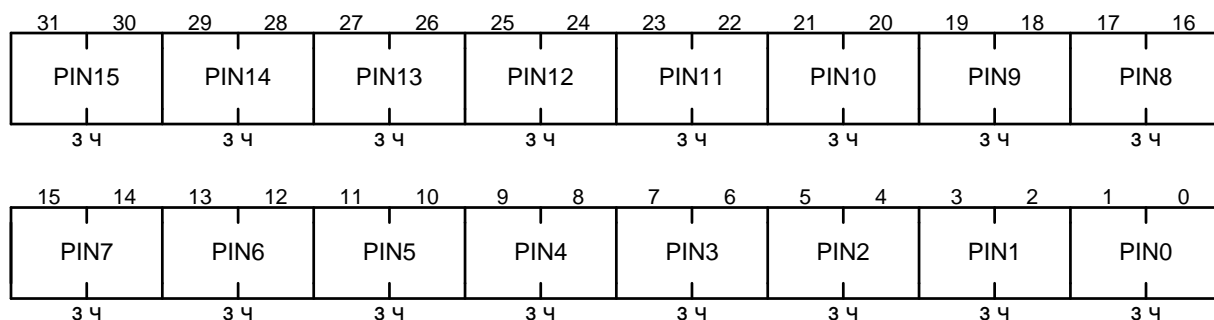


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Отключает вывод n от его цифровой схемы, т. е. запрещает функционирование вывода как входа/выхода порта
–	31-16	Зарезервировано	

### INMODE – регистр выбора режима входа порта

Смещение: + 1Ch

Сброс: 0h

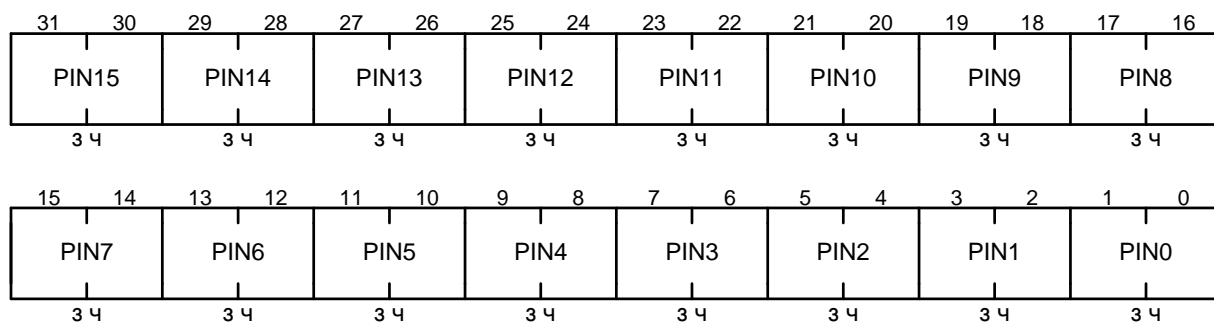


Поле	Биты	Описание
PINn	31-0	Поле выбора режима входа n
		00   Триггер Шмитта
		01, 10   Зарезервировано
		11   Входной буфер отключен

### PULLMODE – регистр выбора режима подтяжки порта

Смещение: + 20h

Сброс: 0h

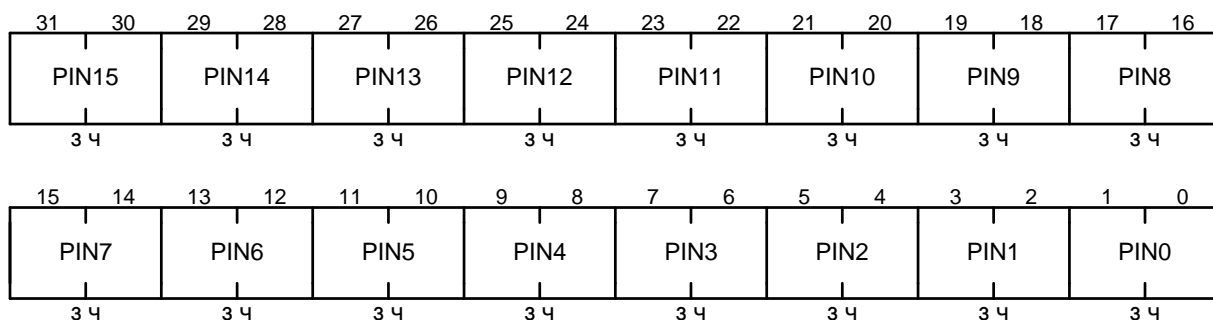


Поле	Биты	Описание
PINn	31-0	Поле выбора режима подтяжки вывода n
		00   Подтяжка отключена
		01   Подтяжка к уровню логической единицы (Pull-up)
		10, 11   Зарезервировано

### OUTMODE – регистр выбора режима выхода порта

Смещение: + 24h

Сброс: 0h

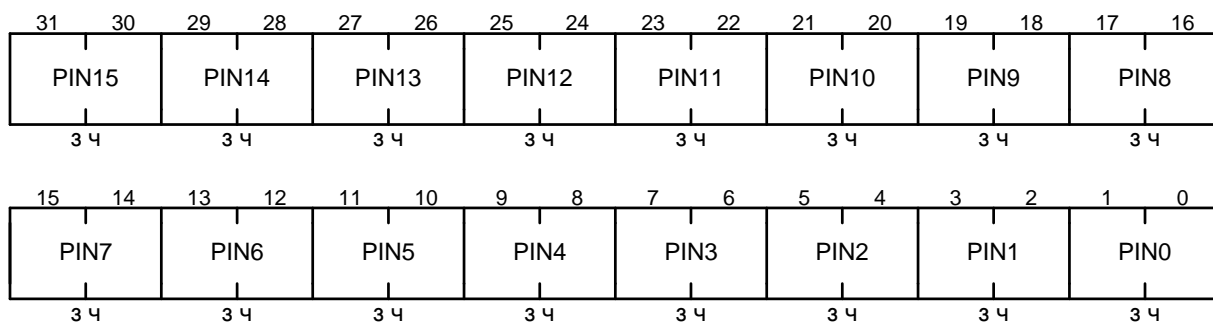


Поле	Биты	Описание	
PINn	31-0	Поле выбора режима выхода n	
		00	Двухтактный выход (Push-Pull)
		01	Выход с открытым стоком (Open Drain)
		10	Выход с открытым истоком (Open Source)
		11	Зарезервировано

### DRIVEMODE – регистр выбора параметров выхода порта

Смещение: + 28h

Сброс: 0h

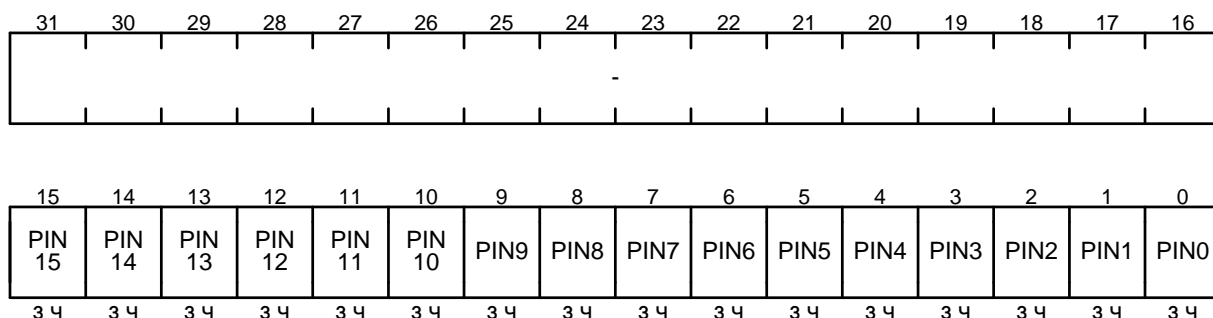


Поле	Биты	Описание	
PINn	31-0	Поле выбора параметров выхода порта	
		00	Повышенная нагрузочная способность
		01	Зарезервировано
		10	Пониженная нагрузочная способность
		11	Зарезервировано

## OUTENSET – регистр разрешения управления выходом порта

Смещение: + 2Ch

Сброс: 0h

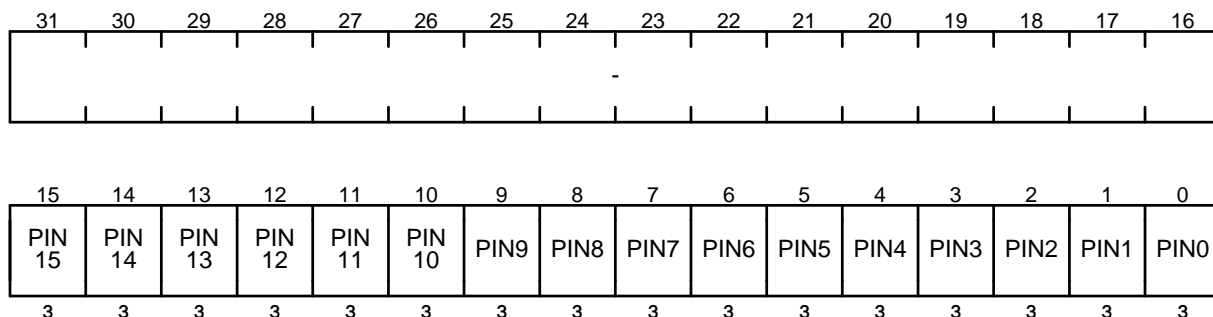


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Источник выходного сигнала (соответствующий бит регистра DATAOUT) отключен от вывода n
			1	Состояние вывода n может задаваться источником выходного сигнала (соответствующий бит регистра DATAOUT)
		Запись нуля	Не выполняется	
			Запись единицы	Разрешает управлять состоянием вывода с помощью соответствующего бита регистра DATAOUT
–	31-16	Зарезервировано		

## OUTENCLR – регистр запрещения управления выходом порта

Смещение: + 30h

Сброс: 0h

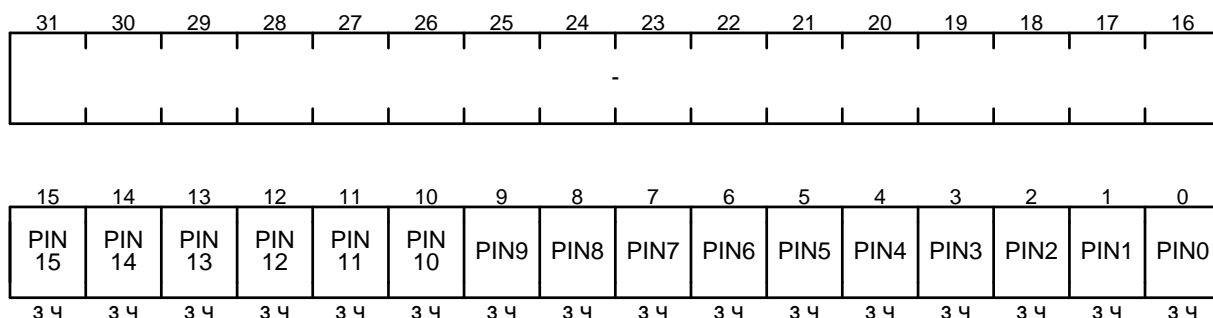


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n регистра OUTENSET и запрещает управлять состоянием вывода с помощью соответствующего бита регистра DATAOUT
–	31-16	Зарезервировано	

## ALTFUNCSET– регистр включения альтернативной функции порта

Смещение: + 34h

Сброс: 0h (для порта B)

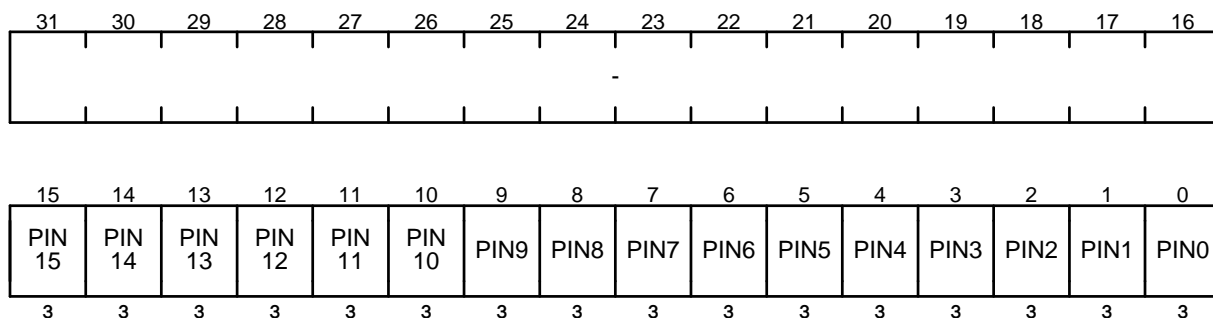


Поле	Биты	Описание	
PINn	15-0	Чтение	0 Вывод n в режиме входа/выхода общего назначения
			1 Вывод n в режиме альтернативной функции
		Запись нуля	Не выполняется
		Запись единицы	Включает режима альтернативной функции вывода n
–	31-16	Зарезервировано	

## ALTFUNCCLR – регистр выключения альтернативной функции порта

Смещение: + 38h

Сброс: 0h

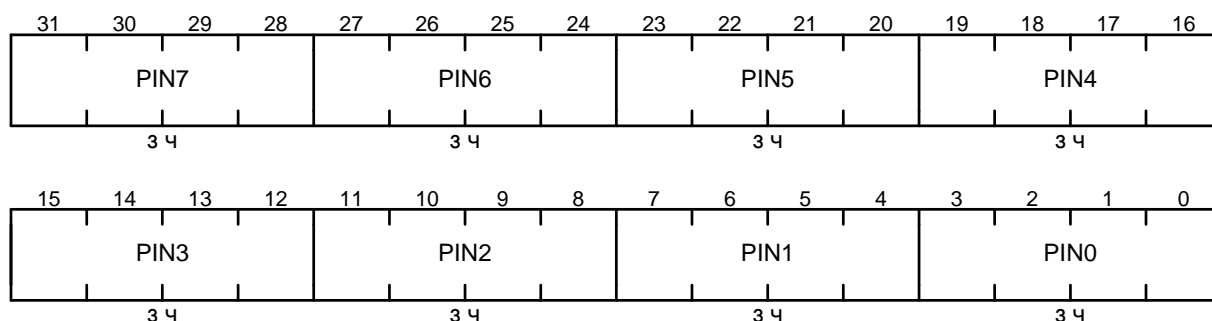


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	0 Не выполняется
		Запись единицы	1 Включает режима входа/выхода общего назначения
–	31-16	Зарезервировано	

### ALTFUNCNUM0 – регистр выбора альтернативной функции младшего байта порта

Смещение: + 3Ch

Сброс: 0h

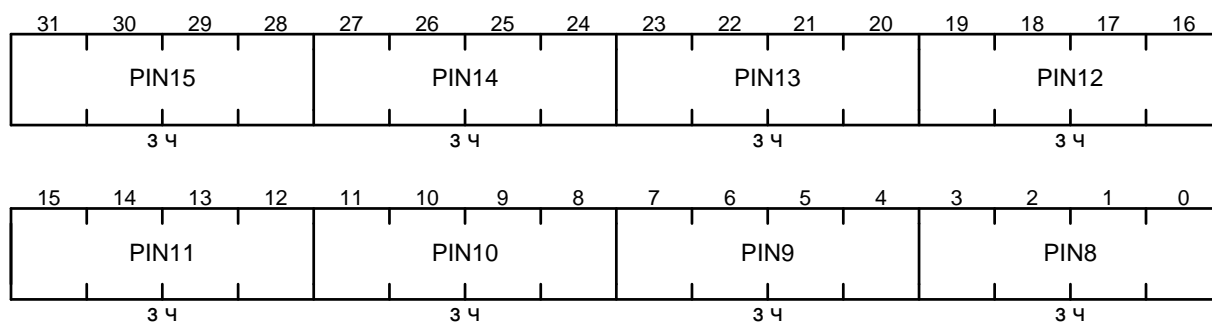


Поле	Биты	Описание	
PINn	31-0	Поле выбора альтернативной функции пина	
		0h	Альтернативная функция пина не выбрана
		1h	Альтернативная функция 1
		2h	Альтернативная функция 2
		3h	Альтернативная функция 3
		4h	Альтернативная функция 4

### ALTFUNCNUM1 – регистр выбора альтернативной функции старшего байта порта

Смещение: + 40h

Сброс: 0h



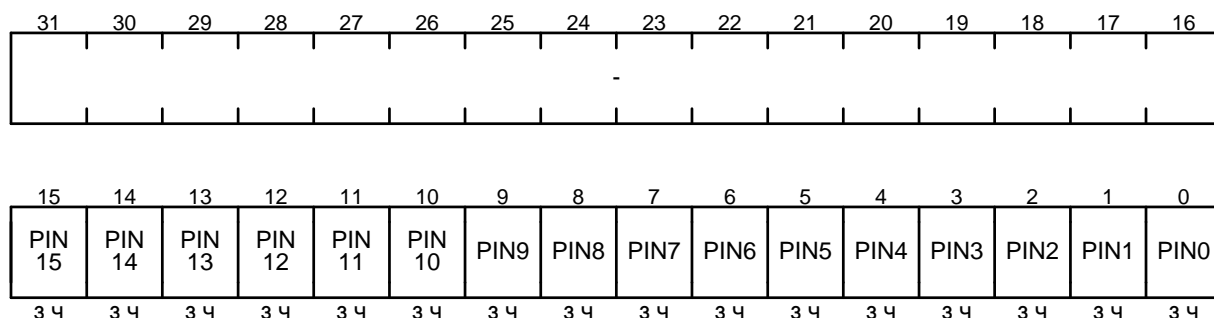
Поле	Биты	Описание	
PINn	31-0	Поле выбора альтернативной функции пина	
		0h	Альтернативная функция пина не выбрана
		1h	Альтернативная функция 1
		2h	Альтернативная функция 2
		3h	Альтернативная функция 3
		4h	Альтернативная функция 4



## SYNCSET – регистр включения дополнительной синхронизации входов портов

Смещение: + 44h

Сброс: 0h

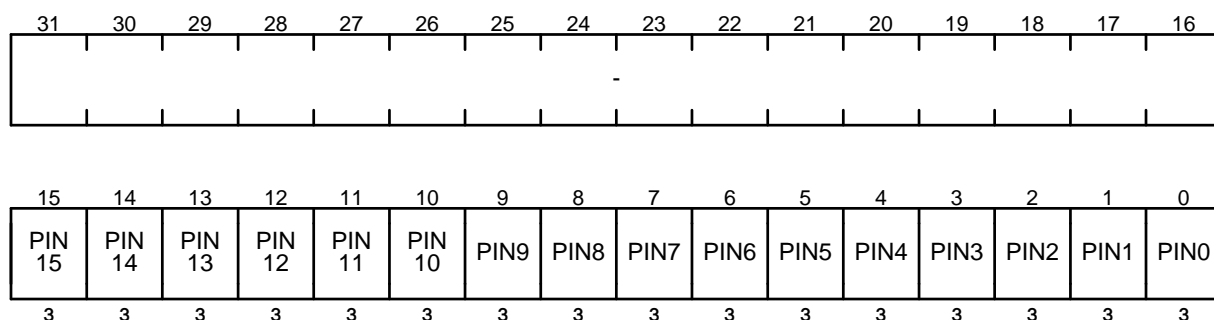


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Сигнал с вывода n передается в регистр DATA после двухтактной синхронизации (базовая)
			1	Сигнал с вывода n передается в регистр DATA после четырехтактной синхронизации (базовая и дополнительная)
		Запись нуля	Не выполняется	
		Запись единицы	Подключает дополнительную схему для получения четырехтактной синхронизации	
–	31-16	Зарезервировано		

## SYNCCLR – регистр выключения дополнительной синхронизации входов портов

Смещение: + 48h

Сброс: 0h

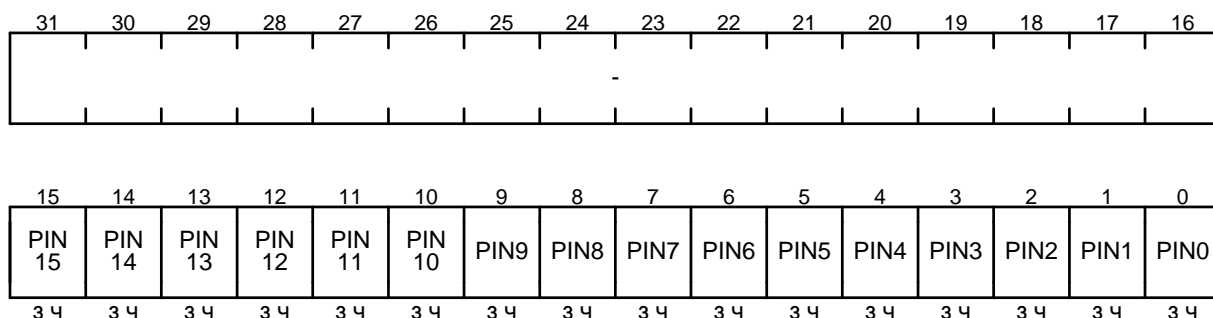


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Отключает дополнительную схему синхронизации вывода n
–	31-16	Зарезервировано	

## QUALSET – регистр включения фильтров портов

Смещение: + 4Ch

Сброс: 0h

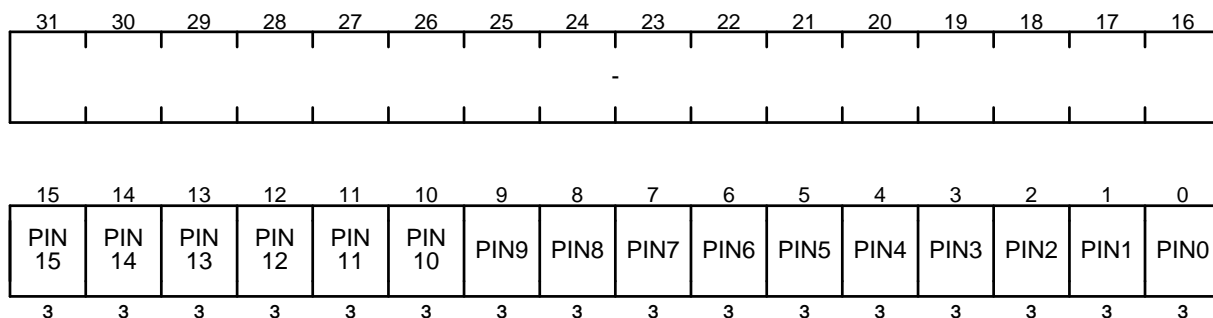


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Входной фильтр вывода n отключен
			1	Входной фильтр вывода n включен
		Запись нуля	Не выполняется	
			Запись единицы	Подключает входной фильтр вывода n
–	31-16	Зарезервировано		

## QUALCLR – регистр отключения фильтров портов

Смещение: + 50h

Сброс: 0h

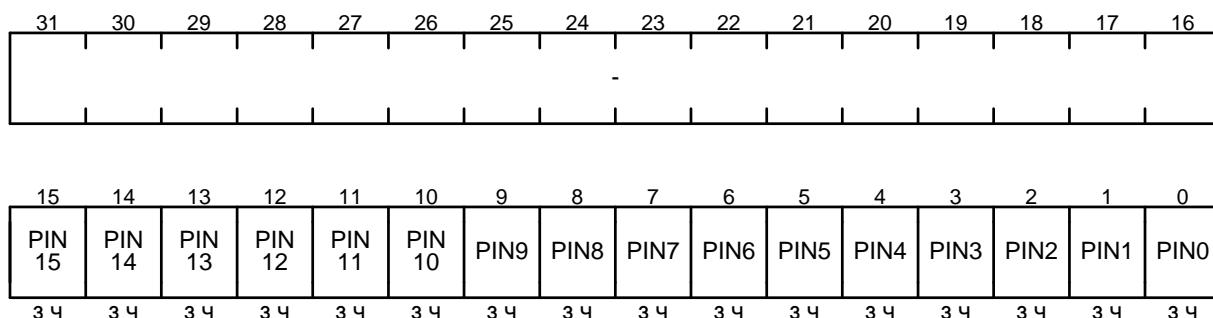


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Отключает входной фильтра вывода n
–	31-16	Зарезервировано	

## QUALMODESET – регистр режима фильтра порта

Смещение: + 54h

Сброс: 0h

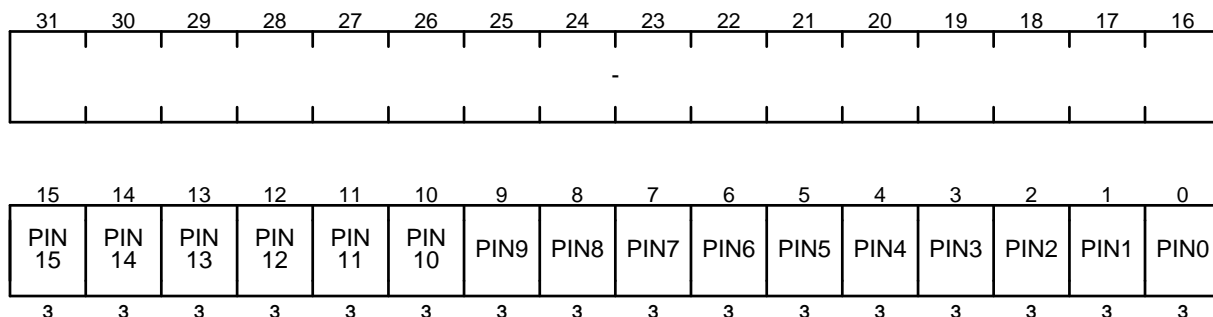


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Включен режим измерения уровня сигнала на выводе n по трем отсчетам
			1	Включен режим измерения уровня сигнала на выводе n по шести отсчетам
		Запись нуля	Не выполняется	
		Запись единицы	Включает режим измерения уровня сигнала на выводе n по шести отсчетам	
–	31-16	Зарезервировано		

## QUALMODECLR – регистр сброса режима фильтра порта

Смещение: + 58h

Сброс: 0h

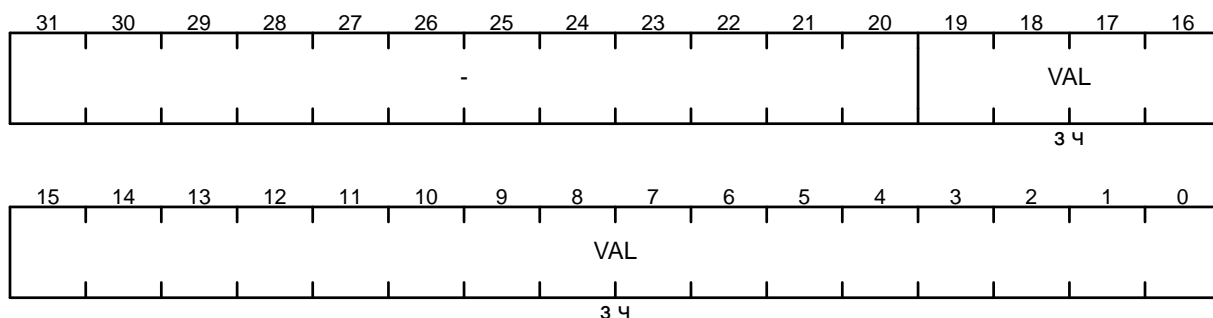


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре QUALMODESET и включает режим измерения уровня сигнала на выводе n по трем отсчетам
–	31-16	Зарезервировано	

## QUALSAMPLE – регистр настройки фильтра порта

Смещение: + 5Ch

Сброс: 0h

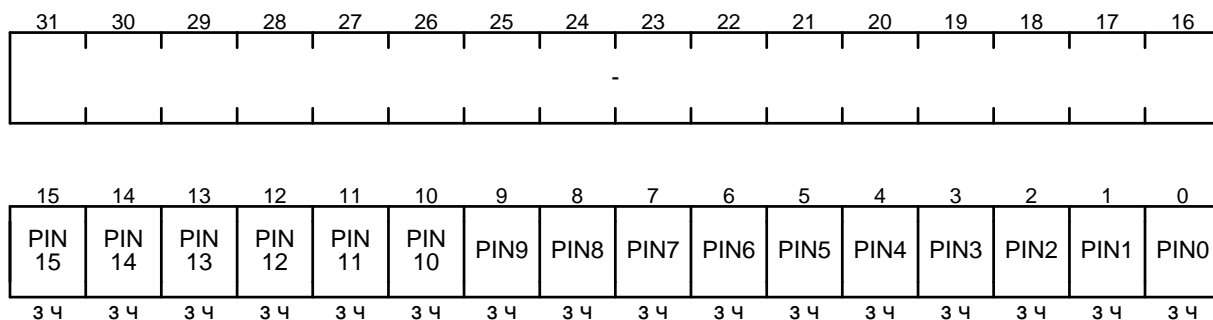


Поле	Биты	Описание
PERIOD	19-0	Временной интервал (в тактах FCLK) между отсчетами при измерениях уровней сигналов на выводах порта. Заданное значение является единым для всех выводов порта
–	31-20	Зарезервировано

## INTENSET – регистр разрешения прерываний порта

Смещение: + 60h

Сброс: 0h

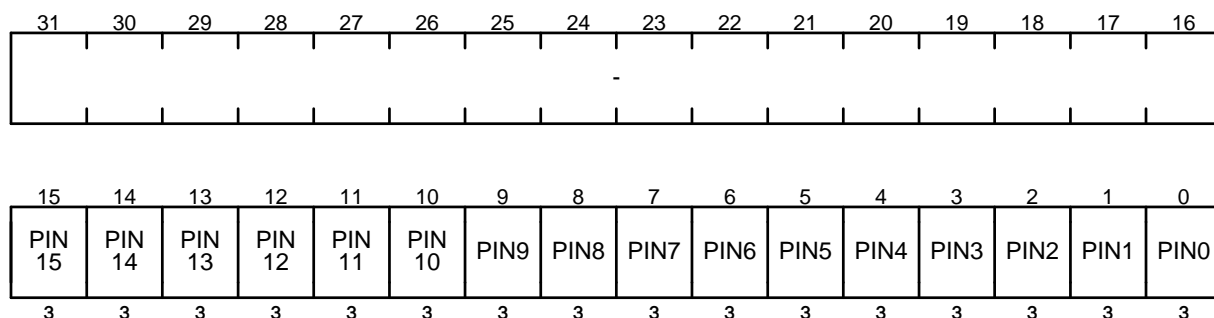


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания вывода n запрещены
			1	Прерывания вывода n разрешены
		Запись нуля	Не выполняется	
			Запись единицы	Разрешает прерывания вывода n
–	31-16	Зарезервировано		

## INTENCLR – регистр сброса разрешения прерываний порта

Смещение: + 64h

Сброс: 0h

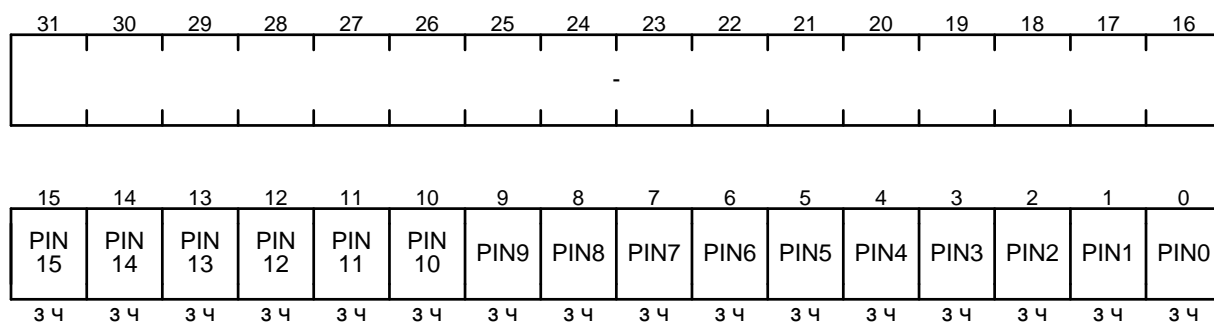


Поле	Биты	Описание	
PINn	15-0	Чтение	Всегда читаются нули
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре INTENSET и запрещает прерывания вывода n
–	31-16	Зарезервировано	

## INTTYPESET – регистр типа прерываний порта

Смещение: + 68h

Сброс: 0h

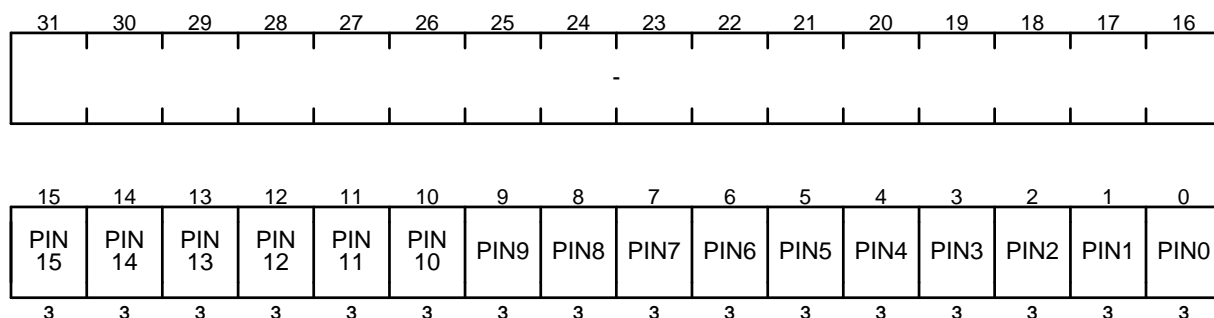


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по уровню сигнала на выводе n
			1	Прерывания по фронту сигнала на выводе n
		Запись нуля	Не выполняется	
		Запись единицы	Выбирает прерывания по фронту сигнала на выводе n	

## INTTYPECLR – регистр сброса типа прерываний порта

Смещение: + 6Ch

Сброс: 0h

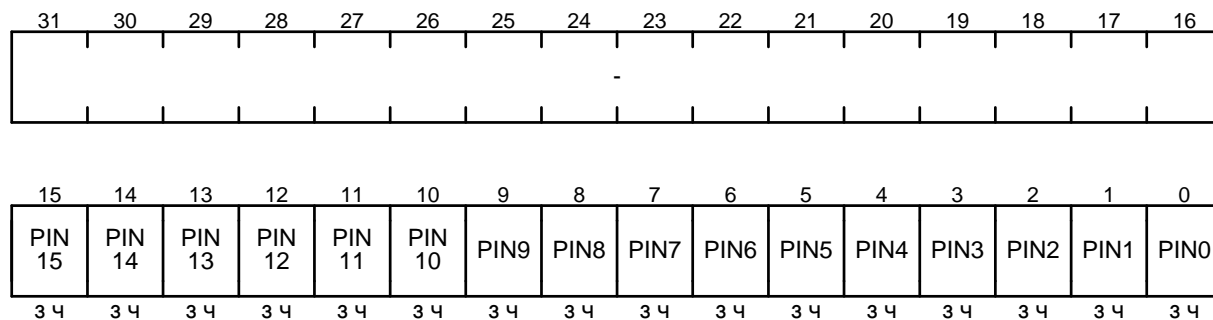


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Нет реакции
		Запись единицы	Сбрасывает соответствующий бит n в регистре INTTYPESET и выбирает прерывания по уровню сигнала на выводе n
–	31-16	Зарезервировано	

## INTPOLSET – регистр полярности события прерывания порта

Смещение: + 70h

Сброс: 0h

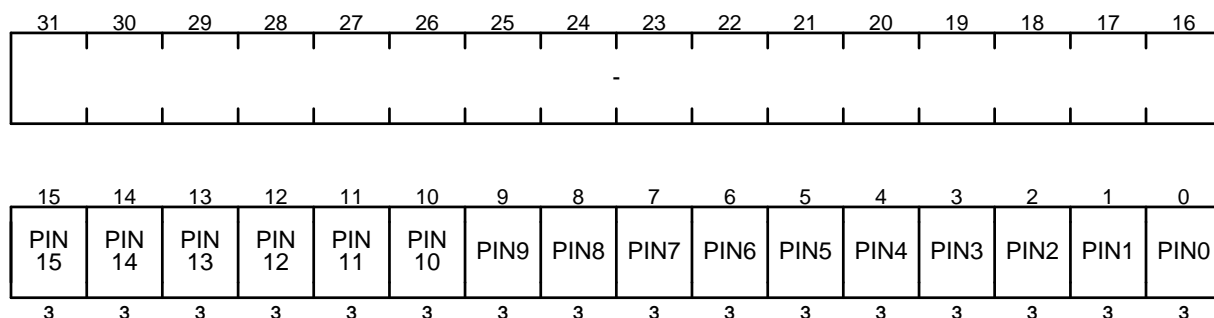


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по низкому уровню или отрицательному фронту сигнала на выводе n
			1	Прерывания по высокому уровню или положительному фронту сигнала на выводе n
		Запись нуля	Не выполняется	
		Запись единицы	Выбирает прерывания по высокому уровню или положительному фронту	
–	31-16	Зарезервировано		

## INTPOLCLR – регистр сброса полярности события прерывания порта

Смещение: + 74h

Сброс: 0h

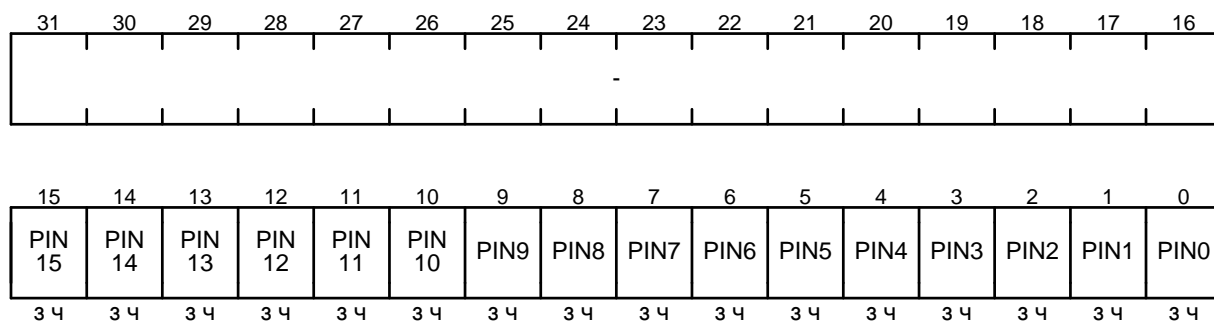


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре INTPOLCLR и выбирает прерывания по низкому уровню или отрицательному фронту
–	31-16	Зарезервировано	

## INTEDGESET – регистр включения прерывания по любому перепаду

Смещение: + 78h

Сброс: 0h

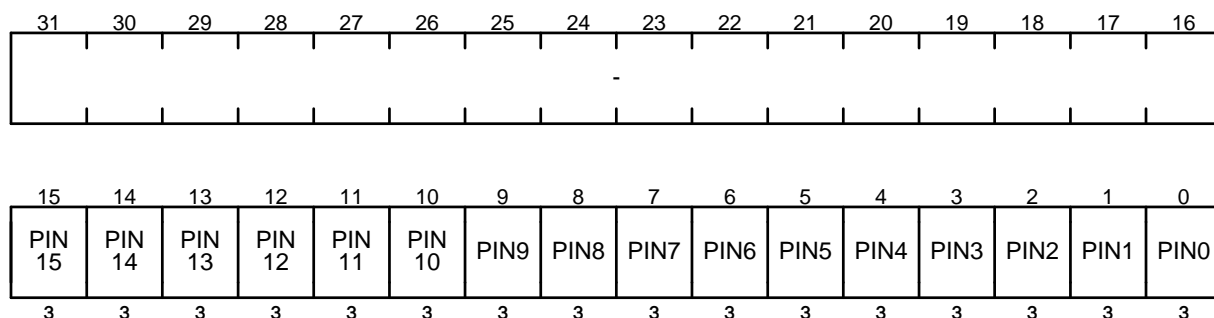


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по любому фронту сигнала на выводе n запрещены
			1	Прерывания по любому фронту сигнала на выводе n разрешены
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает прерывания по любому фронту сигнала на выводе n	
–	31-16	Зарезервировано		

## INTEDGECLR – регистр отключения прерывания по любому перепаду

Смещение: + 7Ch

Сброс: 0h

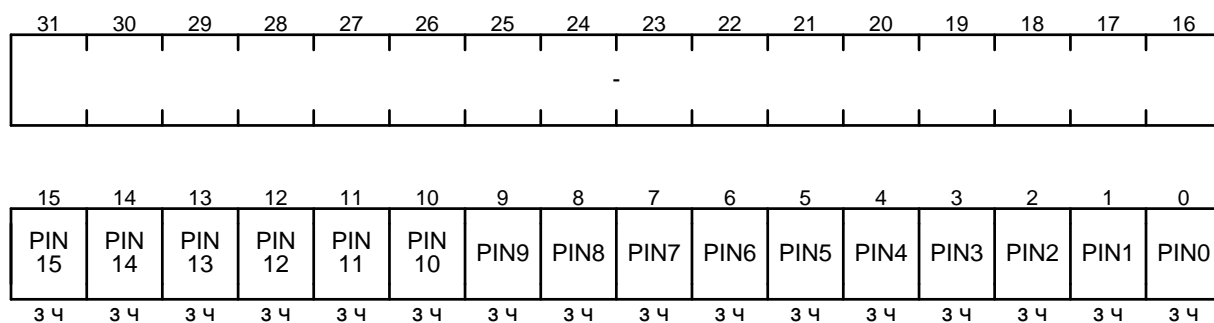


Поле	Биты	Описание	
PINn	15-0	Чтение	Всегда читаются нули
		Запись нуля	Не выполняется
		Запись единиц	Сбрасывает соответствующий бит n в регистре INTEDGESET и запрещает прерывания по любому фронту сигнала на выводе n
–	31-16	Зарезервировано	

## INTSTATUS – регистр состояния и сброса прерываний порта

Смещение: + 80h

Сброс: 0h



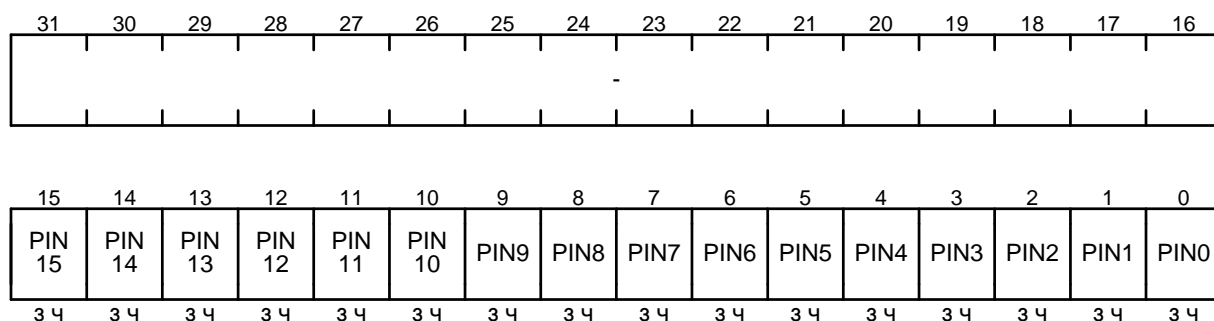
Поле	Биты	Описание		
PINn	15-0	Чтение	0	Нет прерывания
			1	Флаг запроса на прерывание. Бит не сбрасывается аппаратно.
		Запись нуля	Не выполняется	
		Запись единицы	Обнуляет бит PINn и таким образом сбрасывает флаг прерывания, если он был установлен	
–	31-16	Зарезервировано		



## DMAREQSET – регистр включения генерации запросов DMA по прерыванию порта

Смещение: + 84h

Сброс: 0h

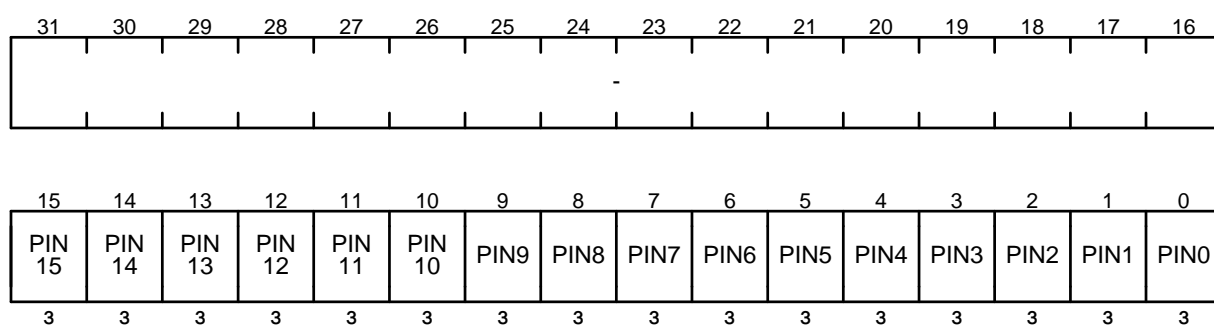


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос DMA не генерируется
			1	Запрос DMA генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
			Запись единицы	Разрешает генерирование запросов DMA по прерыванию вывода n (в том числе немаскированному)
–	31-16	Зарезервировано		

## DMAREQCLR – регистр отключения генерации запросов DMA по прерыванию порта

Смещение: + 88h

Сброс: 0h

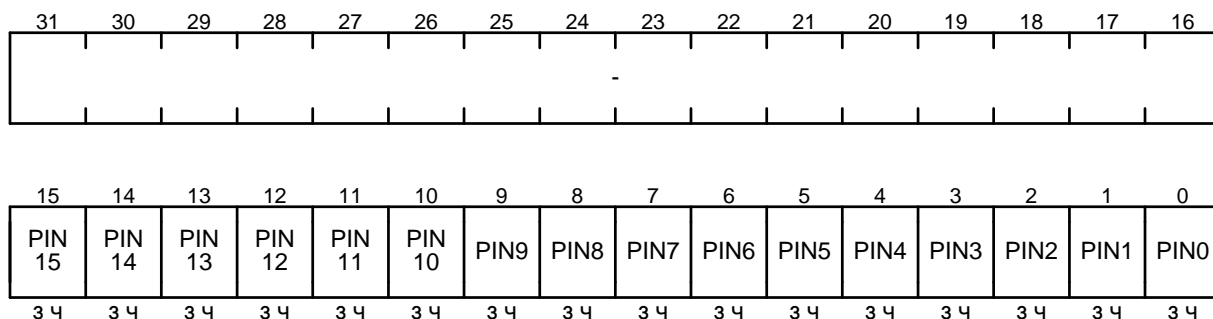


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре DMAREQSET и запрещает генерирование запросов DMA по прерыванию вывода n
–	31-16	Зарезервировано	

## ADCSOCSET – регистр включения генерации запросов начала преобразования АЦП по прерыванию порта

Смещение: + 8Ch

Сброс: 0h

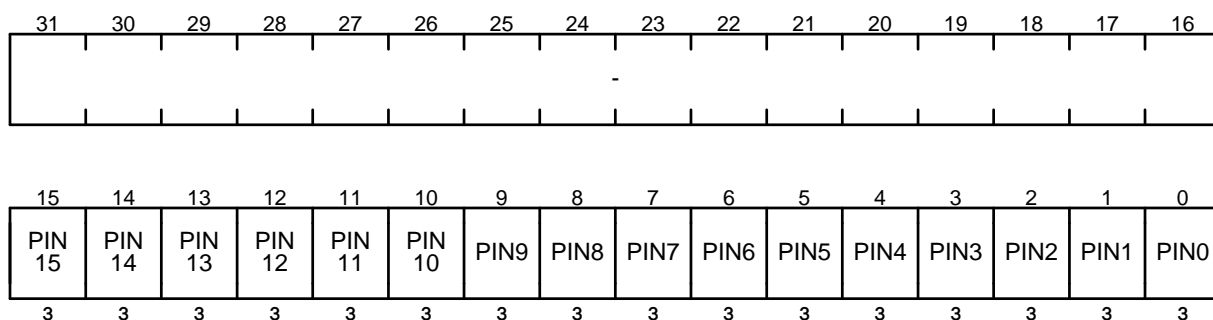


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос начала преобразования АЦП не генерируется
			1	Запрос начала преобразования АЦП генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов начала преобразования АЦП по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

## ADCSOCCLR – регистр отключения генерации запросов начала преобразования АЦП по прерыванию порта

Смещение: + 90h

Сброс: 0h

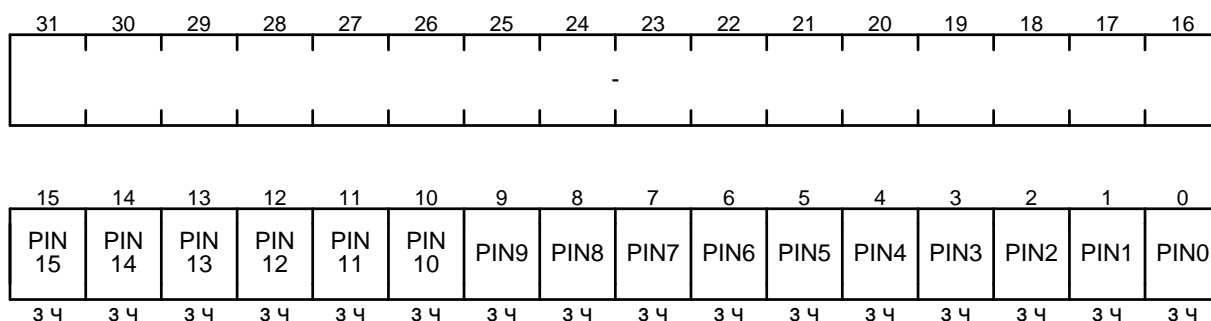


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единиц	Сбрасывает соответствующий бит n в регистре ADCSOCSET и запрещает генерирование запросов АЦП по прерыванию вывода n
–	31-16	Зарезервировано	

## RXEVSET – регистр включения генерации запросов RXEV к ядру по прерыванию порта

Смещение: +94h

Сброс: 0h

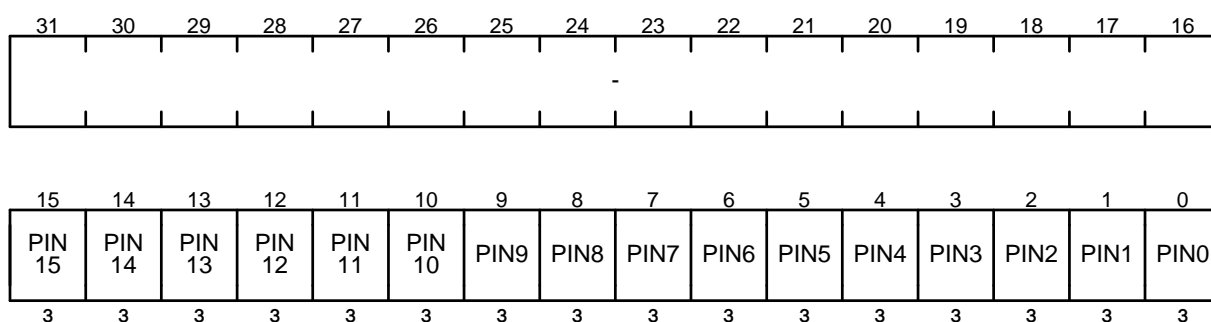


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос RXEV к ядру не генерируется
			1	Запрос RXEV к ядру генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов RXEV к ядру по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

## RXEVCLR – регистр отключения генерации запросов RXEV к ядру по прерыванию порта

Смещение: + 98h

Сброс: 0h

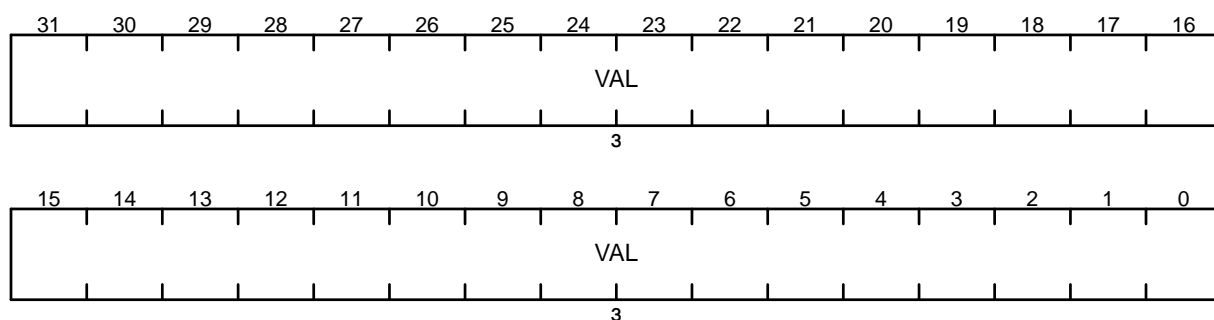


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает соответствующий бит n в регистре RXEVSET и запрещает генерирование запросов RXEV к ядру по прерыванию вывода n
–	31-16	Зарезервировано	

## LOCKKEY – регистр ключа блокировки

Смещение: + 9Ch

Сброс: 0h

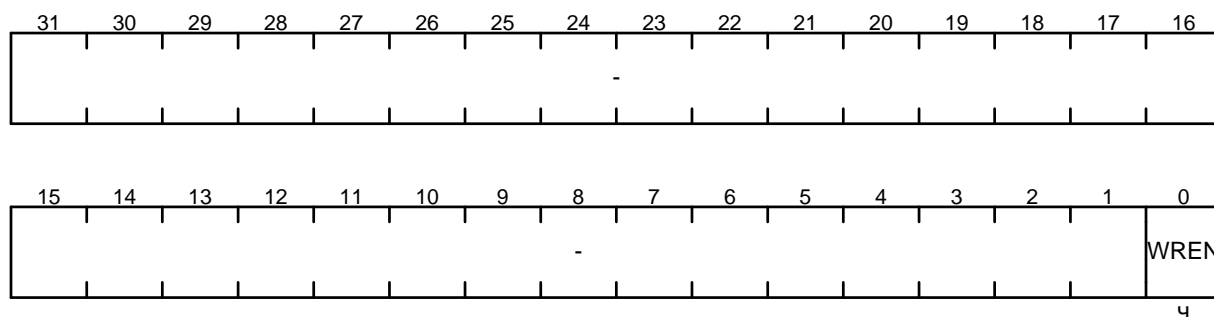


Поле	Биты	Описание	
VAL	31-0	Значение ключа – ADEADBEEh	
		Чтение	Не выполняется
		Запись любого значения отличного от ключа	Блокирует доступ к регистрам LOCKSET и LOCKCLR
		Запись значения ключа	Разрешает доступ к регистрам LOCKSET и LOCKCLR

## LOCKSTAT– регистр статуса блокировки

Смещение: + 9Ch

Сброс: 0h

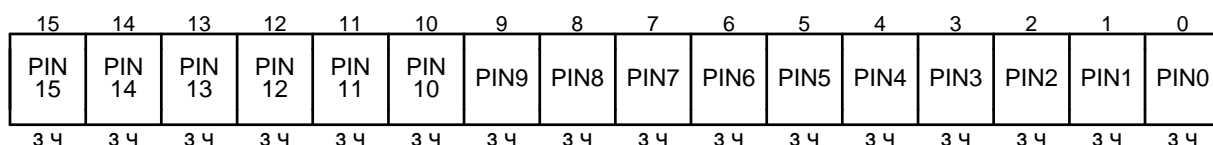
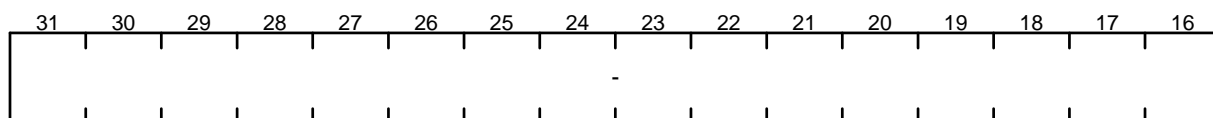


Поле	Биты	Описание	
WREN	0	Индикатор доступа регистров LOCKSET и LOCKCLR для записи	
		0	Запрещена
		1	Разрешена
		Бит не доступен для записи	
–	31-1	Зарезервировано	

## LOCKSET – регистр включения блокировки изменения конфигурации вывода

Смещение: + A0h

Сброс: 0h

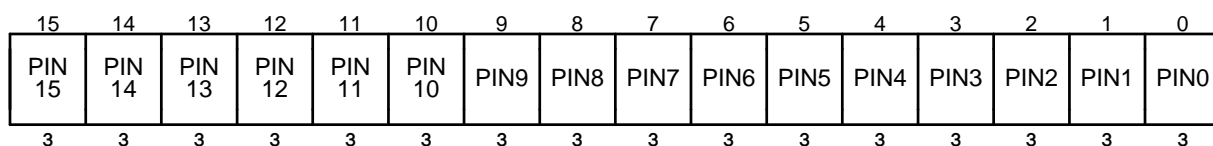
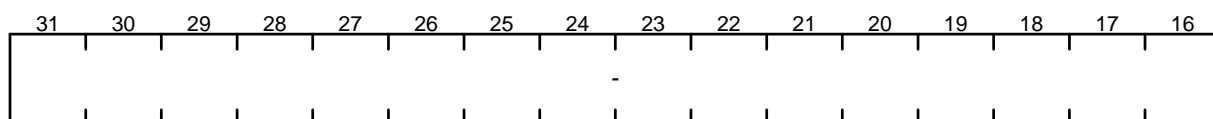


Поле	Биты	Описание		
PINn	15-0	Чтение	0	Блокировка отключена
			1	Блокировка включена
		Запись нуля	Не выполняется	
		Запись единицы (возможна после установки ключа в регистре LOCKKEY)	Блокирует изменение конфигурации вывода n. Соответствующие биты n регистров становятся недоступными для записи. Исключения: - регистр флагов прерываний INTSTATUS; - регистр временного интервала измерений QUALSAMPLE	
-	31-16	Зарезервировано		

## LOCKCLR – регистр отключения блокировки изменения конфигурации вывода

Смещение: + A4h

Сброс: 0h

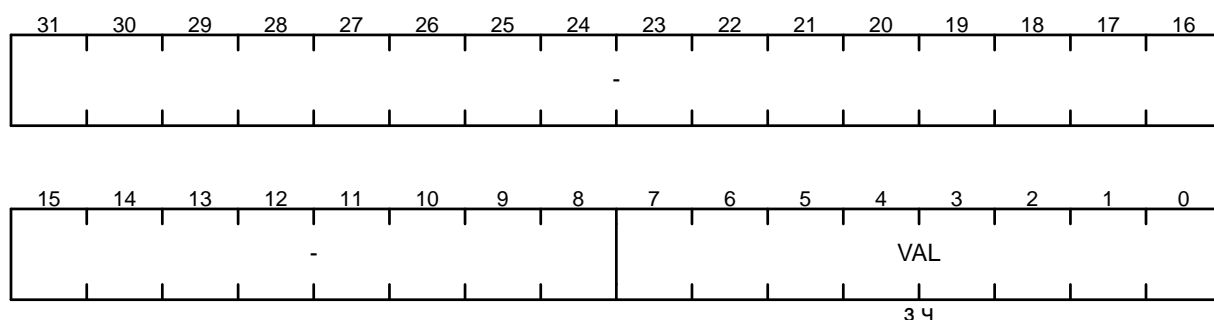


Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы (возможна после установки ключа в регистре LOCKKEY)	Сбрасывает соответствующий бит n в регистре LOCKSET и разрешает изменения конфигурации вывода n
-	31-16	Зарезервировано	

### MASKLB – регистр массива масок младшего байта порта

Адрес:  $\text{GPIO}_p + \text{MASKLB} + (4 \cdot i)_h$ ,  $i = 0h, \dots, FFh$

Сброс: 000000xxh

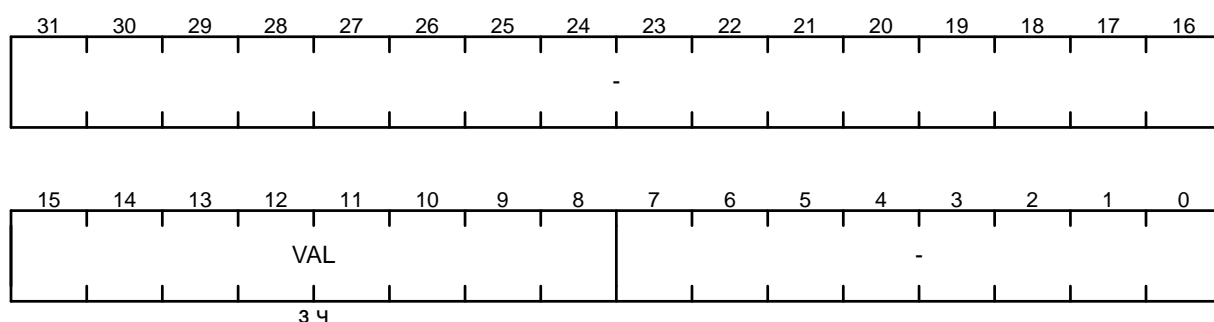


Поле	Биты	Описание
VAL	7-0	Доступ по маске для младших восьми бит порта
–	31-8	Зарезервировано

### MASKHB – регистр массива масок старшего байта порта

Адрес:  $\text{GPIO}_p + \text{MASKHB} + (4 \cdot i)_h$ ,  $i = 0h, \dots, FFh$

Сброс: 0000xx00h



Поле	Биты	Описание
VAL	15-8	Доступ по маске для старших восьми бит порта
–	31-16, 7-0	Зарезервировано

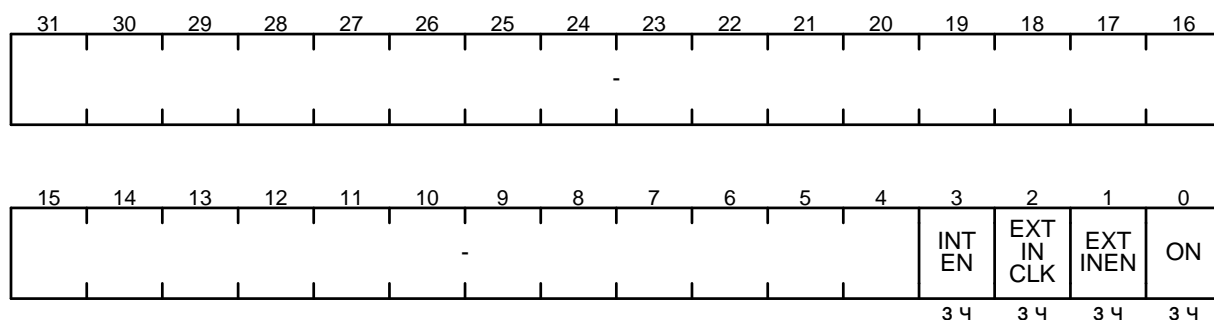
## А.9 Регистры таймера TMR

<b>Базовый адрес:</b>	4009_1000h	Регистры таймера TMR0;
	4009_2000h	Регистры таймера TMR1;
	4009_3000h	Регистры таймера TMR2;
	4009_4000h	Регистры таймера TMR3

### CTRL – регистр управления блока таймера

Смещение: + 00h

Сброс: 0h

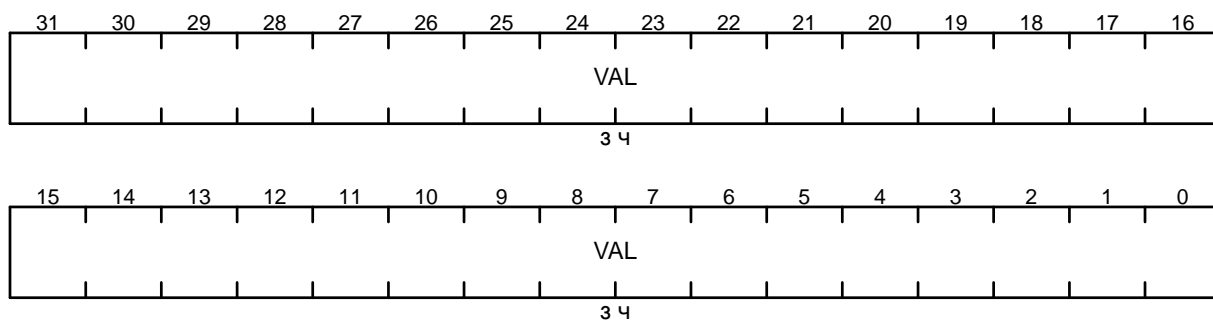


Поле	Биты	Описание
INTEN	3	Бит разрешения прерывания таймера
		0   Запрещено
		1   Разрешено
EXTINCLK	2	Бит включения внешнего входа синхронизации как тактового
		0   Нет действий
		1   Сигнал на входе TMR <sub>x</sub> _IN (x – номер таймера от 0 до 2) является тактовым
EXTINEN	1	Бит разрешения работы таймера, если сигнал на внешнем входе равен единице
		0   Запрещено
		1   Таймер декрементируется с частотой PCLK
ON	0	Бит включения таймера
		0   Выключен
		1   Таймер декрементируется с частотой PCLK
–	31-4	Зарезервировано

### VALUE – регистр текущего значения таймера

Смещение: + 04h

Сброс: 0h

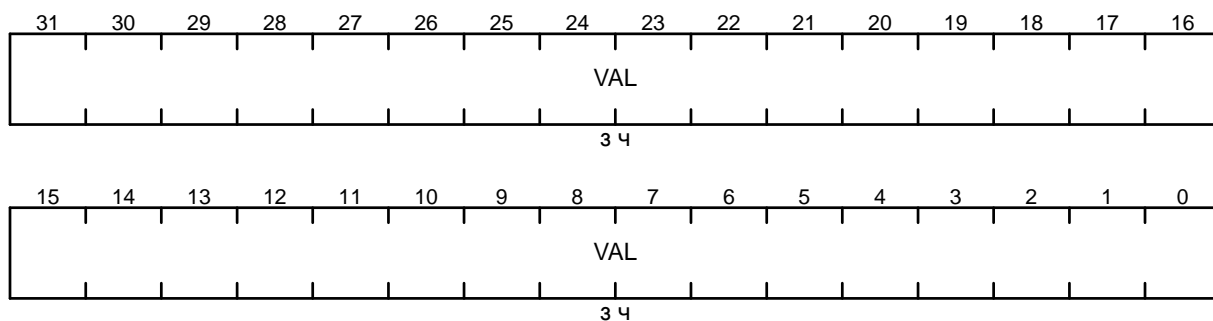


Поле	Биты	Описание
VAL	31-0	Текущее значение таймера

### LOAD – регистр начального значения счетчика таймера

Смещение: + 08h

Сброс: 0h



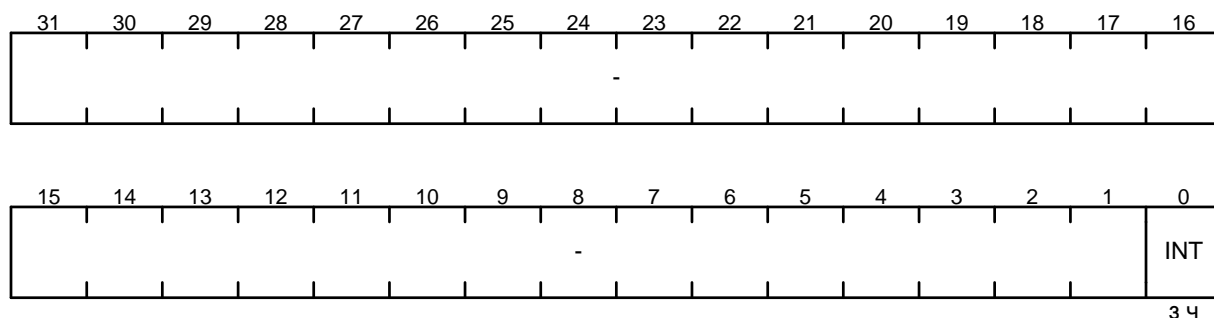
Поле	Биты	Описание
VAL	31-0	Значение перезагрузки таймера



## INTSTATUS – регистр прерывания таймера

Смещение: + 0Ch

Сброс: 0h

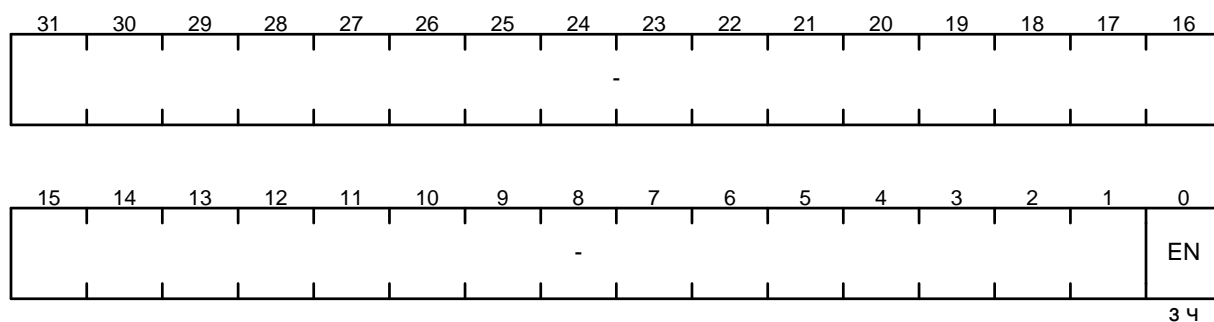


Поле	Биты	Описание
INT	0	Флаг прерывания таймера
		0   Нет прерывания
		1   Запрос на прерывание
		Флаг сбрасывается записью единицы
–	31-1	Зарезервировано

## DMAREQ – регистр управления запросом DMA

Смещение: + 10h

Сброс: 0h

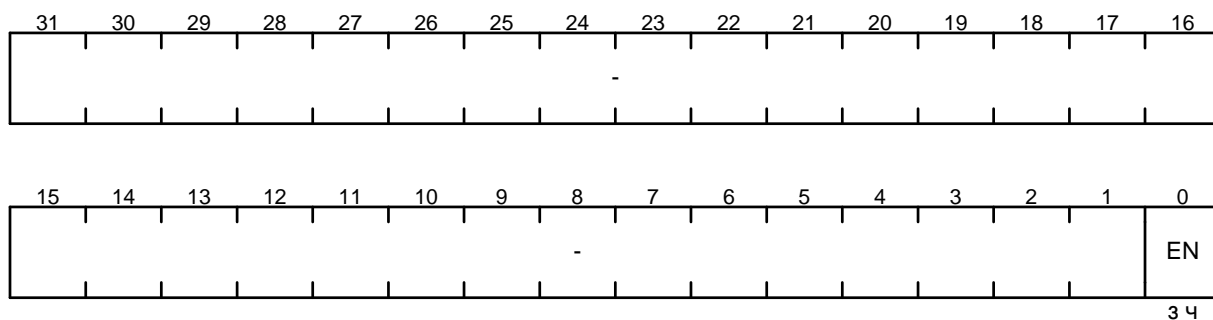


Поле	Биты	Описание
EN	0	Разрешение генерации запроса контроллера DMA по опустошению таймера
		0   Нет запроса
		1   Запрос разрешен
–	31-1	Зарезервировано

## ADCSOC – регистр управления запуском АЦП

Смещение: + 14h

Сброс: 0h



Поле	Биты	Описание
EN	0	Разрешение генерации сигнала запуска АЦП по опустошению таймера
		0   Нет сигнала
		1   Генерация разрешена
–	31-1	Зарезервировано

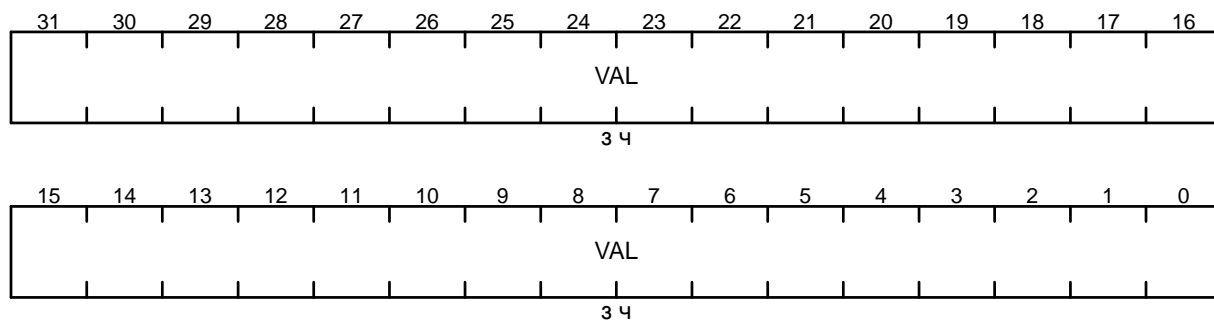
## A.10 Регистры расширенного таймера ETMR

<b>Базовый адрес:</b>	4009_5000h	Регистры таймера ETMR0
	4009_6000h	Регистры таймера ETMR1
	4009_7000h	Регистры таймера ETMR2
	4009_8000h	Регистры таймера ETMR3

### LOAD – регистр начального значения счетчика таймера

Смещение: + 00h

Сброс: 0h

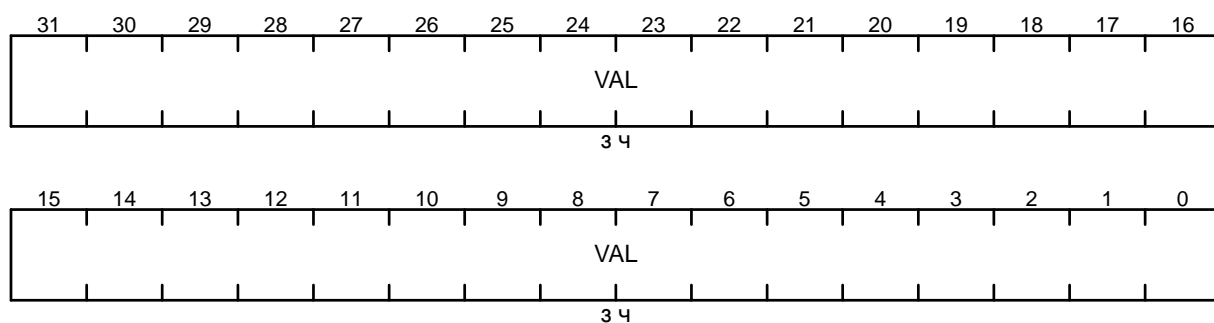


Поле	Биты	Описание
VAL	31-0	Значение перезагрузки таймера

### VALUE – регистр текущего значения таймера

Смещение: + 04h

Сброс: FFFF\_FFFFh

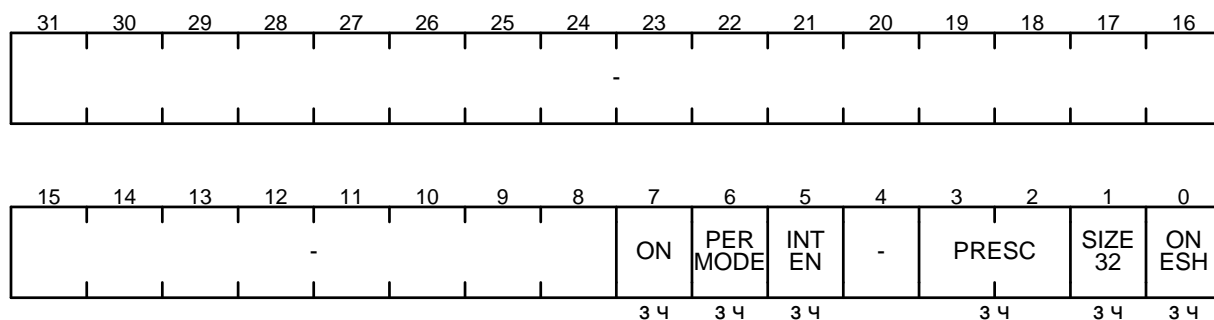


Поле	Биты	Описание
VAL	31-0	Текущее значение таймера

## CTRL – регистр управления блока таймера

Смещение: + 08h

Сброс: 0000\_0020h

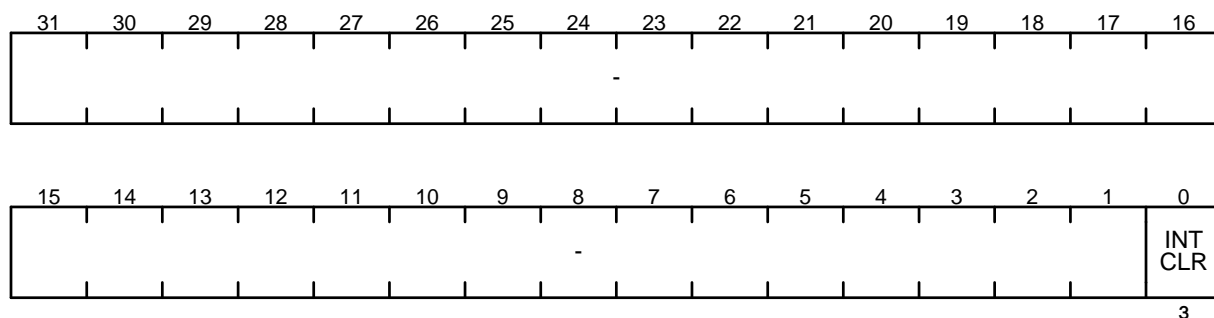


Поле	Биты	Описание
ON	7	Бит включения таймера
		0   Выключен
		1   Таймер декрементируется
PERMODE	6	Бит выбора режима счёта
		0   Свободный счёт
		1   Периодичный счёт
INTEN	5	Бит разрешения прерывания таймера
		0   Запрещено
		1   Разрешено
PRESC	3-2	Выбор величины деления тактового сигнала PCLK
		00   PCLK/1
		01   PCLK/16
		10   PCLK/256
		11   Зарезервировано
SIZE32	1	Бит выбора разрядности таймера
		0   16-битный режим
		1   32-битный режим
ONESH	2	Бит включения однократного режима счёта
		0   Выключен
		1   Включен
–	31-8, 4	Зарезервировано

## INTCLR – регистр сброса флагов таймера

Смещение: + 0Ch

Сброс: 0h

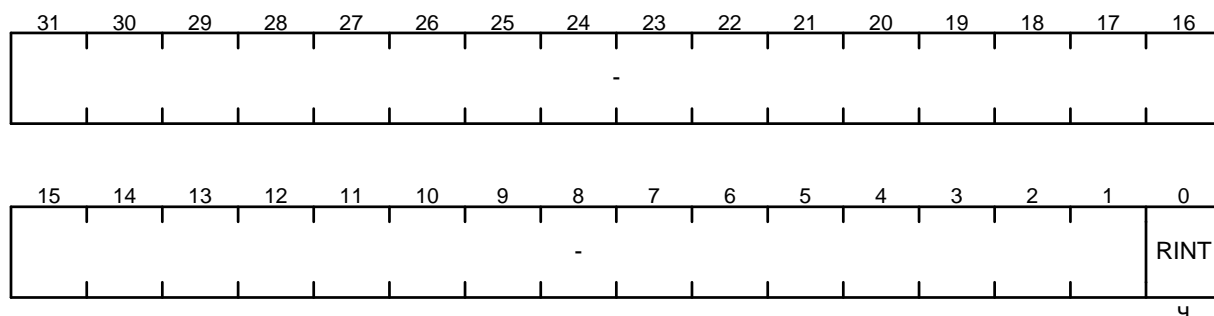


Поле	Биты	Описание
INTCLR	0	Бит сброса флагов регистров RIS, MIS
		0   Нет реакции
		1   Сброс флагов
–	31-1	Зарезервировано

## RIS – регистр немаскированных флагов прерывания таймера

Смещение: + 10h

Сброс: 0h

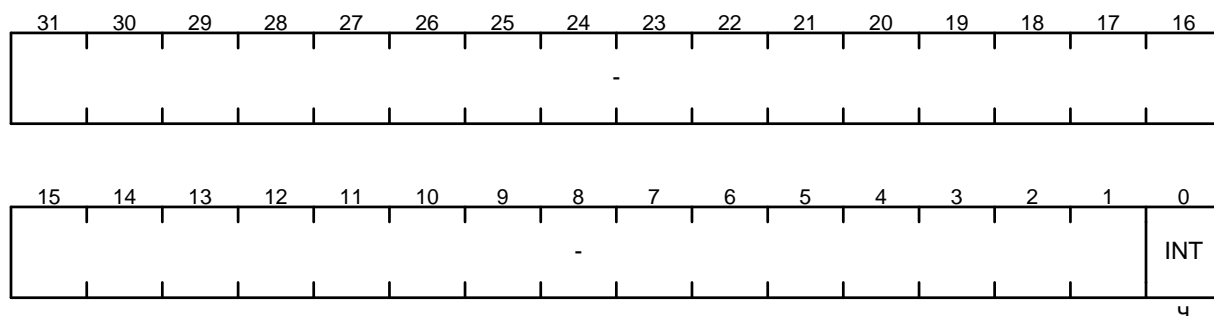


Поле	Биты	Описание
RINT	0	Флаг немаскированного прерывания таймера
		0   Нет события
		1   Счётчик досчитал до нуля
–	31-1	Зарезервировано

### MIS – регистр немаскированных флагов прерывания таймера

Смещение: + 14h

Сброс: 0h

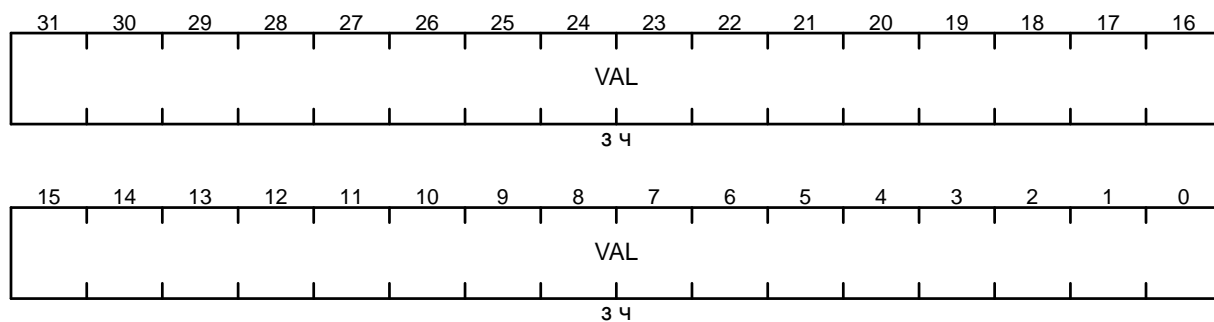


Поле	Биты	Описание
INT	0	Флаг маскированного прерывания таймера
		0   Нет события
		1   Счётчик досчитал до нуля и было вызвано прерывание
–	31-1	Зарезервировано

### BGLOAD – регистр теневого значения загрузки счетчика таймера

Смещение: + 18h

Сброс: 0h

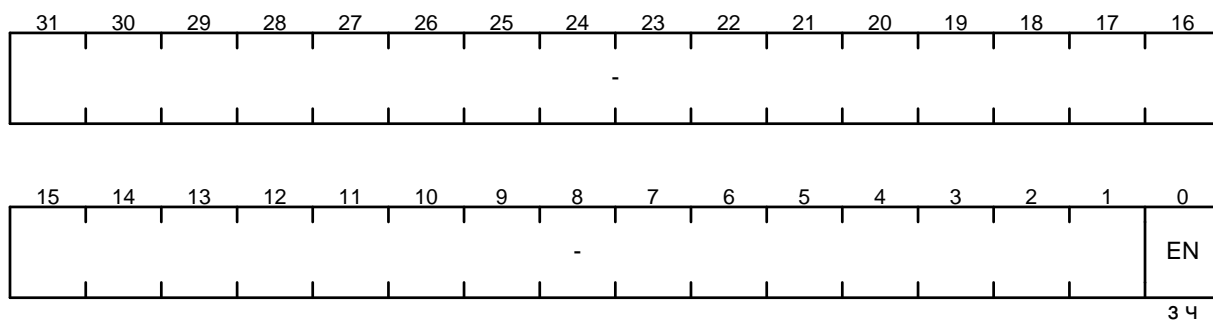


Поле	Биты	Описание
VAL	31-0	Теневое значение перезагрузки таймера. Записанная величина обновит регистр LOAD по опустошению таймера.

## DMAREQ – регистр управления запросом DMA

Смещение: + 1Ch

Сброс: 0h



Поле	Биты	Описание
EN	0	Разрешение генерации запроса контроллера DMA по опустошению таймера
		0   Нет запроса
		1   Запрос разрешен
–	31-1	Зарезервировано

## A.11 Регистры блока программируемых логических ячеек LAU

**Базовый адрес:** 400A\_E000h

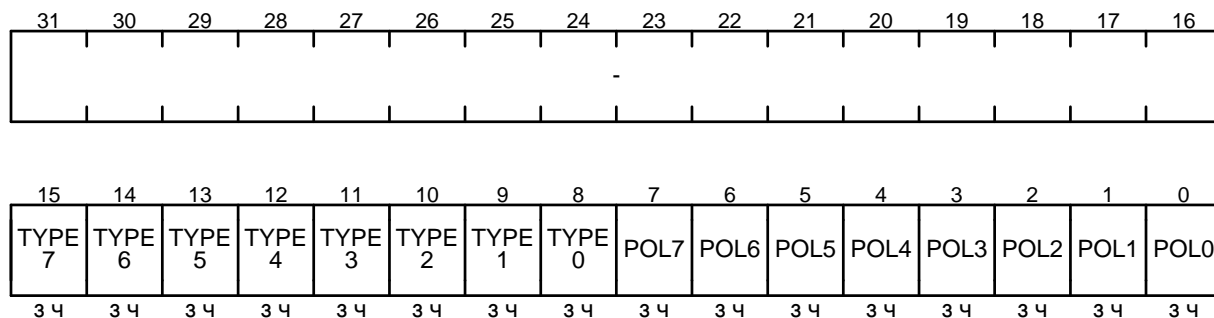
<b>Смещение:</b>	+ 020h (LM0) + 098h (LM1) + 110h (LM2) + 188h (LM3) + 200h (LM4) + 278h (LM5) + 2F0h (LM6) + 368h (LM7) + LMn + 18h (LUT0) + LMn + 24h (LUT1) + LMn + 30h (LUT2) + LMn + 3Ch (LUT3) + LMn + 48h (LUT4) + LMn + 54h (LUT5) + LMn + 60h (LUT6) + LMn + 6Ch (LUT7)	Регистры логической макроячейки LM0 Регистры логической макроячейки LM1 Регистры логической макроячейки LM2 Регистры логической макроячейки LM3 Регистры логической макроячейки LM4 Регистры логической макроячейки LM5 Регистры логической макроячейки LM6 Регистры логической макроячейки LM7 Регистры логического блока LUT0 Регистры логического блока LUT1 Регистры логического блока LUT2 Регистры логического блока LUT3 Регистры логического блока LUT4 Регистры логического блока LUT5 Регистры логического блока LUT6 Регистры логического блока LUT7
------------------	--	--

**Мнемоника:** LMn;  
LUTi;

Примечание – n – номер логической макроячейки LM от 0 до 7;  
i – номер логического блока LUT от 0 до 7

### INTCTL – регистр управления прерываниями

Смещение: + 00h  
Сброс: 0000\_0000h



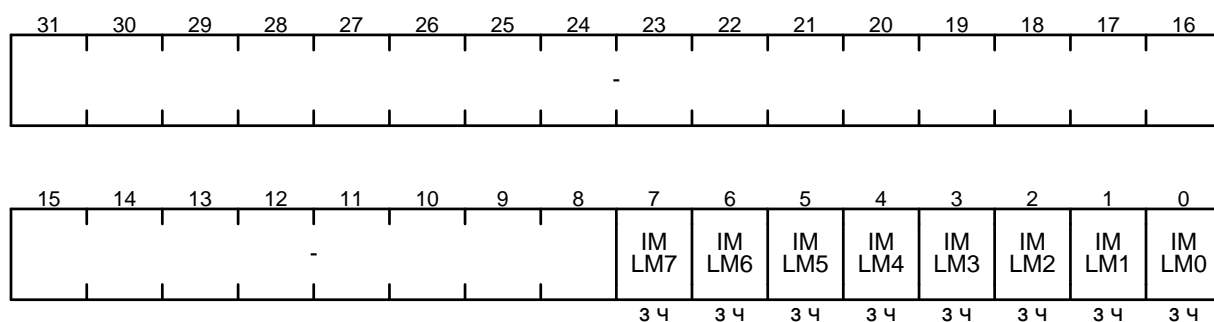
Поле	Биты	Описание
TYPE <sub>n</sub>	15-8	Типы прерываний блоков LM <sub>n</sub>
		0   Прерывание по уровню
		1   Прерывание по фронту
POL <sub>n</sub>	7-0	Полярность сигналов прерываний блоков LM <sub>n</sub>
		0   Прерывание по низкому уровню/спаду
		1   Прерывание по высокому уровню/фронту
–	31-16	Зарезервировано



### IMSC – регистр маски прерываний

Смещение: + 04h

Сброс: 0000\_0000h

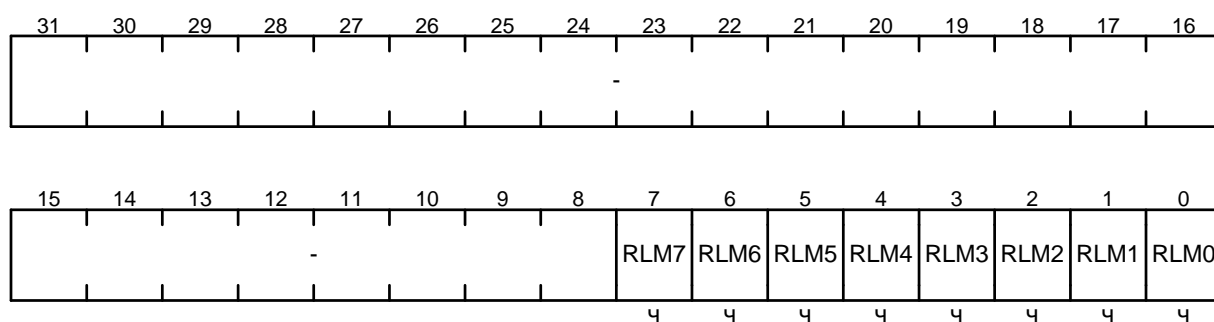


Поле	Биты	Описание
IMLMn	7-0	Разрешение прерывания блока LMn
–	31-8	Зарезервировано

### RIS – регистр не маскированных прерываний

Смещение: + 08h

Сброс: 0000\_0000h

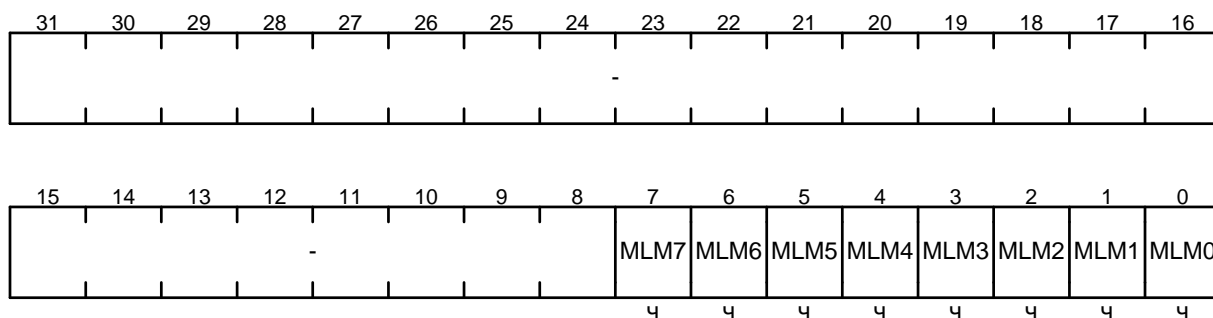


Поле	Биты	Описание
RLMn	7-0	Статус немаскированных прерываний блока LMn
–	31-8	Зарезервировано

### MIS – регистр маскированных прерываний

Смещение: + 0Ch

Сброс: 0000\_0000h

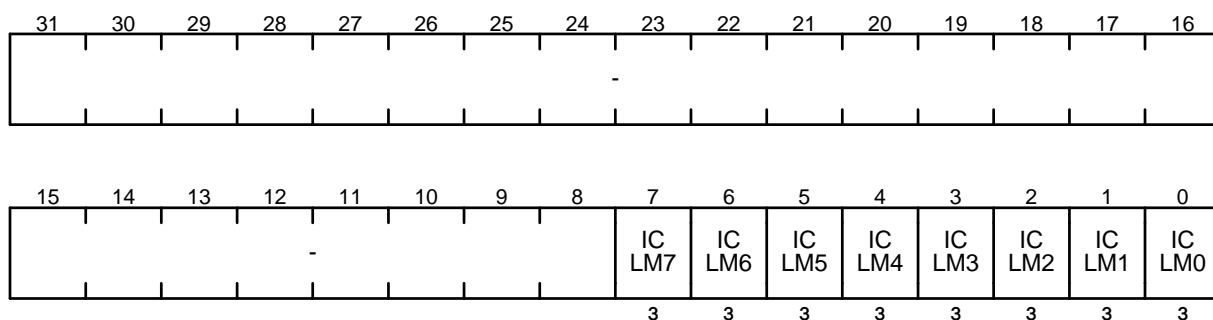


Поле	Биты	Описание
MLMn	7-0	Статус маскированных прерываний блока LMn
-	31-8	Зарезервировано

### ICR – регистр сброса прерываний

Смещение: + 10h

Сброс: 0000\_0000h

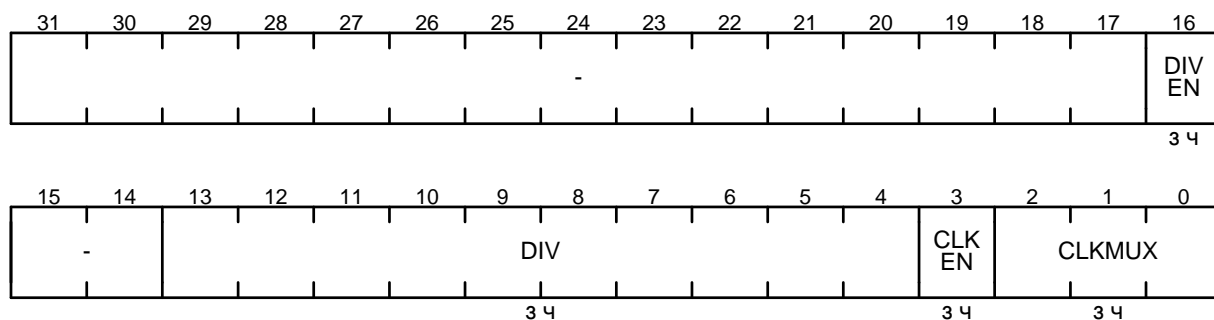


Поле	Биты	Описание
ICLMn	7-0	Сброс прерываний блока LMn
-	31-8	Зарезервировано

## CLKCTL – регистр выбора тактового сигнала блока LMn

Смещение: LMn + 00h

Сброс: 0000\_0000h

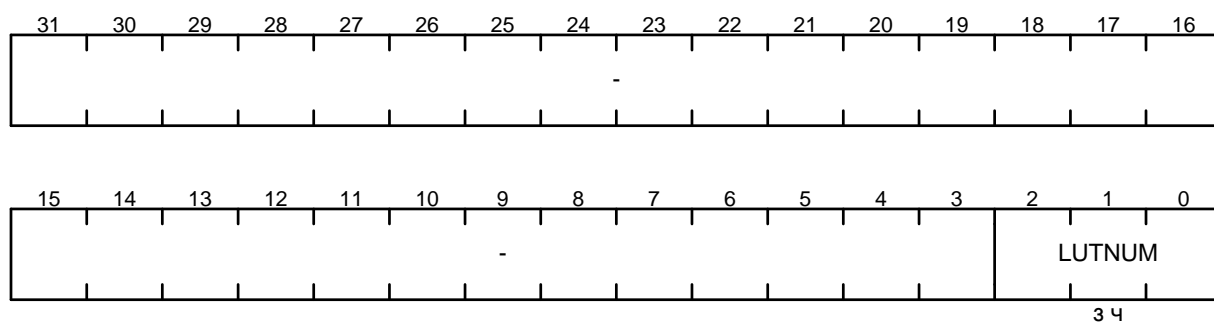


Поле	Биты	Описание	
DIVEN	16	Бит включения делителя	
DIV	13-4	Поле задания делителя. Коэффициент деления определяется как $2 \times (DIV + 1)$	
CLKEN	3	Разрешение подачи тактового сигнала	
CLKMUX	2-0	Выбор источника тактового сигнала	
		0	Тактовый сигнал PCLK
		1	Внешний сигнал LAU_CLK[0]
		2	Внешний сигнал LAU_CLK[1]
		3	Внешний сигнал LAU_CLK[2]
4	Внешний сигнал LAU_CLK[3]		
–	31-17, 15-14	Зарезервировано	

### INTSEL – регистр выбора источника прерывания блока LMn

Смещение: LMn + 04h

Сброс: 0000\_0000h

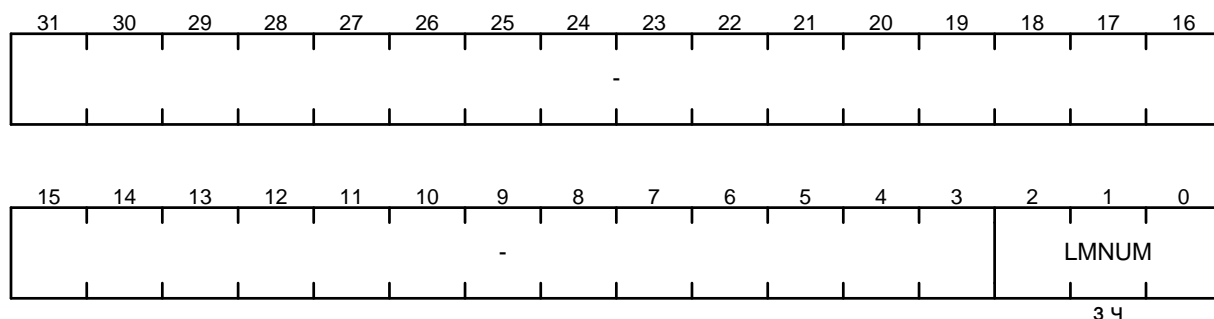


Поле	Биты	Описание
LUTNUM	2-0	Номер LUT, сигнал которого будет использоваться в качестве выходного прерывания блока LMn
–	31-3	Зарезервировано

### BUSSEL – регистр выбора источника сигнала LM\_BUS

Смещение: LMn + 08h

Сброс: 0h

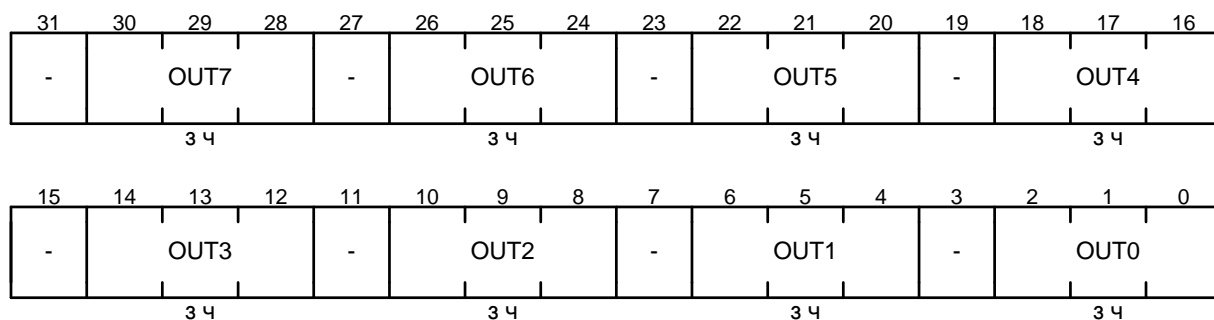


Поле	Биты	Описание
LMNUM	2-0	Номер LUT, сигнал которого будет выведен в локальную шину LM_BUS
–	31-3	Зарезервировано

## OUTMUX – регистр выбора источника сигналов LM\_OUT

Смещение: LMn + 0Ch

Сброс: 0000\_0000h

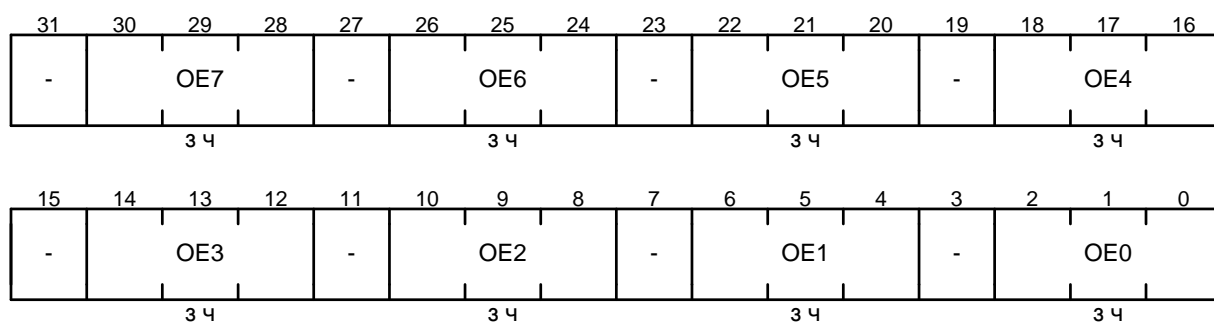


Поле	Биты	Описание
OUT0	2-0	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[0]
OUT1	6-4	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[1]
OUT2	10-8	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[2]
OUT3	14-12	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[3]
OUT4	18-16	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[4]
OUT5	22-20	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[5]
OUT6	26-24	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[6]
OUT7	30-28	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[7]
-	31, 27, 23, 19, 15, 11, 7, 3	Зарезервировано

## OEMUX – регистр выбора источника сигналов LM\_OE

Смещение: LMn + 10h

Сброс: 0000\_0000h

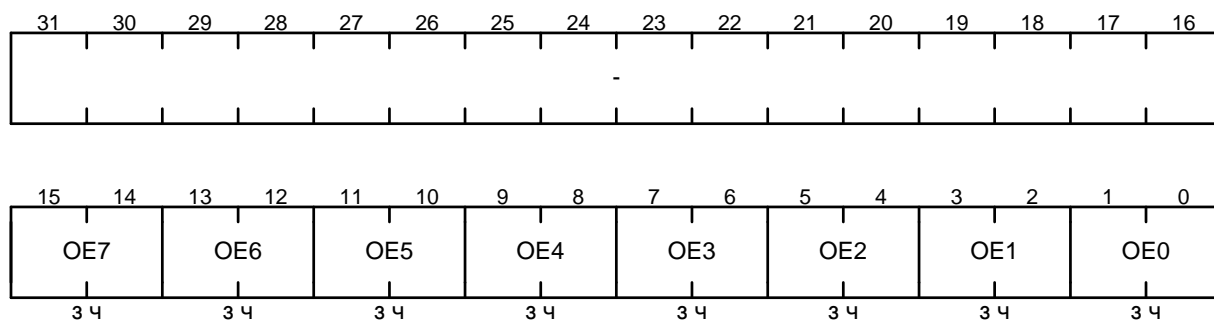


Поле	Биты	Описание
OE <sub>i</sub>	2-0, 6-4, 10-8, 14-12, 18-16, 22-20, 26-24, 30-28	Номер LUT, сигнал которого будет выбран в качестве LM_OUT[i]
–	31, 27, 23, 19, 15, 11, 7, 3	Зарезервировано

## OECTL – регистр настройки сигналов LM\_OE

Смещение: LMn + 14h

Сброс: 0000\_0000h

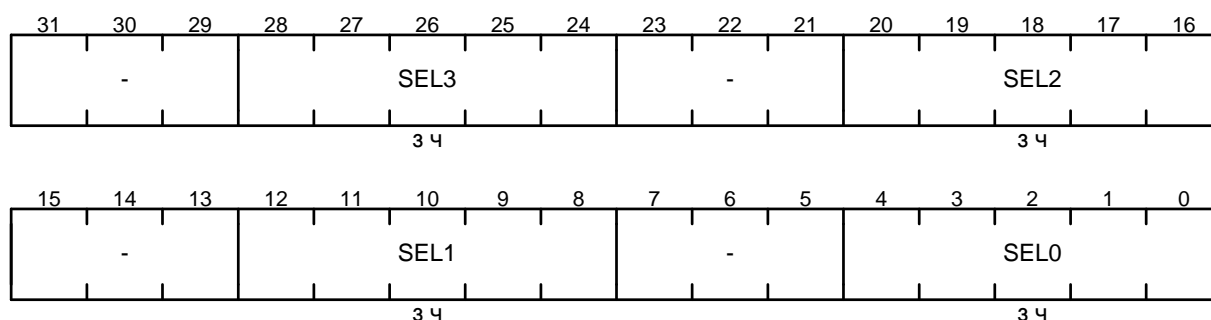


Поле	Биты	Описание	
OEi	15-14,	Выбор способа формирования сигнала OEi	
	13-12,	00 b	Логический ноль, OE в режиме входа
	11-10,		
	9-8,	01 b	Определяется сигналом, выбранным в соответствующем поле регистра OEMUX
	7-6,		
	5-4,	10 b	Определяется сигналом, выбранным в соответствующем поле регистра OEMUX
3-2,			
1-0	11 b	Логическая единица, OE в режиме выхода	
–	31-16	Зарезервировано	

## LUTMUXi – регистр настройки мультиплексора LUTi блока LMn

Смещение: LUTi + 00h

Сброс: 0000\_0000h



Поле	Биты	Описание
SELi	28-24, 20-16, 12-8, 4-0	Выбор входного i-го сигнала LUT
		0   Логический ноль
		1   Логическая единица
		2   Логический ноль
		3   Прерывание TMR0
		4   Прерывание TMR1
		5   Прерывание TMR2
		6   Прерывание TMR3
		7   Внешний сигнал LM_IN[0]
		8   Внешний сигнал LM_IN[1]
		9   Внешний сигнал LM_IN[2]
		10   Внешний сигнал LM_IN[3]
		11   Внешний сигнал LM_IN[4]
		12   Внешний сигнал LM_IN[5]
		13   Внешний сигнал LM_IN[6]
		14   Внешний сигнал LM_IN[7]
		15   Выходной сигнал LUT[0] данного LM блока
		16   Выходной сигнал LUT[1] данного LM блока
		17   Выходной сигнал LUT[2] данного LM блока
		18   Выходной сигнал LUT[3] данного LM блока
		19   Выходной сигнал LUT[4] данного LM блока
		20   Выходной сигнал LUT[5] данного LM блока
		21   Выходной сигнал LUT[6] данного LM блока
		22   Выходной сигнал LUT[7] данного LM блока
23   Сигнал межблочной коммутации LM_BUS[0]		

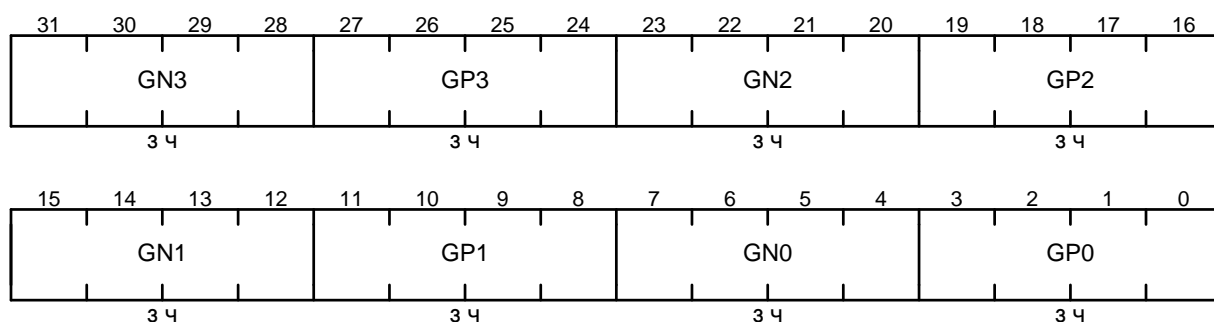


Поле	Биты	Описание
SELi		24   Сигнал межблочной коммутации LM_BUS[1]
		25   Сигнал межблочной коммутации LM_BUS[2]
		26   Сигнал межблочной коммутации LM_BUS[3]
		27   Сигнал межблочной коммутации LM_BUS[4]
		28   Сигнал межблочной коммутации LM_BUS[5]
		29   Сигнал межблочной коммутации LM_BUS[6]
		30   Сигнал межблочной коммутации LM_BUS[7]
–	31-29, 23-21, 15-13, 7-5	Зарезервировано

**LUTGATEi – регистр настройки схемы предварительной комбинации сигналов LUTi блока LMn**

Смещение: LUTi + 04h

Сброс: 0000\_0000h

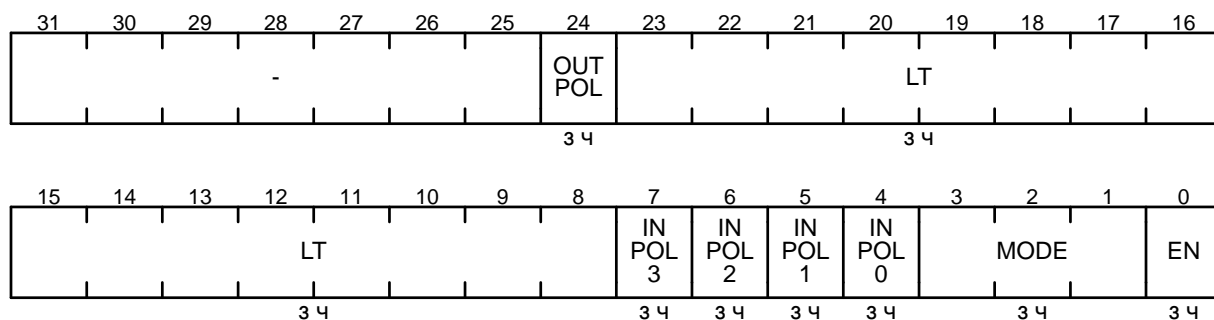


Поле	Биты	Описание
GP0	3-0	Разрешение влияния прямых сигналов на 0 бит данных PLF
GN0	7-4	Разрешение влияния инверсных сигналов на 0 бит данных PLF
GP1	11-8	Разрешение влияния прямых сигналов на 1 бит данных PLF
GN1	15-12	Разрешение влияния инверсных сигналов на 1 бит данных PLF
GP2	19-16	Разрешение влияния прямых сигналов на 2 бит данных PLF
GN2	23-20	Разрешение влияния инверсных сигналов на 2 бит данных PLF
GP3	27-24	Разрешение влияния прямых сигналов на 3 бит данных PLF
GN3	31-28	Разрешение влияния инверсных сигналов на 3 бит данных PLF

## LUTPLFi – регистр настройки логической функции LUTi блока LMn

Смещение: LUTi + 08h

Сброс: 0000\_0000h



Поле	Биты	Описание	
OUTPOL	24	Включение инверсии выхода логической функции	
LT	23-8	Вектор задания значений табличной функции	
INPOL	7-4	Включение инверсии входных сигналов	
MODE	3-1	Тип логической функции	
		0	Нет функции, логический ноль на выходе
		1	Таблица истинности
		2	Асинхронный D-триггер с входом разрешения загрузки LE
		3	Асинхронный RS-триггер
		4	Синхронный D-триггер со входами установки S и сброса R
		5	Синхронный D-триггер с входом сброса R
		6	Синхронный D-триггер с входом разрешения загрузки LE
7	Синхронный JK-триггер		
EN	0	Разрешение выхода логической функции	
–	31-25	Зарезервировано	

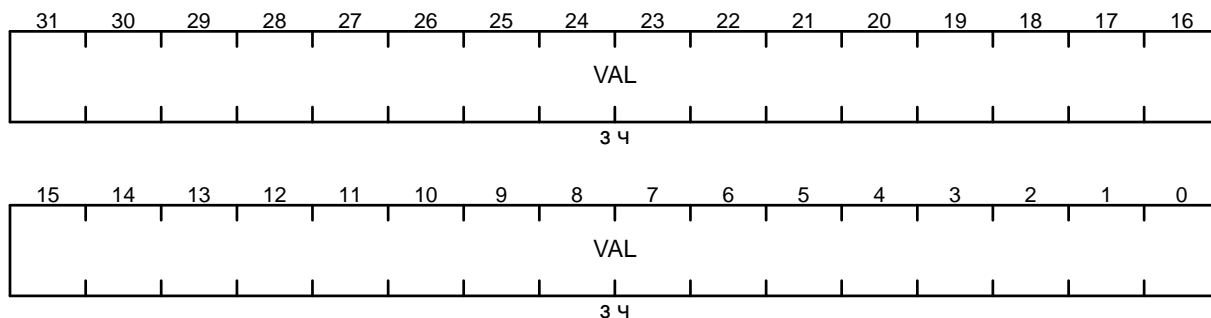
## A.12 Регистры блока тригонометрических вычислений TMU

Базовый адрес: 4004\_0000h

### XIN - регистр аргумента X

Смещение: + 00h

Сброс: 0000\_0000h

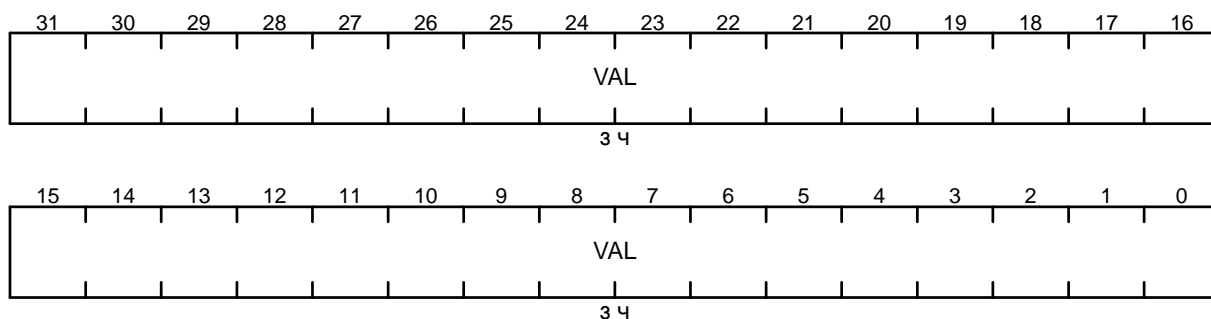


Поле	Биты	Описание
VAL	31-0	Значение входного аргумента X

### YIN - регистр аргумента Y

Смещение: + 04h

Сброс: 0000\_0000h

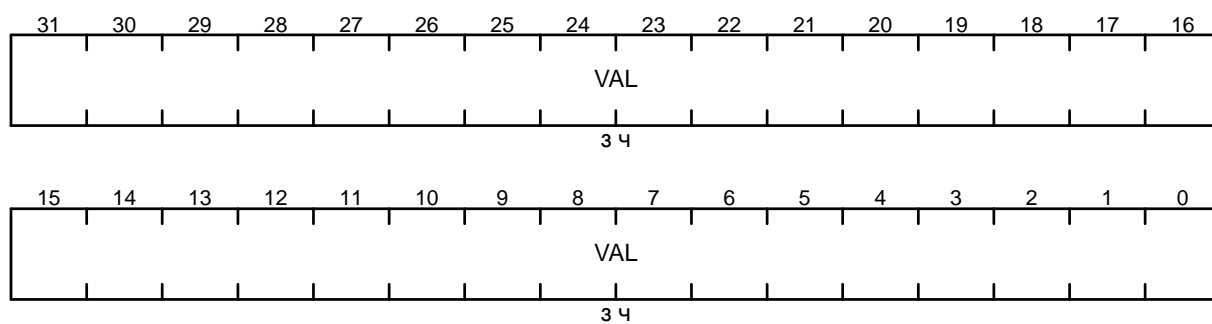


Поле	Биты	Описание
VAL	31-0	Значение входного аргумента Y

## PHIN – регистр аргумента фазы

Смещение: + 08h

Сброс: 0000\_0000h

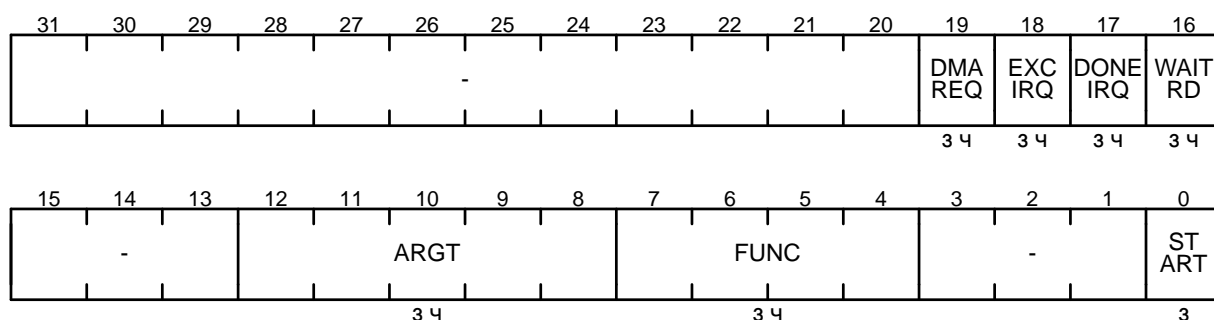


Поле	Биты	Описание
VAL	31-0	Значение входного аргумента фазы

## CMD – регистр команд

Смещение: + 0Ch

Сброс: 0000\_0000h

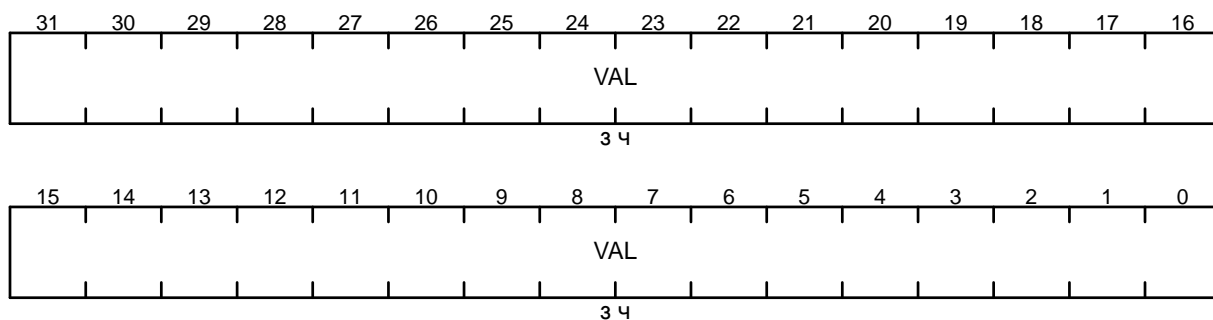


Поле	Биты	Описание
DMAREQ	19	Разрешение генерации запроса к DMA
		0   Запрос запрещен 1   Запрос разрешен
EXCIRQ	18	Разрешение прерывания по появлению любого из исключений
		0   Прерывание запрещено 1   Прерывание разрешено
DONEIRQ	17	Разрешение прерывания по окончанию вычисления
		0   Прерывание запрещено 1   Прерывание разрешено
WAITRD	16	Разрешение блокирующего чтения результатов
		0   Регистры XOUT, YOUT, RHOУT могут быть прочитаны только после окончания вычисления 1   Любой из регистров XOUT, YOUT, RHOУT может быть прочитан сразу после старта вычислений. Но сама процедура чтения завершится лишь после их окончания.
ARGT	12-8	Выбор формата аргументов и результатов
		0h   Q28
		1h   Q27
		2h   Q26
		...
		14h   Q8
		15h-1Eh   Зарезервировано 1Fh   IEEE 754
FUNC	7-4	Выбор функции
		0h   Вычисление синуса и косинуса
		1h   Перевод из прямоугольной системы координат в полярную
		2h   Перевод из полярной системы координат в прямоугольную 3h   Поворот вектора на заданный угол
START	0	Запись единицы запускает вычисление
-	31-20, 15-13, 3-1	Зарезервировано

### XOUT – регистр результата X

Смещение: + 10h

Сброс: 0000\_0000h

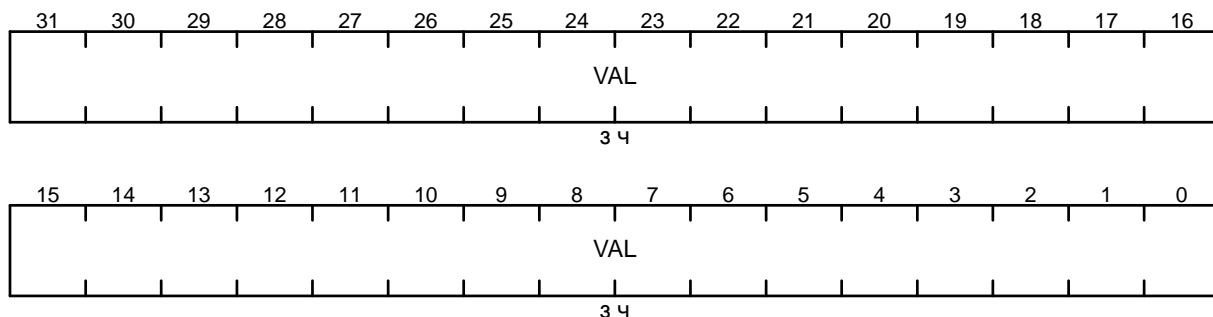


Поле	Биты	Описание
VAL	31-0	Значение результата X

### YOUT - регистр результата Y

Смещение: + 14h

Сброс: 0000\_0000h

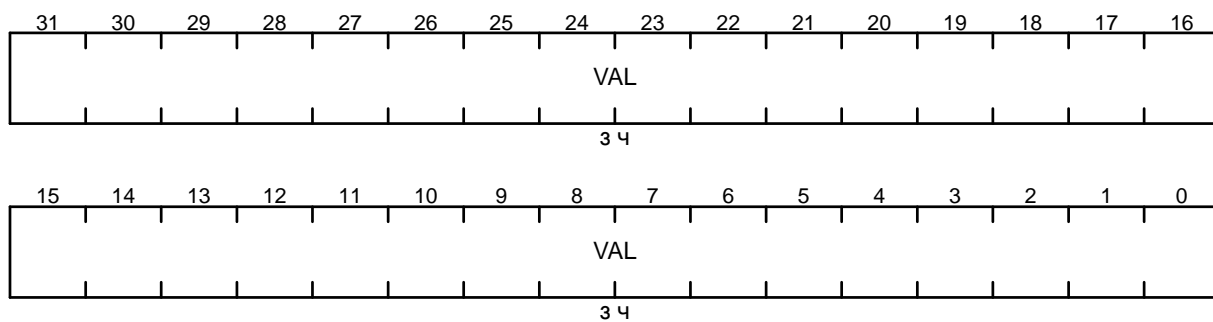


Поле	Биты	Описание
VAL	31-0	Значение результата Y

## PHOUT – регистр результата фазы

Смещение: + 18h

Сброс: 0000\_0000h

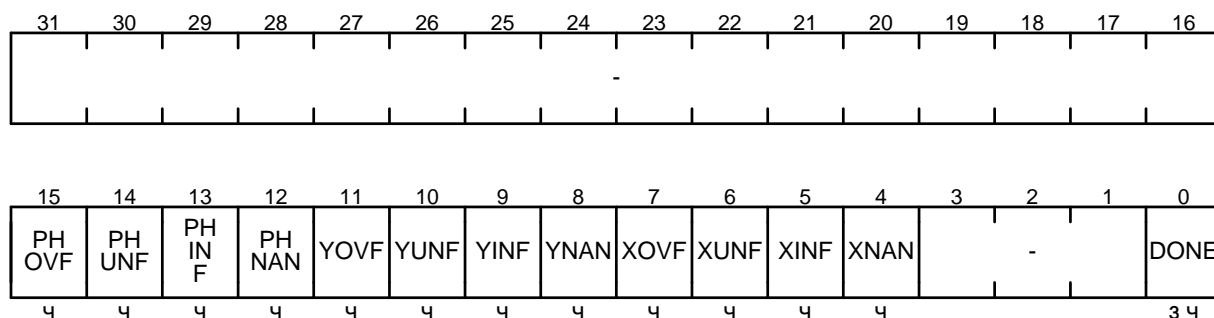


Поле	Биты	Описание
VAL	31-0	Значение результата фазы

## STAT – регистр статуса

Смещение: + 1Ch

Сброс: 0000\_0000h



Поле	Биты	Описание
PHOVF	15	Флаг исключения - переполнение Q28 в PHIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
PHUNF	14	Флаг исключения - недозаполнение Q28 в PHIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
PHINF	13	Флаг исключения - бесконечность (Inf) в PHIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
PHNAN	12	Флаг исключения - нечисло (NaN) в PHIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
YOVSF	11	Флаг исключения - переполнение Q28 в YIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
YUNF	10	Флаг исключения - недозаполнение Q28 в YIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
YINF	9	Флаг исключения - бесконечность (Inf) в YIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления



Поле	Биты	Описание
YNAN	8	Флаг исключения - нечисло (NaN) в YIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
XOVF	7	Флаг исключения - переполнение Q28 в XIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
XUNF	6	Флаг исключения - недозаполнение Q28 в XIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
XINF	5	Флаг исключения - бесконечность (Inf) в XIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
XNAN	4	Флаг исключения - нечисло (NaN) в XIN
		0   Нет исключения
		1   Произошло исключение
		Флаг сбрасывается автоматически при старте нового вычисления
DONE	0	Флаг окончания вычисления
		0   Вычисление не завершено
		1   Вычисление завершено
		Флаг сбрасывается записью единицы
–	31-12, 7-2	Зарезервировано

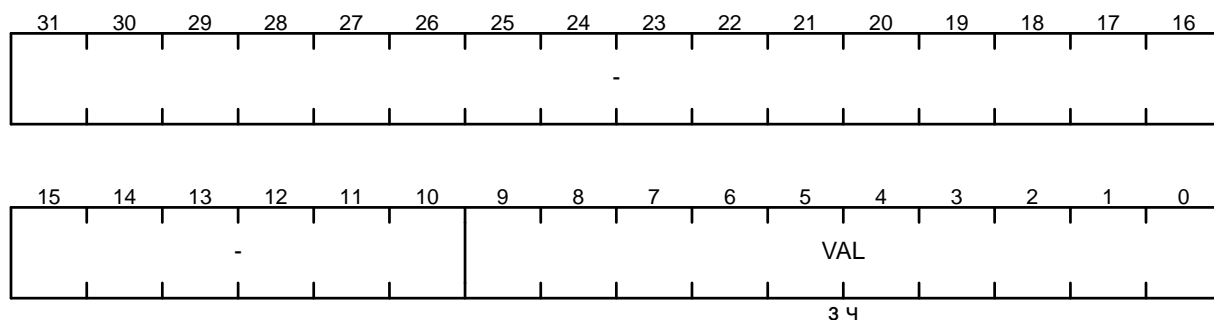
### A.13 Регистры часов реального времени RTC

Базовый адрес: 4009\_9000h

#### POS – регистр долей секунд

Смещение: + 00h

Сброс: 0000\_0000h

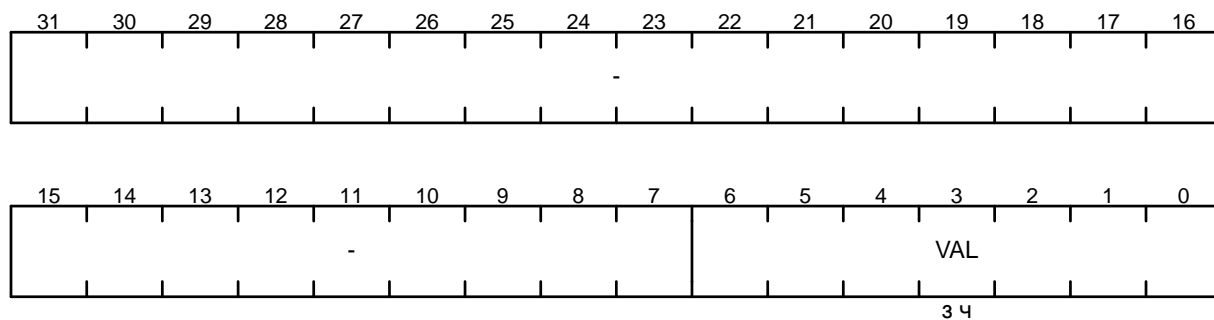


Поле	Биты	Описание
VAL	9-0	Доля секунды (Двоичный формат). Корректные значения: 000h – 3FFh
–	31-10	Зарезервировано

#### SEC – регистр секунд

Смещение: + 04h

Сброс: 0000\_0000h

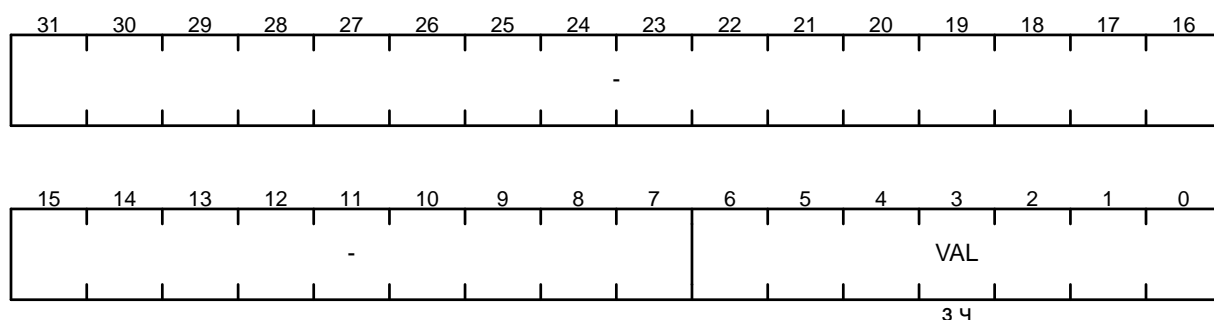


Поле	Биты	Описание
VAL	6-0	Секунда (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h
–	31-7	Зарезервировано

### MIN – регистр минут

Смещение: + 08h

Сброс: 0000\_0000h

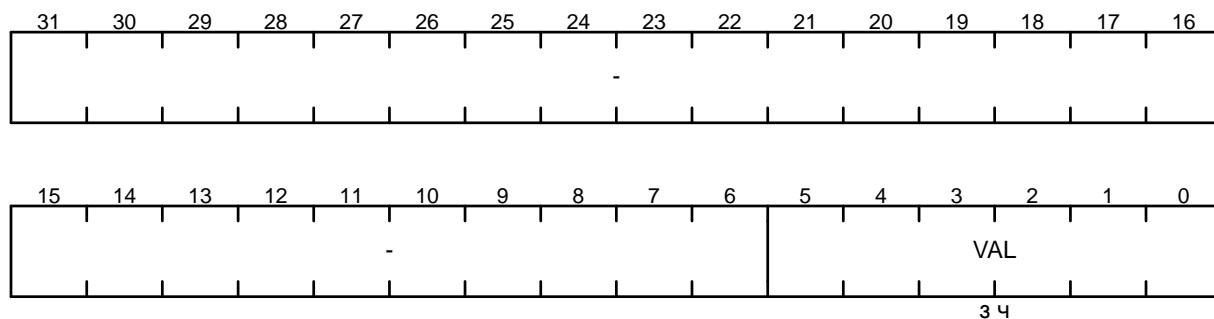


Поле	Биты	Описание
VAL	6-0	Минута (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h
–	31-7	Зарезервировано

### HOUR – регистр часов

Смещение: + 10h

Сброс: 16h

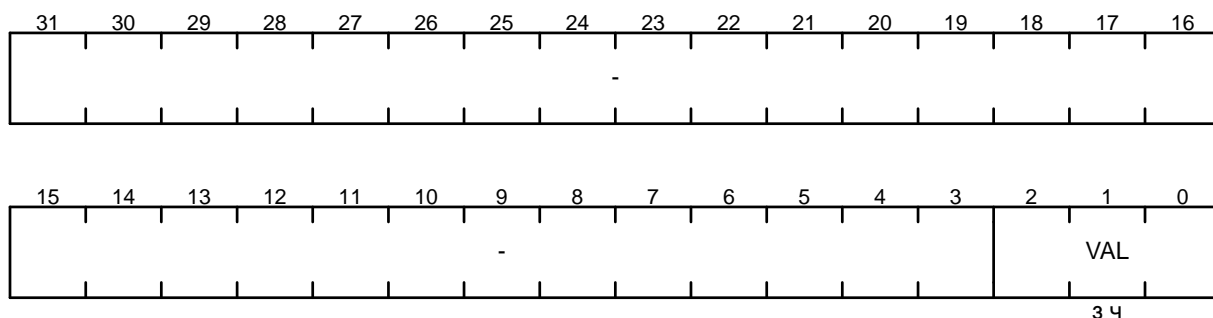


Поле	Биты	Описание
VAL	5-0	Час (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 23h
–	31-6	Зарезервировано

### DOW – регистр дней недели

Смещение: + 18h

Сброс: 4h

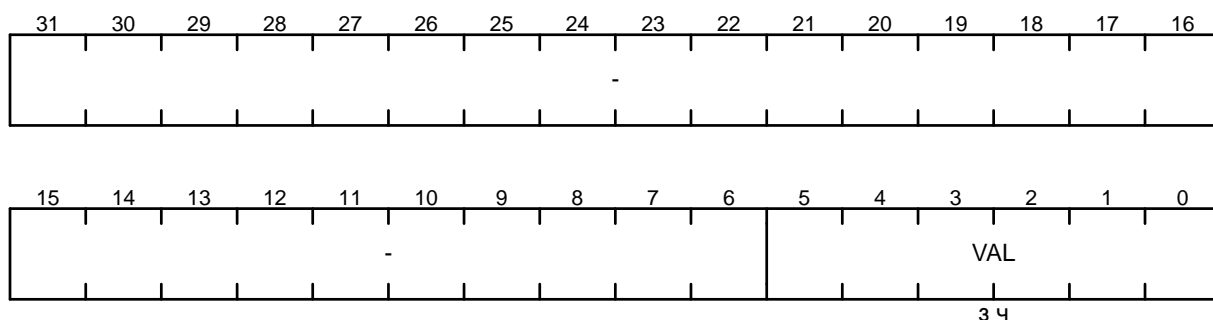


Поле	Биты	Описание
VAL	3-0	День недели (BCD формат). Корректные значения: 00h – 07h
–	31-4	Зарезервировано

### DAY – регистр дней месяца

Смещение: + 20h

Сброс: 14h

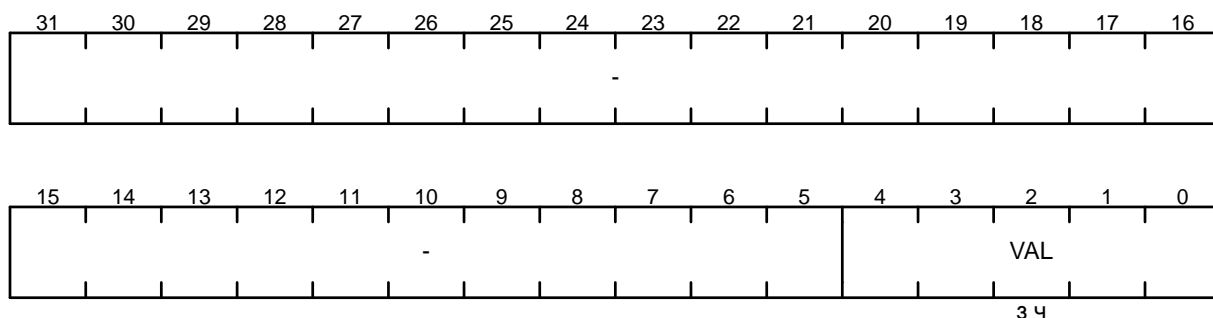


Поле	Биты	Описание
VAL	5-0	Число (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 31h
–	31-6	Зарезервировано

### MONTH – регистр месяцев

Смещение: + 24h

Сброс: 9h

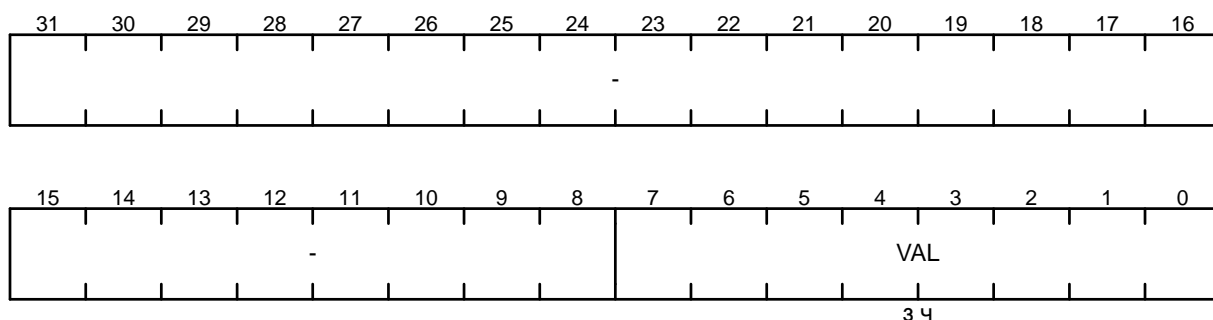


Поле	Биты	Описание
VAL	4-0	Месяц (BCD формат). Корректные значения: 00h – 09h, 10h – 12h
–	31-5	Зарезервировано

### YEAR – регистр лет

Смещение: + 28h

Сброс: 48h

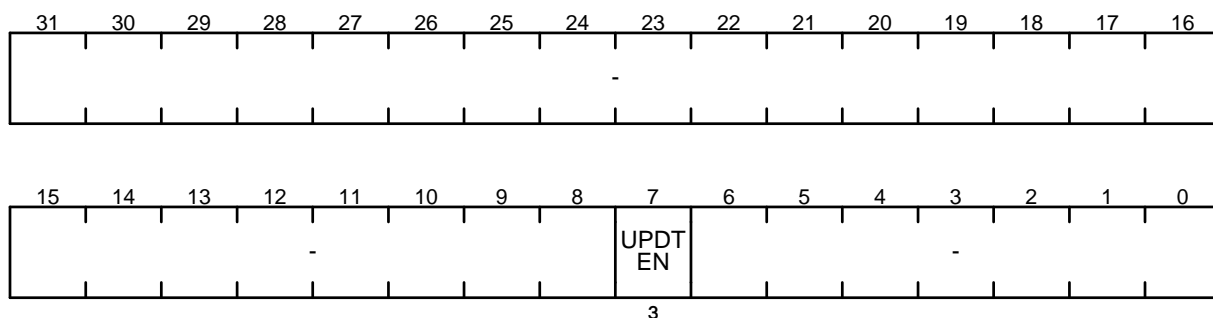


Поле	Биты	Описание
VAL	7-0	Год (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h, 60h – 69h, 70h – 79h, 80h – 89h, 90h – 99h
–	31-8	Зарезервировано

### SHDW – регистр управления теневого загрузкой

Смещение: + 2Ch

Сброс: 80h

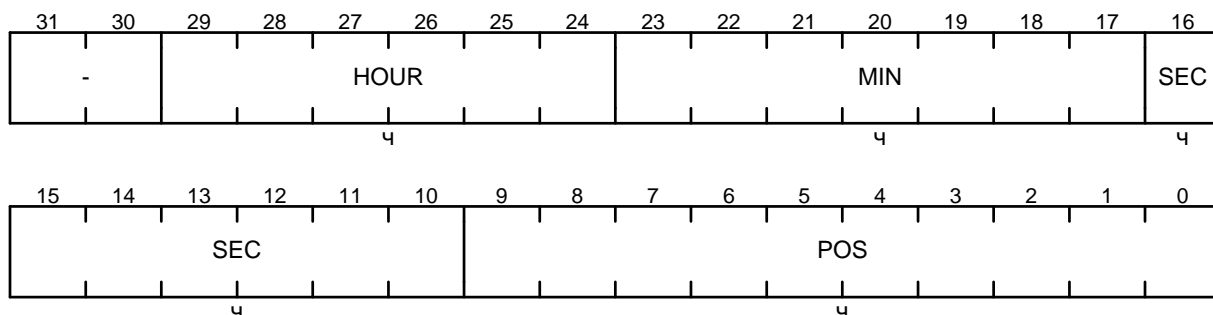


Поле	Биты	Описание
UPDTEN	7	Бит разрешения обновлений теневого регистров. По умолчанию равен единице
-	31-8, 6-0	Зарезервировано

### TIME – общий регистр времени

Смещение: + 30h

Сброс: 1600\_0000h



Поле	Биты	Описание
HOUR	29-24	Час (BCD формат).
MIN	23-17	Минута (BCD формат).
SEC	16-10	Секунда (BCD формат).
POS	9-0	Доли секунд (двоичный формат).
-	31-30	Зарезервировано

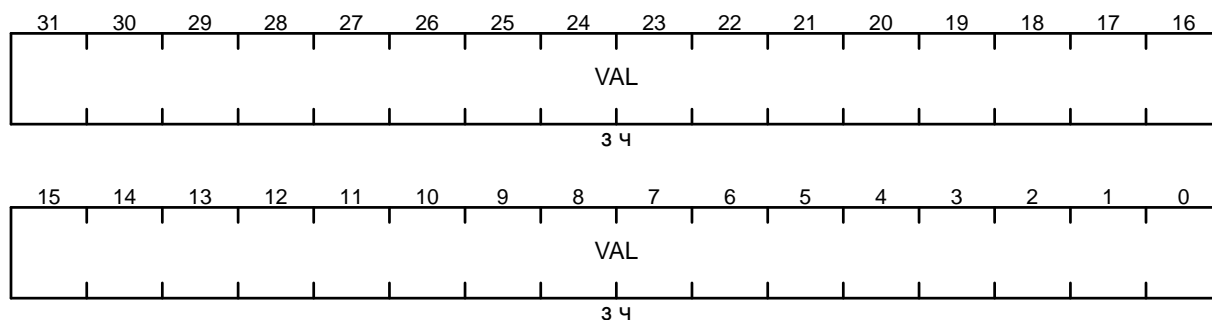
## A.14 Регистры сторожевого таймера WDT

Базовый адрес: 4008\_3000h

### LOAD – регистр загрузки сторожевого таймера

Смещение: + 00h

Сброс: FFFF\_FFFFh

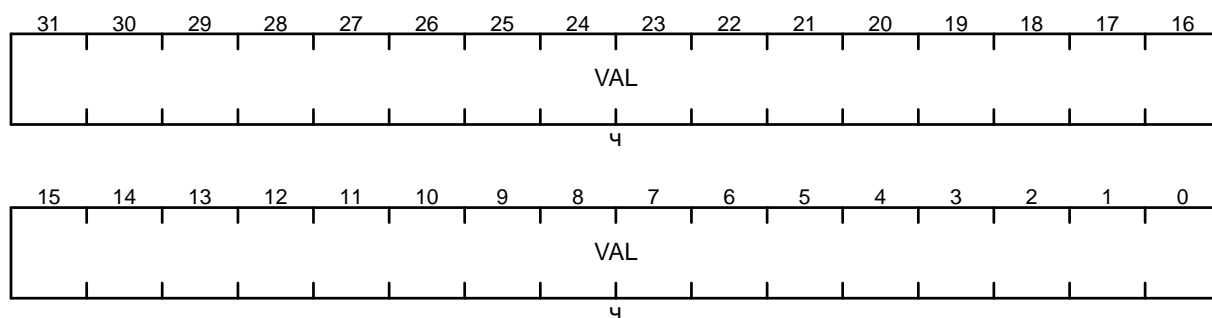


Поле	Биты	Описание
VAL	31-0	32-разрядный регистр, хранящий начальное значение счетчика. Когда происходит запись в этот регистр, счетчик сразу иницируется этим новым значением. Минимальное допустимое значение 0000_0001h

### VALUE – регистр значения счетчика сторожевого таймера

Смещение: + 04h

Сброс: FFFF\_FFFFh

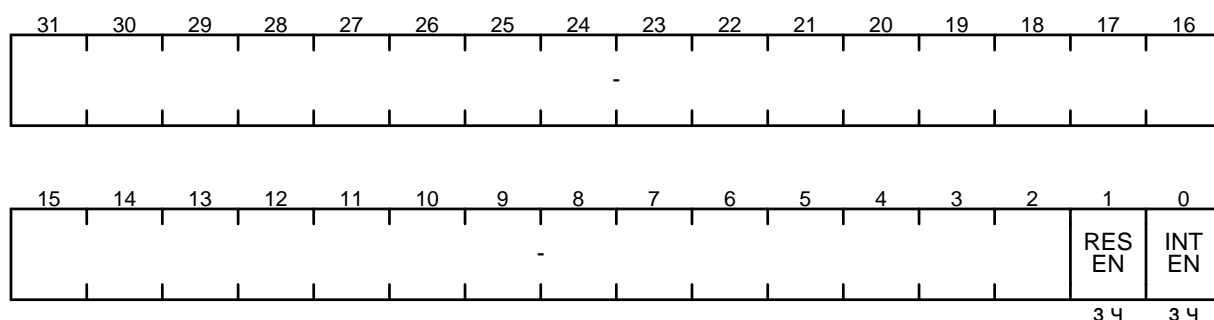


Поле	Биты	Описание
VAL	31-0	32-разрядный регистр текущего значения счетчика

## CTRL – регистр управления сторожевого таймера

Смещение: + 08h

Сброс: 0h

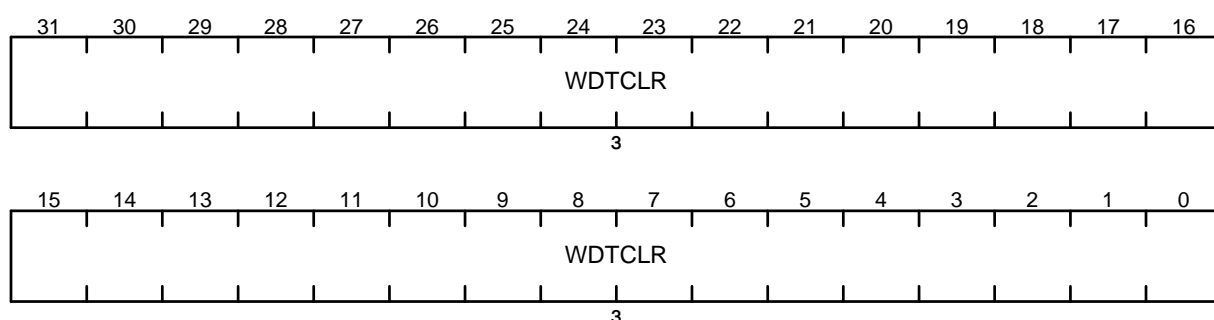


Поле	Биты	Описание
RESEN	1	Бит разрешения сброса микроконтроллера по сторожевому таймеру. Работает по функции «Логическое И» с битом INTEN регистра WDTCTRL
		0   Сброс бита выключает сброс
		1   Установка включает сброс
INTEN	0	Бит включения счета и разрешения прерывания сторожевого таймера
		0   Сброс бита выключает счетчик и снимает прерывание
		1   Установка бита включает счетчик и генерирует прерывание. Если счетчик был включен на момент установки бита, то он иницируется значением из регистра LOAD
–	31-2	Зарезервировано

## INTCLR – регистр сброса сторожевого таймера

Смещение: + 0Ch

Сброс: 0h



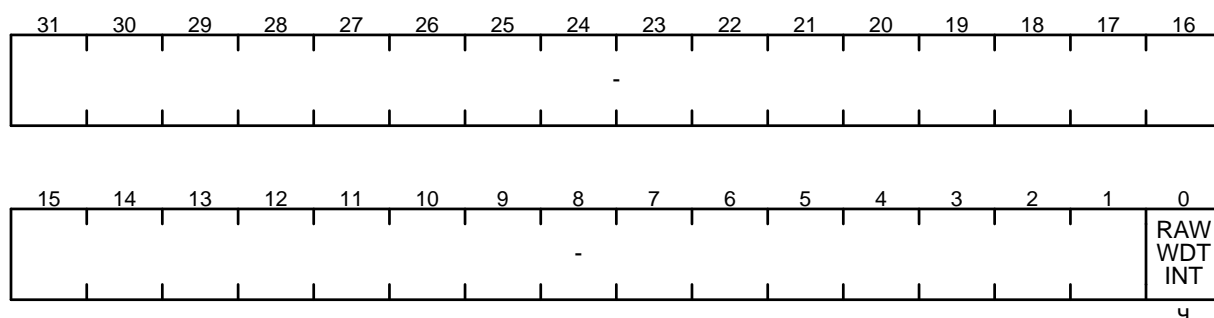
Поле	Биты	Описание
WDTCLR	31-0	32-разрядный регистр сброса сторожевого таймера. Запись любого значения в этот регистр приводит к сбросу прерывания сторожевого таймера и загрузке счетчика значением из регистра LOAD



## RIS – регистр прерывания сторожевого таймера

Смещение: + 10h

Сброс: 0h

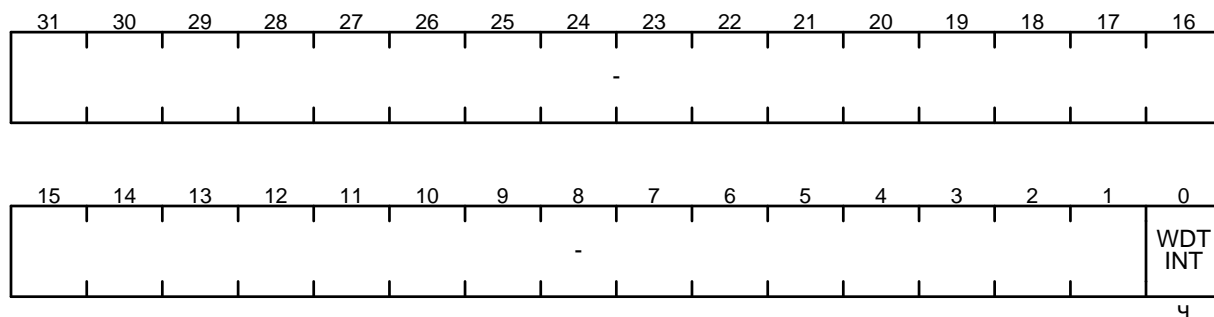


Поле	Биты	Описание
RAWWDTINT	0	Индикатор состояния немаскированного бита прерывания
		0   Сброшен
		1   Установлен
–	31-1	Зарезервировано

## MIS – регистр маскированного прерывания сторожевого таймера

Смещение: + 14h

Сброс: 0h

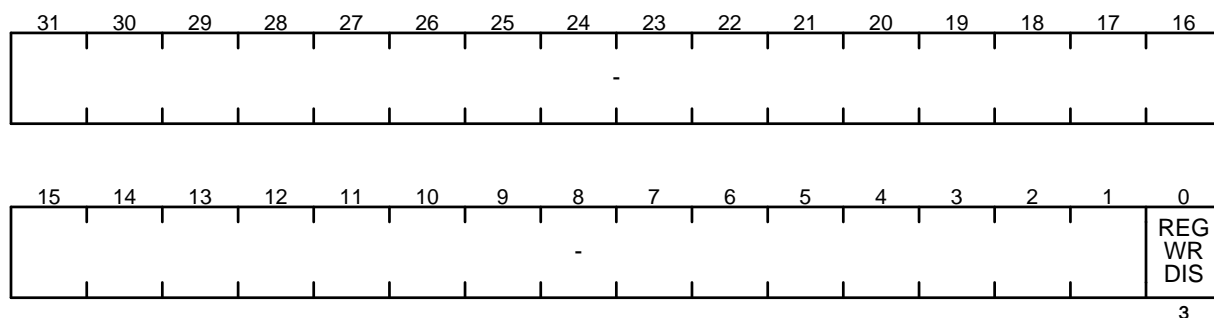


Поле	Биты	Описание
WDTINT	0	Индикатор состояния маскированного бита прерывания. Сигнализирует о появлении маскированного прерывания от счетчика. Состояние бита WDTINT это «логическое И» битов RAWWDTINT и INTEN
		0   Сброшен
		1   Установлен
–	31-1	Зарезервировано

## LOCK – регистр блокировки сторожевого таймера

Смещение: + C00h

Сброс: 0h



Поле	Биты	Описание
REGWRDIS	0	Бит запрета записи во все регистры сторожевого таймера (кроме регистра LOCK). Функция необходима для предотвращения отключения сторожевого таймера сбойными программами
		0   Разрешена (по умолчанию). Для сброса бита следует записать в регистр LOCK значение 1ACCE551h
		1   Запрещена. Для установки бита следует записать в регистр LOCK любое значение, кроме 1ACCE551h
–	31-1	Зарезервировано

## А.15 Регистры блока АЦП

**Базовый адрес:** 400В\_1000h

<b>Смещение:</b>	+ 040h (SEQ0)	Регистры секвенсора 0
	+ 07Ch (SEQ1)	Регистры секвенсора 1
	+ 0B8h (SEQ2)	Регистры секвенсора 2
	+ 0F4h (SEQ3)	Регистры секвенсора 3
	+ 130h (SEQ4)	Регистры секвенсора 4
	+ 16Ch (SEQ5)	Регистры секвенсора 5
	+ 1A8h (SEQ6)	Регистры секвенсора 6
	+ 1E4h (SEQ7)	Регистры секвенсора 7
	+ 400h (DC0)	Регистры цифрового компаратора 0
	+ 40Ch (DC1)	Регистры цифрового компаратора 1
	+ 418h (DC2)	Регистры цифрового компаратора 2
	+ 424h (DC3)	Регистры цифрового компаратора 3
	+ 430h (DC4)	Регистры цифрового компаратора 4
	+ 43Ch (DC5)	Регистры цифрового компаратора 5
	+ 448h (DC6)	Регистры цифрового компаратора 6
	+ 454h (DC7)	Регистры цифрового компаратора 7
	+ 460h (DC8)	Регистры цифрового компаратора 8
	+ 46Ch (DC9)	Регистры цифрового компаратора 9
	+ 478h (DC10)	Регистры цифрового компаратора 10
	+ 484h (DC11)	Регистры цифрового компаратора 11
	+ 490h (DC12)	Регистры цифрового компаратора 12
	+ 49Ch (DC13)	Регистры цифрового компаратора 13
	+ 4A8h (DC14)	Регистры цифрового компаратора 14
	+ 4B4h (DC15)	Регистры цифрового компаратора 15
	+ 4C0h (DC16)	Регистры цифрового компаратора 16
	+ 4CCh (DC17)	Регистры цифрового компаратора 17
	+ 4D8h (DC18)	Регистры цифрового компаратора 18
	+ 4E4h (DC19)	Регистры цифрового компаратора 19
	+ 4F0h (DC20)	Регистры цифрового компаратора 20
	+ 4FCh (DC21)	Регистры цифрового компаратора 21
	+ 508h (DC22)	Регистры цифрового компаратора 22
	+ 514h (DC23)	Регистры цифрового компаратора 23
	+ 540h (ACTL)	Массив регистров настройки АЦП
	+ 580h (CHCTL)	Массив регистров настройки каналов

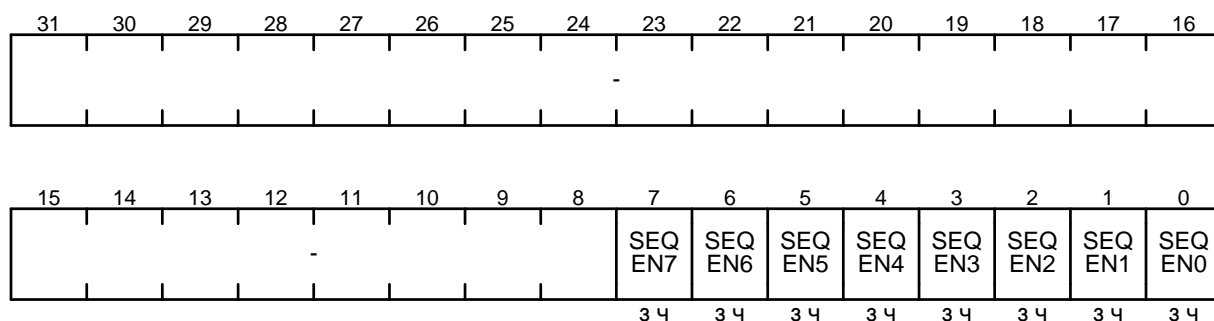
**Мнемоника:** SEQs;  
DCd

**Примечание –** s – номер секвенсора 0 – 7;  
d – номер по порядку цифрового компаратора от 0 до 23.  
m – номер модуля АЦП от 0 до 3  
n – номер канала АЦП от 0 до 47

## SEQEN – регистр включения секвенсоров

Смещение: + 00h

Сброс: 0000\_0000h

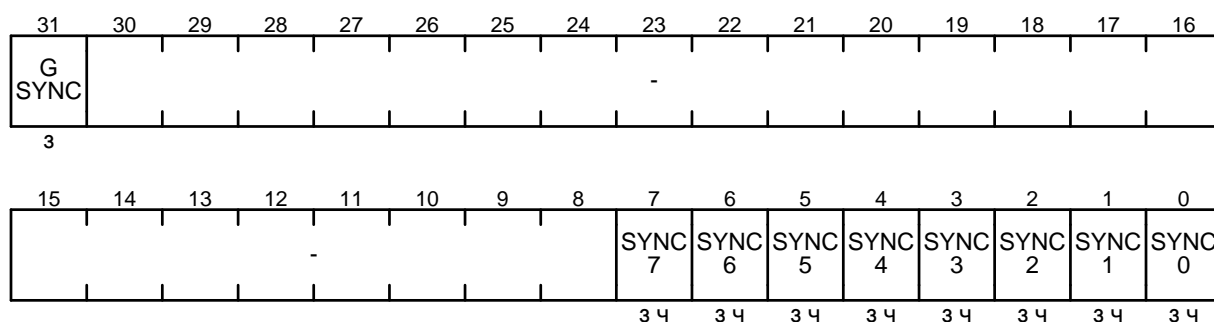


Поле	Биты	Описание
SEQENs	7-0	Бит разрешения работы секвенсора s
		0   Запрещено
		1   Разрешено
–	31-8	Зарезервировано

## SEQSYNC – регистр программной синхронизации секвенсоров

Смещение: 04h

Сброс: 0000\_0000h

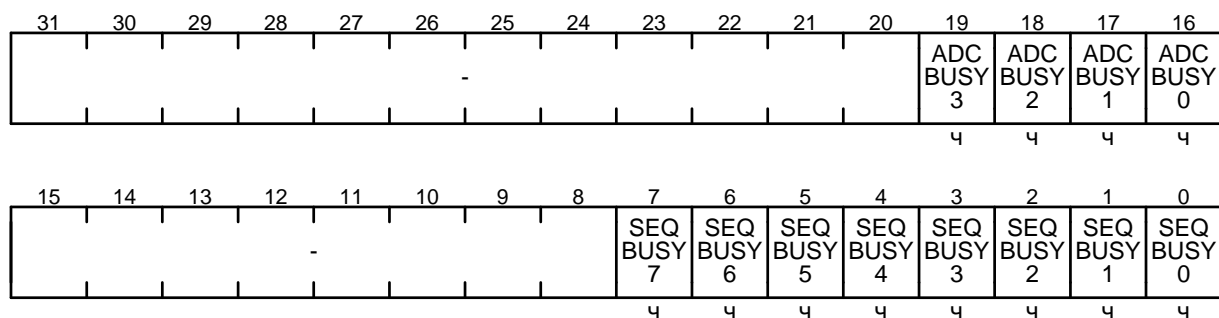


Поле	Биты	Описание
GSYNC	31	Бит запуска секвенсоров. Запись единицы запускает секвенсоры, работа которых разрешена и для которых установлены биты SYNCs.
SYNCs	7-0	Бит разрешения запуска секвенсора s
		0   Запрещено
		1   Разрешено (если установлен бит SEQENs в регистре SEQEN)
–	30-8	Зарезервировано

## BSTAT – регистр флагов занятости

Смещение: 08h

Сброс: 0000\_0000h

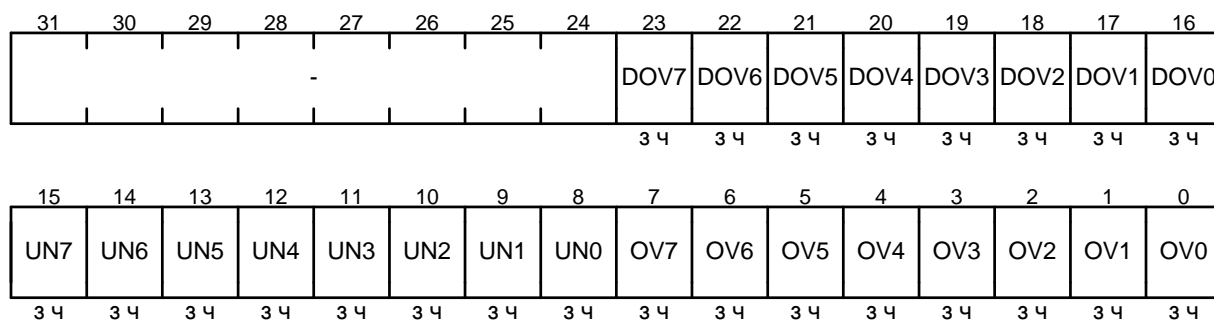


Поле	Биты	Описание
ADCBUSYm	19-16	Флаг занятости модуля АЦП m
		0   АЦП выключен или в режиме ожидания запроса
		1   АЦП проводит измерения по активным запросам
SEQBUSYs	7-0	Флаг занятости секвенсора s
		0   Секвенсор выключен или в режиме ожидания сигнала запуска
		1   Секвенсор производит запуск/перезапуск или выполняет задержку перезапуска
–	31-20, 15-8	Зарезервировано

## FSTAT – регистр флагов буферов результатов и блока DMA

Смещение: 0Ch

Сброс: 0000\_0000h

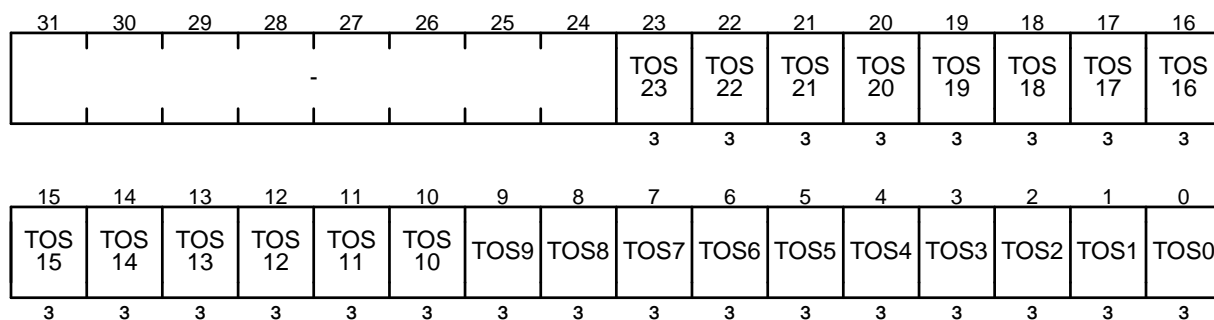


Поле	Биты	Описание	
DOVs	23-16	Флаг ошибки DMA	
		0	Нет ошибки
		1	При наличии обрабатываемого запроса DMA от секвенсора s, пришел еще один запрос, который не может быть обработан
		Флаг сбрасывается записью единицы	
UNs	15-8	Флаг пустоты буфера секвенсора s	
		0	Буфер не пуст
		1	Буфер пуст
		Флаг может быть сброшен программно записью единицы	
OVs	7-0	Флаг заполнения буфера секвенсора s	
		0	В буфере есть как минимум одна свободная ячейка
		1	Буфер заполнен. Все последующие записи в буфер блокируются до появления как минимум одной свободной ячейки
		Флаг сбрасывается записью единицы	
–	31-24	Зарезервировано	

## DCTRIG – регистр сброса флагов выходных триггеров компараторов

Смещение: 10h

Сброс: 0000\_0000h

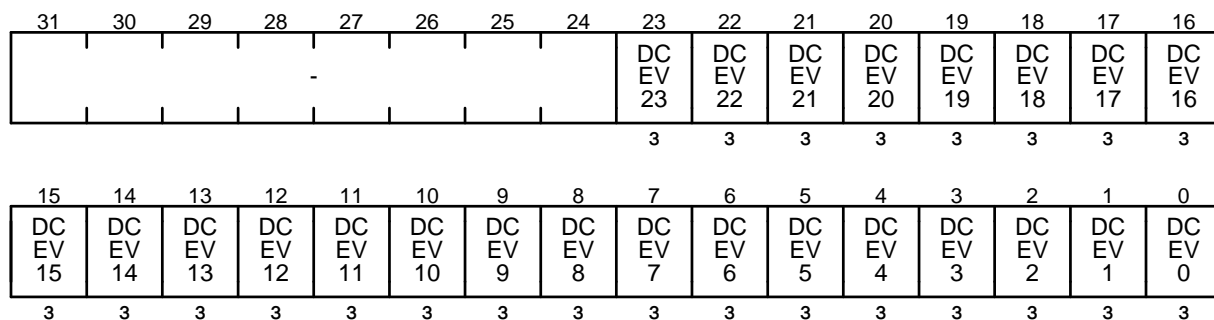


Поле	Биты	Описание
TOSd	23-0	Флаг состояния выходного триггера компаратора d
		0   Триггер не сработал
		1   Триггер сработал
		Флаг сбрасывается записью единицы
–	31-24	Зарезервировано

## DCEV – регистр сброса флагов компараторов

Смещение: 14h

Сброс: 0000\_0000h

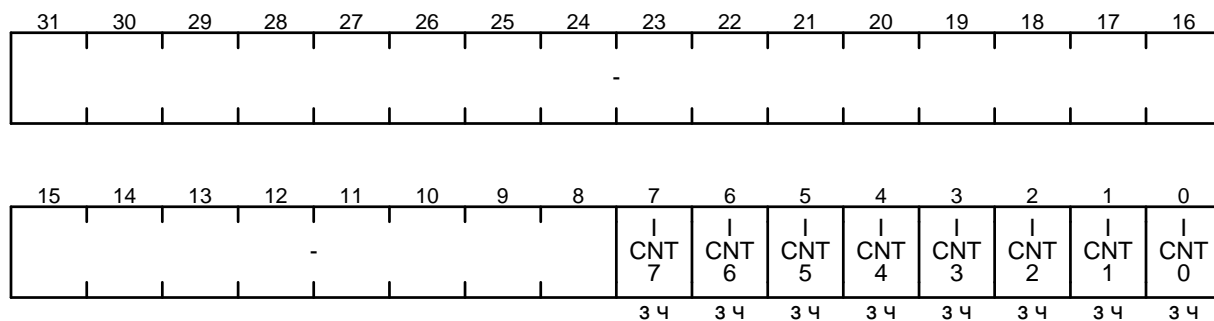


Поле	Биты	Описание
DCEVd	23-0	Флаг события сравнения компаратора d
		0   Сравнение не выполнялось
		1   Сравнение выполнялось
		Флаг сбрасывается записью единицы
–	31-24	Зарезервировано

## ICNT – регистр настройки режима сброса счетчика прерываний

Смещение: 18h

Сброс: 0000\_0000h



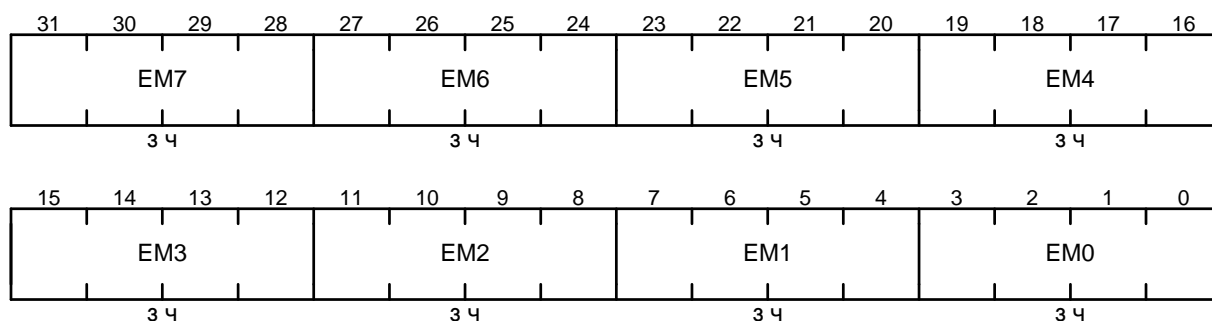
Поле	Биты	Описание
ICNTs	7-0	Бит выбора режима сброса счетчика, используемого для генерации прерываний секвенсора s
		0   Счетчик будет сбрасываться по запуску секвенсора
		1   Запрет сброса счетчика по запуску секвенсора.
–	31 – 8	Зарезервировано



## EMUX – регистр выбора событий запуска секвенсоров

Смещение: 1Ch

Сброс: 0000\_0000h



Поле	Биты	Описание	
EMs	31-0	Поле выбора события для запуска секвенсора s	
		0h	Установка бита GSYNC в регистре SEQSYNC
		1h	Сигнал от GPIOA, GPIOB, GPIOC, GPIOD
		2h	Сигнал от GPIOE, GPIOF, GPIOG, GPIOH
		3h	Сигнал от GPIOJ, GPIOK, GPIOL, GPIOM
		4h	Зарезервировано
		5h	Сигнал от блока TMR0
		6h	Сигнал от блока TMR1
		7h	Сигнал от блока TMR2
		8h	Сигнал от блока TMR3
		9h	Сигналы от блоков PWM0, PWM1, PWM2 – канал А
		Ah	Сигналы от блоков PWM0, PWM1, PWM2 – канал В
		Bh	Сигналы от блоков PWM3, PWM4, PWM5 – канал А
		Ch	Сигналы от блоков PWM3, PWM4, PWM5 – канал В
		Dh	Сигналы от блоков PWM6, PWM7, PWM8, PWM9 – канал А
Eh	Сигналы от блоков PWM6, PWM7, PWM8, PWM9 – канал В		
Fh	Циклическая работа. Активируется после установки бита GSYNC в регистре SEQSYNC		
–	31–8	Зарезервировано	

## RIS – регистр флагов немаскированных прерываний

Смещение: 20h

Сброс: 0000\_0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DC RIS 23	DC RIS 22	DC RIS 21	DC RIS 20	DC RIS 19	DC RIS 18	DC RIS 17	DC RIS 16	DC RIS 15	DC RIS 14	DC RIS 13	DC RIS 12	DC RIS 11	DC RIS 10	DC RIS 9	DC RIS 8
ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC RIS 7	DC RIS 6	DC RIS 5	DC RIS 4	DC RIS 3	DC RIS 2	DC RIS 1	DC RIS 0	SEQ RIS 7	SEQ RIS 6	SEQ RIS 5	SEQ RIS 4	SEQ RIS 3	SEQ RIS 2	SEQ RIS 1	SEQ RIS 0
ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч

Поле	Биты	Описание
DCRISd	31-8	Флаг прерывания компаратора d
		0 Нет прерывания или флаг сброшен
		1 Поступил запрос на прерывание
SEQRISs	7-0	Флаг прерывания секвенсора s
		0 Нет действий
		1 Запрос секвенсора завершился и счетчик прерываний досчитал до значения ICNT регистра SCCTL

## IM – регистр маски прерываний

Смещение: 24h

Сброс: 0000\_0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DC IM23	DC IM22	DC IM21	DC IM20	DC IM19	DC IM18	DC IM17	DC IM16	DC IM15	DC IM14	DC IM13	DC IM12	DC IM11	DC IM10	DC IM9	DC IM8
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC IM7	DC IM6	DC IM5	DC IM4	DC IM3	DC IM2	DC IM1	DC IM0	SEQ IM7	SEQ IM6	SEQ IM5	SEQ IM4	SEQ IM3	SEQ IM2	SEQ IM1	SEQ IM0
3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч	3ч

Поле	Биты	Описание
DCIMd	31-8	Маска прерывания компаратора d
		0 Маскировано
		1 Разрешено
SEQIMs	7-0	Маска прерывания секвенсора s
		0 Маскировано
		1 Разрешено

## MIS – регистр флагов маскированных прерываний

Смещение: 28h

Сброс: 0000\_0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DC MIS 23	DC MIS 22	DC MIS 21	DC MIS 20	DC MIS 19	DC MIS 18	DC MIS 17	DC MIS 16	DC MIS 15	DC MIS 14	DC MIS 13	DC MIS 12	DC MIS 11	DC MIS 10	DC MIS 9	DC MIS 8
ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC MIS 7	DC MIS 6	DC MIS 5	DC MIS 4	DC MIS 3	DC MIS 2	DC MIS 1	DC MIS 0	SEQ MIS 7	SEQ MIS 6	SEQ MIS 5	SEQ MIS 4	SEQ MIS 3	SEQ MIS 2	SEQ MIS 1	SEQ MIS 0
ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч

Поле	Биты	Описание
DCMISd	31-8	Флаг маскированного прерывания компаратора d
		0 Нет прерывания или флаг сброшен
		1 Поступил запрос на прерывание
SEQMISs	7-0	Флаг маскированного прерывания секвенсора s
		0 Нет действий
		1 Запрос секвенсора завершился и счетчик прерываний досчитал до значения ICNT регистра SCCTL

## IC – регистр сброса флагов прерываний

Смещение: 2Ch

Сброс: 0000\_0000h

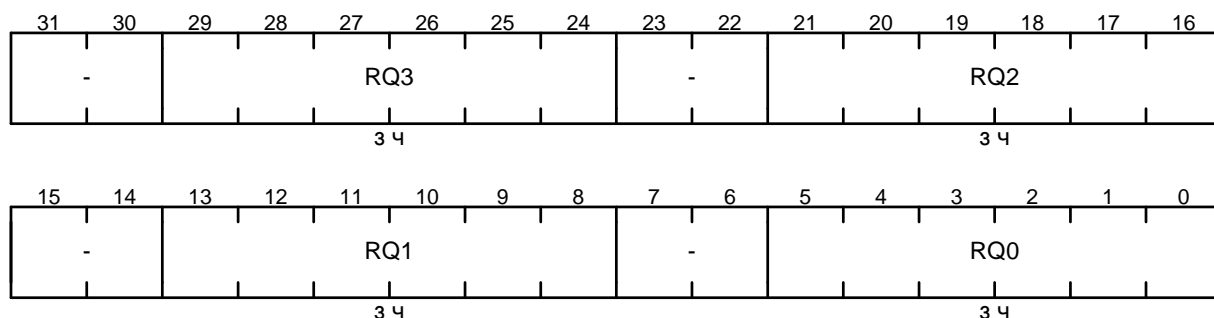
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DC IC23	DC IC22	DC IC21	DC IC20	DC IC19	DC IC18	DC IC17	DC IC16	DC IC15	DC IC14	DC IC13	DC IC12	DC IC11	DC IC10	DC IC9	DC IC8
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC IC7	DC IC6	DC IC5	DC IC4	DC IC3	DC IC2	DC IC1	DC IC0	SEQ IC7	SEQ IC6	SEQ IC5	SEQ IC4	SEQ IC3	SEQ IC2	SEQ IC1	SEQ IC0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Поле	Биты	Описание
DCICd	31-8	Сброс маскированного и немаскированного флага прерывания компаратора d
		0 Нет действий
		1 Сброс флагов
SEQICs	7-0	Сброс маскированного и немаскированного флага прерывания секвенсора s
		0 Нет действий
		1 Сброс флагов

### SRQSEL0 – регистр 0 выбора каналов для запросов секвенсора

Смещение: SEQs + 00h

Сброс: 0000\_0000h

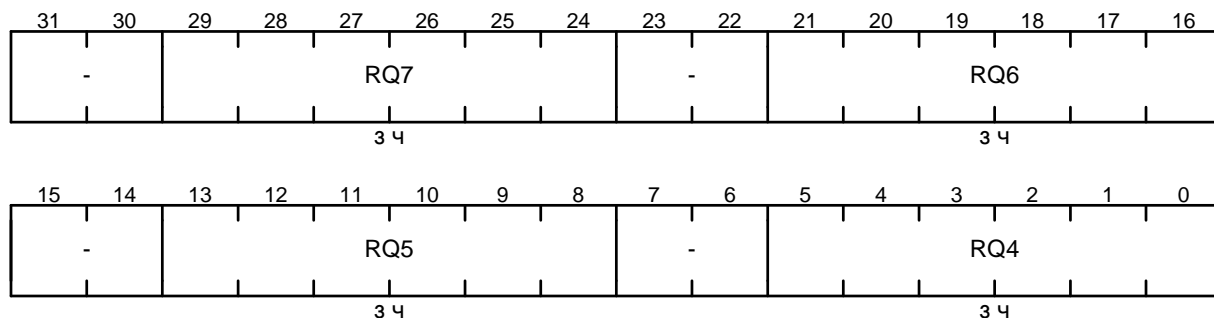


Поле	Биты	Описание
RQ <sub>n</sub>	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса (n от 0 до 3) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

### SRQSEL1 – регистр 1 выбора каналов для запросов секвенсора

Смещение: SEQs + 04h

Сброс: 0000\_0000h

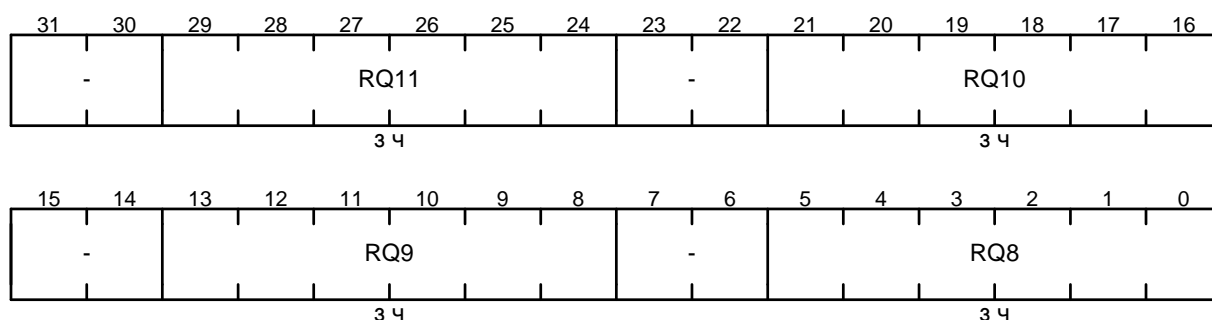


Поле	Биты	Описание
RQ <sub>i</sub>	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 4 до 7) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

## SRQSEL2 – регистр 2 выбора каналов для запросов секвенсора

Смещение: SEQs + 08h

Сброс: 0000\_0000h

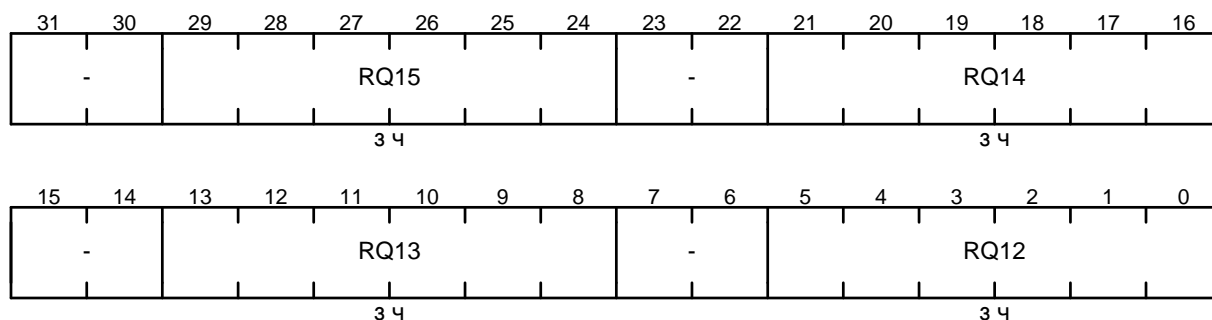


Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 8 до 11) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

## SRQSEL3 – регистр 3 выбора каналов для запросов секвенсора

Смещение: SEQs + 0Ch

Сброс: 0000\_0000h

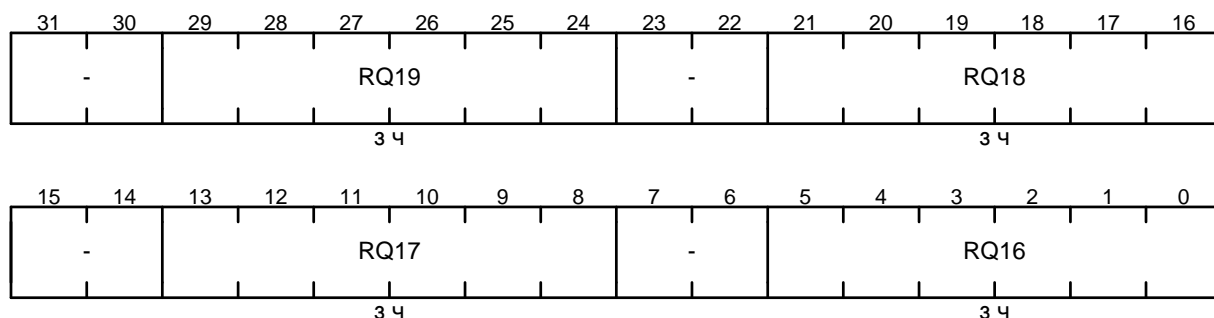


Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 12 до 15) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

### SRQSEL4 – регистр 4 выбора каналов для запросов секвенсора

Смещение: SEQs + 10h

Сброс: 0000\_0000h

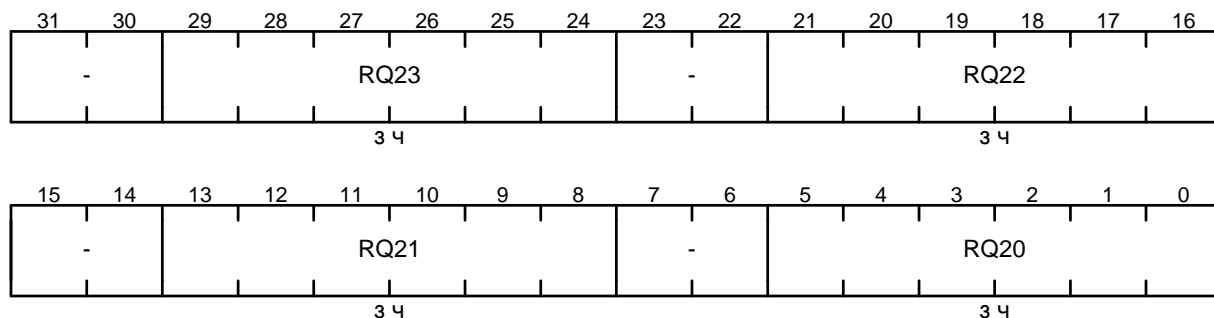


Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 16 до 19) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

### SRQSEL5 – регистр 5 выбора каналов для запросов секвенсора

Смещение: SEQs + 14h

Сброс: 0000\_0000h

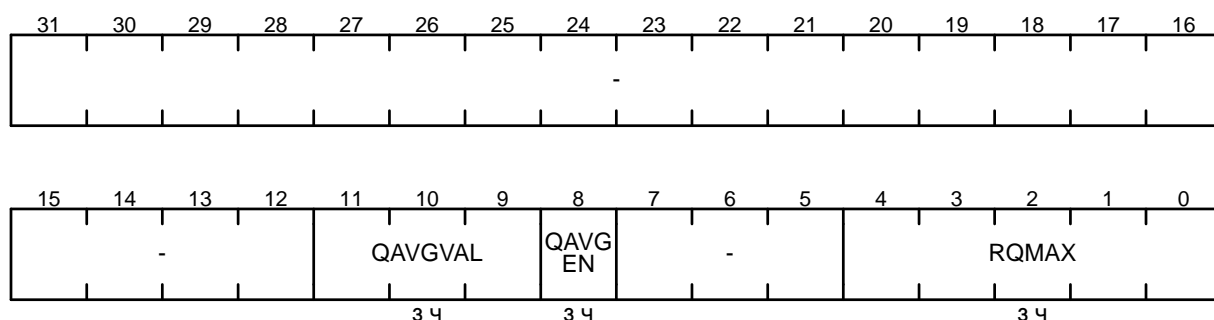


Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 20 до 23) секвенсора s. Допустимые значения 0 – 47.
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

## SRQCTL – регистр управления очередью запросов секвенсора

Смещение: SEQs + 18h

Сброс: 0000\_0000h

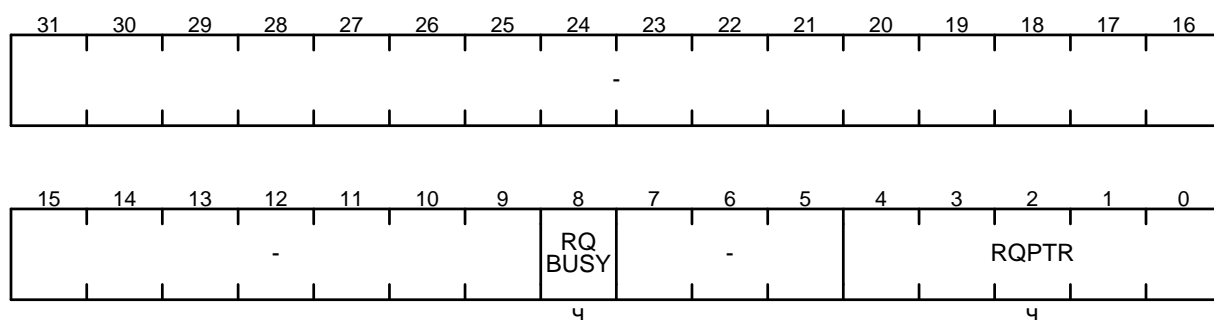


Поле	Биты	Описание
QAVGVAL	11-9	Поле задания количества опросов для усреднения сканированием
		000 Зарезервировано
		001 2
		010 4
		011 8
		100 16
		101 32
		110 64
111 Зарезервировано		
QAVGEN	8	Бит управления режимом усреднения сканированием
		0 Усреднение сканированием очереди запросов отключено
	1	Усреднение сканированием очереди запросов включено
RQMAX	4-0	Поле задания «глубины» очереди запросов секвенсора s (номер последнего элемента)
–	31-12, 7-5	Зарезервировано

## SRQSTAT – регистр статуса очереди запросов секвенсора

Смещение: SEQs + 1Ch

Сброс: 00000000h



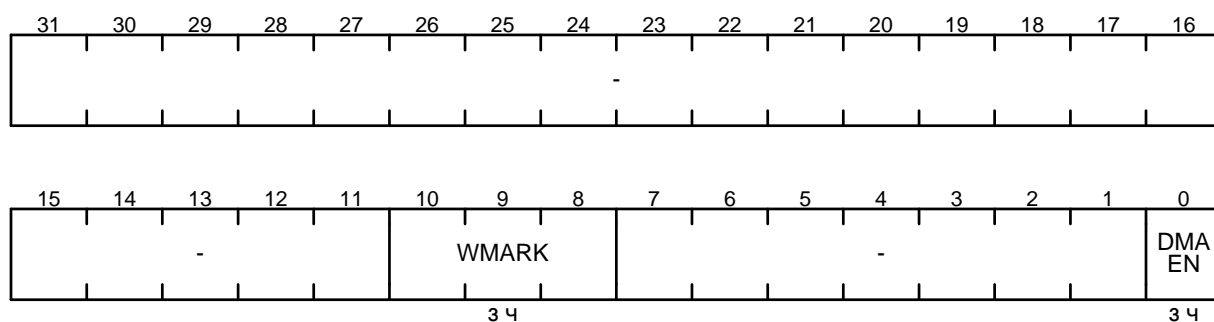
Поле	Биты	Описание
RQBUSY	8	Флаг активного запроса секвенсора
		0   Текущий запрос неактивен
		1   Запрос на измерение выставлен, и находится в состоянии обработки или ожидания
RQPTR	4-0	Номер текущего запроса в очереди
–	31-9, 7-2	Зарезервировано



## SDMACTL – регистр управления запросами DMA секвенсора

Смещение: SEQs + 20h

Сброс: 0000\_0000h

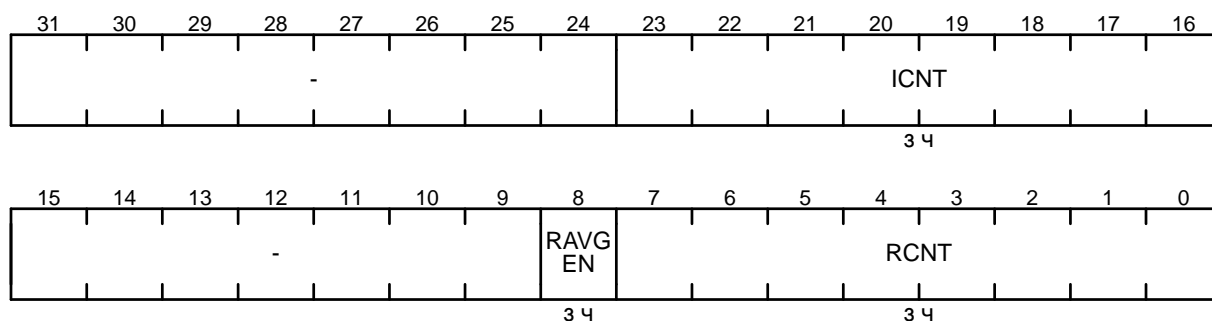


Поле	Биты	Описание
WMARK	10-8	Поле задания количества результатов измерений записанных в буфер секвенсора, по достижению которого вызывается DMA
		000   Зарезервировано
		001   Одна запись в буфер
		010   2
		011   4
		100   8
		101   16
		110   32
		111   Зарезервировано
DMAEN	0	Бит разрешения использования блока DMA
		0   Запрещено
		1   Разрешено
-	31-11, 7-2	Зарезервировано

## SCCTL – регистр управления счетчиками прерывания и перезапуска секвенсора

Смещение: SEQs + 24h

Сброс: 0000\_0000h

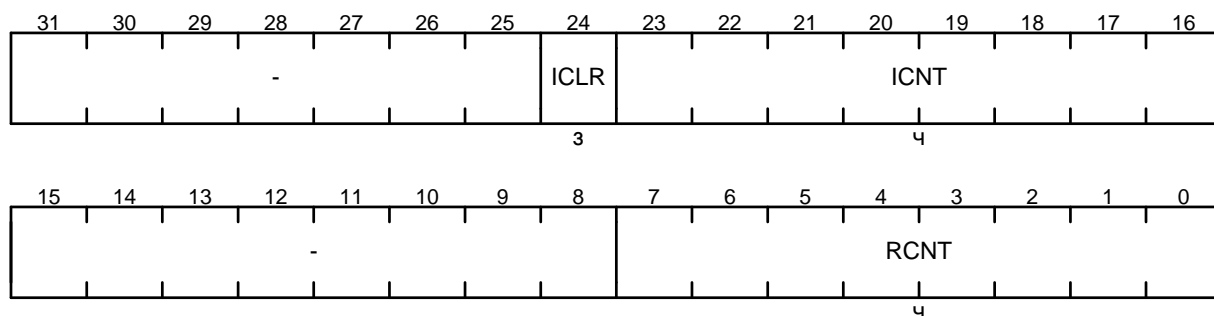


Поле	Биты	Описание
ICNT	23-16	Цикличность прерываний. Поле задания количества запросов секвенсором s модуля АЦП, по достижении которого генерируется прерывание. Значение 00h означает выставление прерывания по каждому запросу модуля АЦП, значение FFh – каждые 256 запросов.
RAVGEN	8	Бит разрешения режима усреднения результатов по перезапускам
		0      Запрещено
		1      Разрешено
		Примечание: для корректной работы этого режима поле RCNT регистра SCCTL должно содержать любое значение, соответствующее $2^p - 1$ , где $p = 1, \dots, 8$ .
RCNT	7-0	Количество перезапусков очереди запросов секвенсора s. Поле задания количества перезапусков очереди запросов секвенсора s после его запуска по событию. Значение 00h соответствует режиму без перезапусков, значение 01h – один перезапуск, FFh – 255 перезапусков
–	31-24, 15-9	Зарезервировано

### SCVAL – регистр состояния счетчиков прерывания и перезапуска секвенсора

Смещение: SEQs + 28h

Сброс: 0000\_0000h

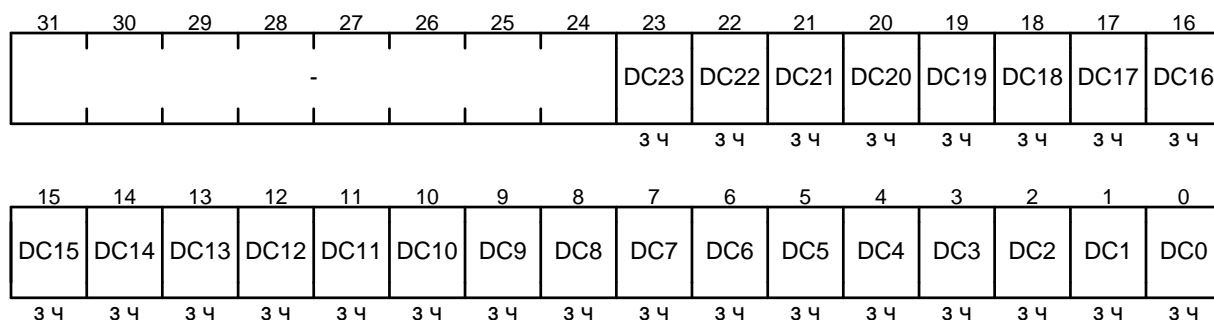


Поле	Биты	Описание	
ICLR	24	Бит сброса счетчика запросов для генерации прерывания	
		0	Нет действий
		1	Сброс счетчика ICNT
ICNT	23-16	Текущее состояние счетчика запросов, используемого для генерации прерываний	
RCNT	7-0	Текущее количество совершенных перезапусков	
–	31-25, 15-8	Зарезервировано	

### SDC – регистр выбора компаратора секвенсором

Смещение: SEQs + 2Ch

Сброс: 0000\_0000h

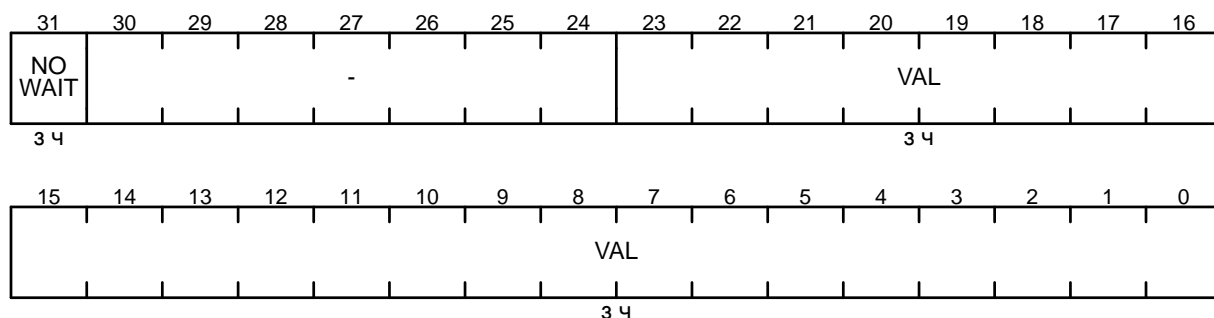


Поле	Биты	Описание	
DCd	23-0	Бит разрешения работы компаратора d секвенсором	
		0	Запрещен
		1	Разрешен
–	31-24	Зарезервировано	

### SRTMR – регистр задержки перезапусков секвенсора

Смещение: SEQs + 30h

Сброс: 0000\_0000h

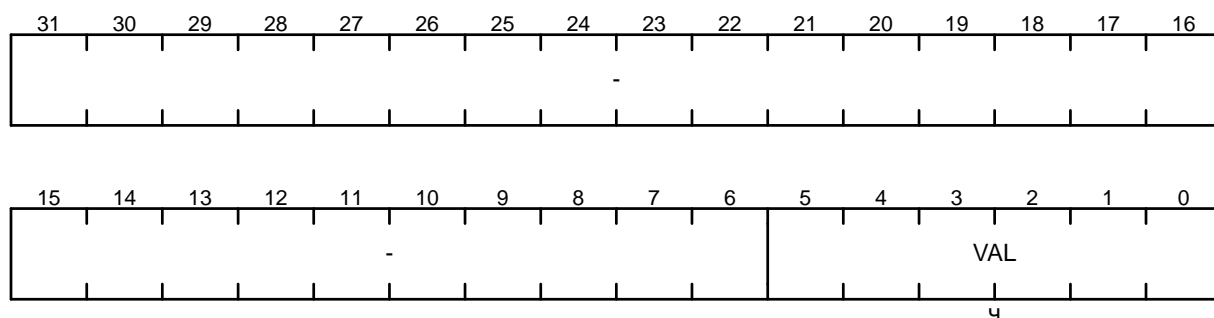


Поле	Биты	Описание	
NOWAIT	31	Бит управления теневой загрузкой значения задержки перезапуска	
		0	Значение обновится по ближайшему событию запуска секвенсора
		1	Значение обновится по ближайшему событию перезапуска
VAL	23-0	Поле задания задержки перезапуска очереди секвенсора. Значение TMR = 000000h задает немедленный перезапуск (если он включен)	
–	30-24	Зарезервировано	

### SFLOAD – регистр загрузки буфера секвенсора

Смещение: SEQs + 34h

Сброс: 0000\_0000h

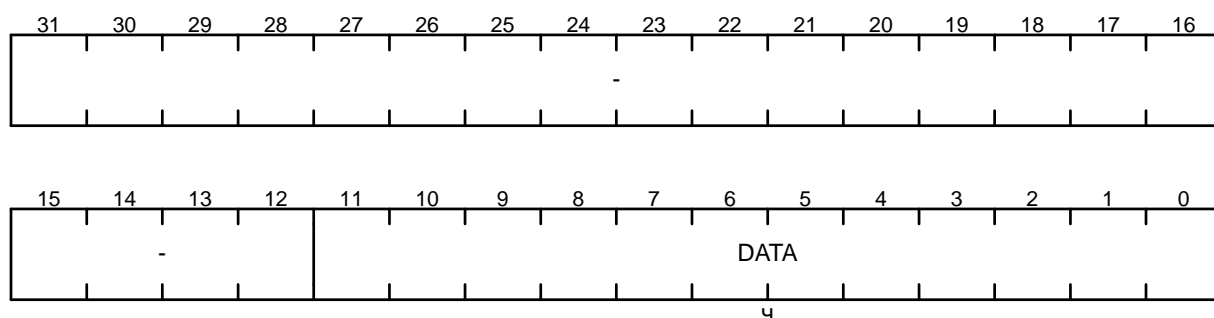


Поле	Биты	Описание
VAL	5-0	Значение количества результатов измерений, сохраненных в буфере секвенсора
–	31-6	Зарезервировано

## SFIFO – регистр результата измерения секвенсора

Смещение: SEQs + 38h

Сброс: 0000\_0000h

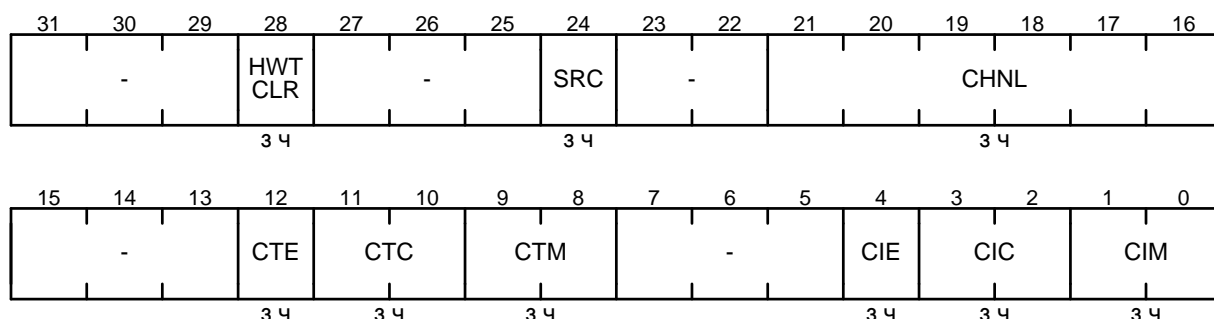


Поле	Биты	Описание
DATA	11-0	Результат измерения. Чтение поля DATA возвращает результат измерения из буфера секвенсора
-	31-12	Зарезервировано

## DCTL – регистр управления компаратора

Смещение: DCd + 00h

Сброс: 0000\_0000h



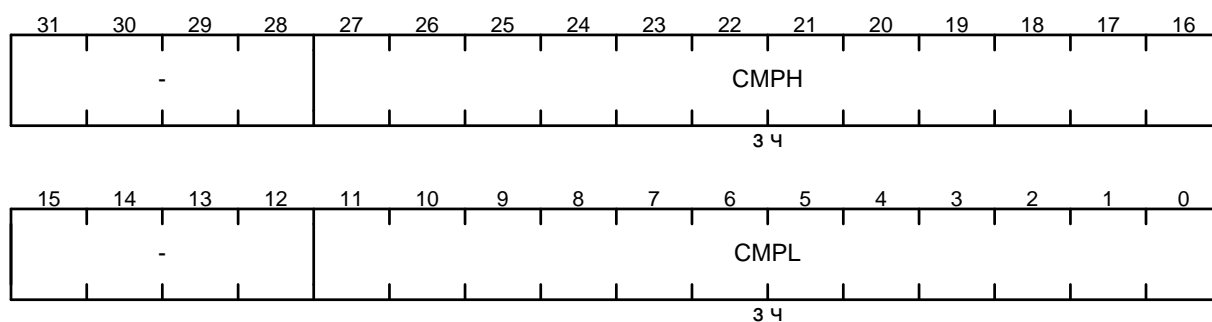
Поле	Биты	Описание
HWTCLR	28	Управление аппаратным сбросом выходного триггера компаратора
		0   Сброс происходит через флаги DCTRIG
SRC	24	1   Сброс происходит аппаратно при первом отрицательном сравнении
		Выбор источника значения для сравнения
CHNL	21-16	0   Окончание измерения АЦП
		1   Запись результата в FIFO секвенсором
CTE	12	Номер канала. Поле выбирает канал, результат измерения которого будет передан на компаратор
		Бит разрешения срабатывания выходного триггера компаратора
		0   Запрещено
		1   Разрешено

Поле	Биты	Описание
СТС	11-10	Поле задания условия срабатывания выходного триггера. Если для значения, полученного в результате измерения, выполняется условие, то состояние триггера единица, в противном случае – ноль
		00   Измерение $\leq$ СМРL
		01   СМРL $\leq$ Измерение $\leq$ СМРH
		10   СМРH $\leq$ Измерение
		11   Зарезервировано
		Параметры СМРL и СМРH задаются в регистре DСМР
СТМ	9-8	Поле задания режима срабатывания выходного триггера
		00   Многократный
		01   Однократный
		10   Многократный с гистерезисом
		11   Однократный с гистерезисом
СIE	4	Бит разрешения прерывания компаратора
		0   Запрещено
		1   Разрешено. Прерывание генерируется каждый раз при одновременном выполнении условий СIС и СIМ
СIС	3-2	Поле задания условия генерирования прерывания
		00   Измерение $\leq$ СМРL
		01   СМРL $\leq$ Измерение $\leq$ СМРH
		10   СМРH $\leq$ Измерение
		11   Зарезервировано
СIМ	1-0	Поле задания режима генерирования прерывания
		00   Многократный
		01   Однократный
		10   Многократный с гистерезисом
		11   Однократный с гистерезисом
–	31-25, 23-22, 15-13, 7-5	Зарезервировано

### DCMP – регистр диапазона компаратора

Смещение: DCd + 04h

Сброс: 0000\_0000h

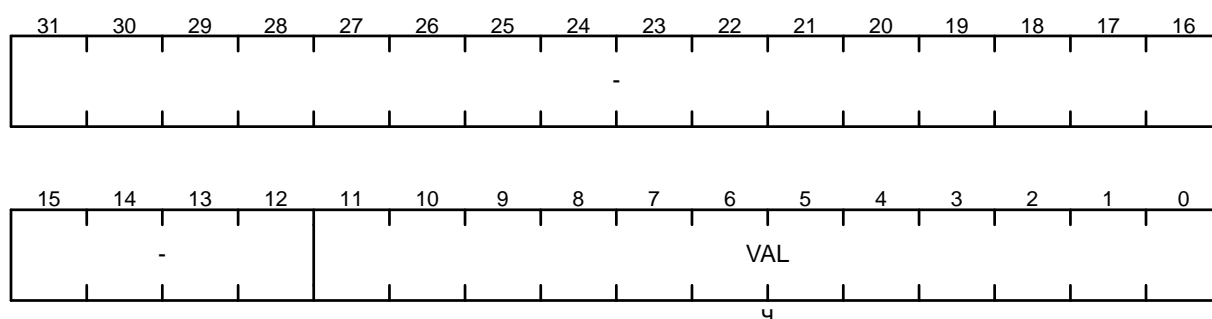


Поле	Биты	Описание
CMPH	27-16,	Поле значения верхнего порога диапазона измерений.
		Всегда должно выполняться условие $CMPL \leq CMPH$
CMPL	11-0	Поле значения нижнего порога диапазона измерений
–	31-28, 15-12	Зарезервировано

### DDATA – регистр результата измерения компаратора

Смещение: DCd + 08h

Сброс: 0000\_0000h

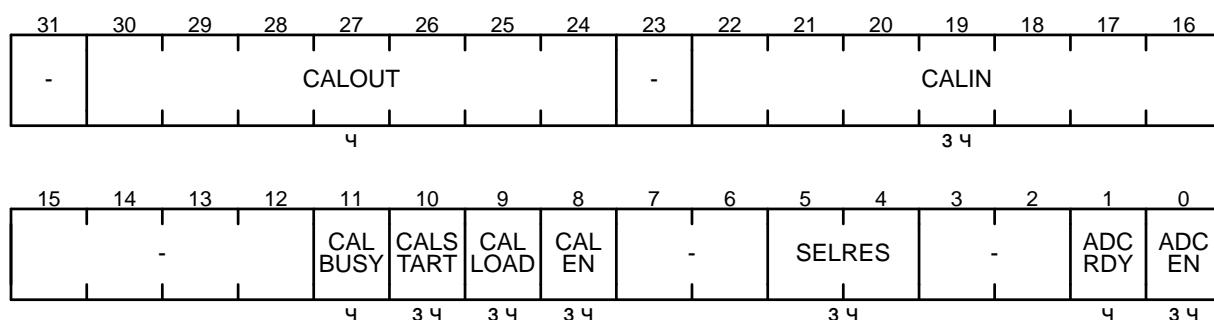


Поле	Биты	Описание
VAL	11-0	Значение результата измерения, которое последним использовалось компаратором при проверке на соответствие условиям СТС и СТМ (см. регистр DCTL)
–	31-12	Зарезервировано

## ACTL – массив регистров управления модулями АЦП

Смещение: ACTL + (4\*m)h

Сброс: 0000\_0000h



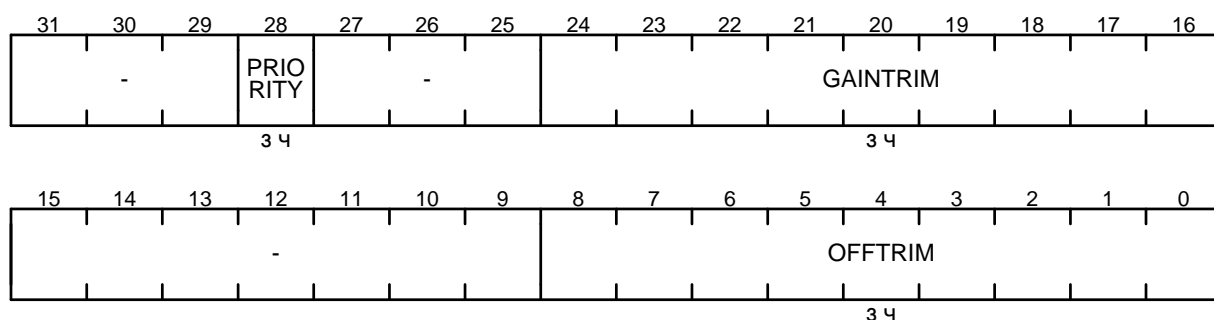
Поле	Биты	Описание
CALOUT	30-24	Поправочное значение, полученное в результате калибровки. Диапазон значений –64, ..., 63, величина в дополнительном коде.
CALIN	22-16	Поправочное значение для ручного занесения в схему калибровки с помощью бита CALLOAD. Диапазон значений –64, ..., 63, величина вносится в дополнительный код.
CALBUSY	11	Флаг активности процедуры калибровки. Сбрасывается автоматически.
		0   Калибровка не проводится 1   Проводится калибровка
CALSTART	10	Старт проведения схемы внутренней калибровки АЦП
		0   Нет реакции 1   Старт калибровки
CALLOAD	9	Внесение значения поля CALIN во внутреннюю схему калибровки
		0   Нет реакции 1   Загрузка значения
CALEN	8	Включение схемы внутренней калибровки
		0   Выключена 1   Включена
SELRES	5-4	Выбор разрядности результата модуля АЦП
		00b   6 бит
		01b   8 бит
		10b   10 бит 11b   12 бит
ADCRDY	1	Флаг готовности АЦП к проведению измерений.
		0   АЦП выключено (ADCEN=0), либо в состоянии инициализации 1   АЦП включено и готово к преобразованиям
ADCEN	0	Включение АЦП. При каждом включении запускается процедура инициализации, которая завершается установкой ADCRDY.
		0   Модуль АЦП выключен 1   Модуль АЦП включен
–	31,23, 15-12, 7-6, 3-2	Зарезервировано



## CHCTL – массив регистров настройки каналов

Смещение: CHCTL + (4\*n)h

Сброс: 0000\_0000h



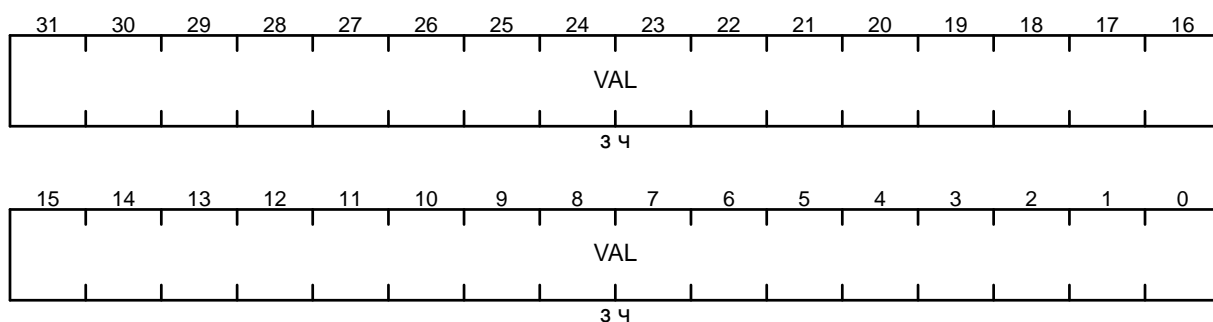
Поле	Биты	Описание
PRIORITY	28	Бит задания приоритета канала
		0   Канал имеет стандартный приоритет
		1   Канал имеет повышенный приоритет
GAINTRIM	24-16	Поле задания коэффициента для коррекции усиления. Диапазон значений –256, ..., 255 в 12-разрядном режиме, величина вносится в дополнительный код: 100h соответствует –256, 000h – 0, 0FFh – 255. При других разрядностях результата диапазон сужается - подробнее в таблице 18.2.
OFFTRIM	8-0	Поле задания коэффициента для коррекции смещения нуля. Диапазон значений –256, ..., 255 в 12-разрядном режиме, величина вносится в дополнительный код: 100h соответствует –256, 000h – 0, 0FFh – 255. При других разрядностях результата диапазон сужается - подробнее в таблице 18.2.
–	31-29, 27-25, 15-9	Зарезервировано

## A.16 Регистры блоков захвата

<b>Базовый адрес:</b>	4009_A000h	Регистры блока захвата 0
	4009_B000h	Регистры блока захвата 1
	4009_C000h	Регистры блока захвата 2
	4009_D000h	Регистры блока захвата 3
	4009_E000h	Регистры блока захвата 4
	4009_F000h	Регистры блока захвата 5

### TSCTR – регистр счетчика таймера

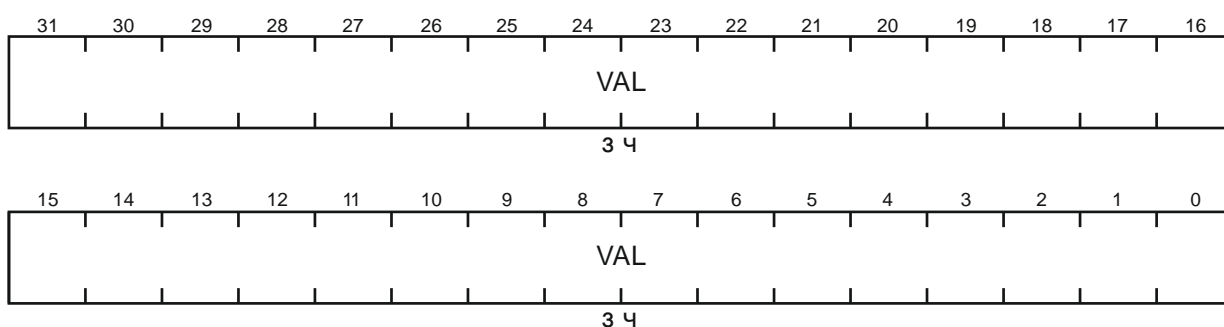
Смещение: + 00h  
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Запись задает начальное значение таймера. Чтение возвращает текущее значение таймера.

### STRPHS – регистр отложенной загрузки таймера

Смещение: + 04h  
Сброс: 0h

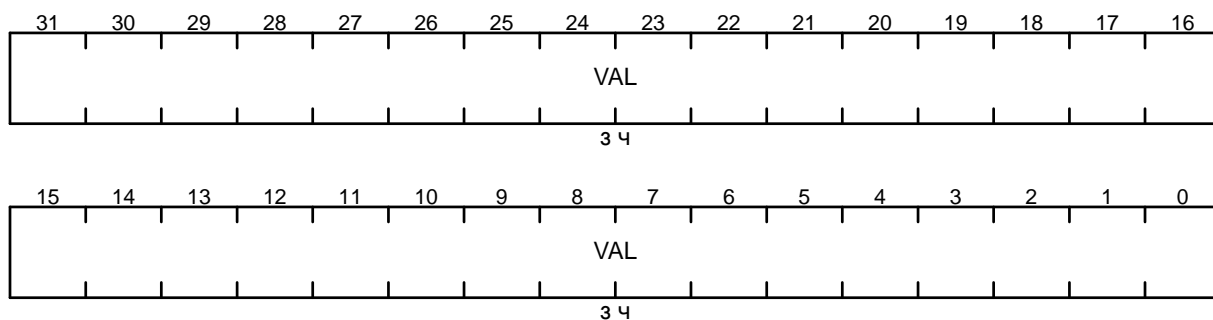


Поле	Биты	Описание
VAL	31-0	Значение из регистра загружается в таймер по событиям SYNCI или под управлением процессора. Регистр используется для синхронизации с другими блоками CAP/PWM.

### САР0 – регистр захвата 0

Смещение: + 08h

Сброс: 0h

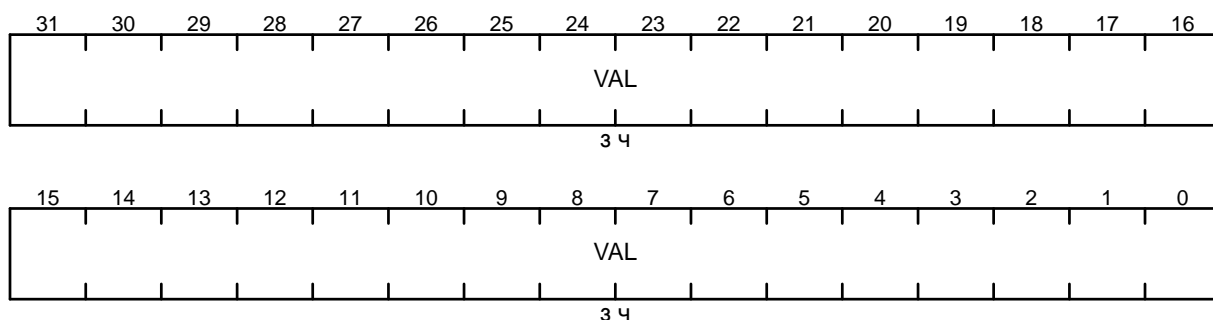


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме захвата и содержит значение таймера, сохраненное по событию EV0

### PRD – регистр периода

Смещение: + 08h

Сброс: 0h

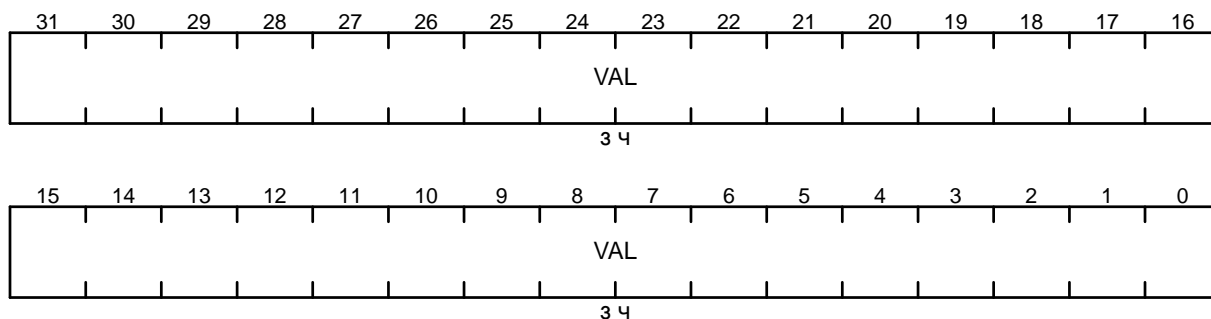


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме APWM и содержит значение периода генерации. Величина может быть обновлена из регистра отложенной загрузки PRDSHDW

### САР1 – регистр захвата 1

Смещение: + 0Ch

Сброс: 0h

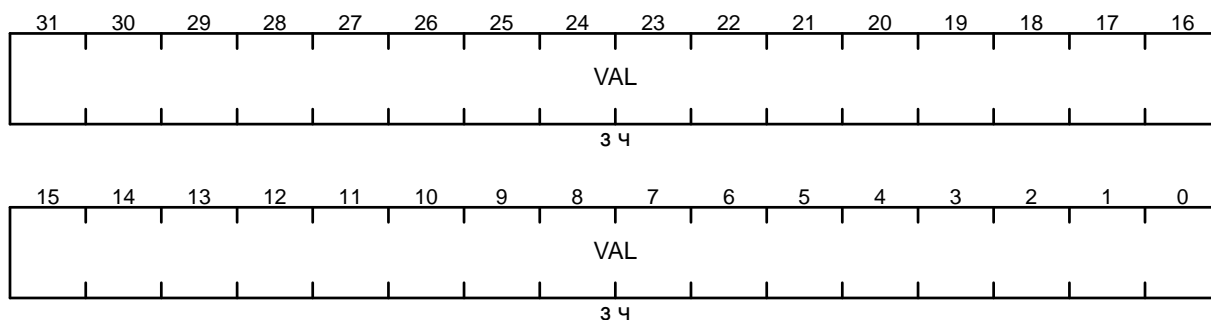


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме захвата и содержит значение таймера, сохраненное по событию EV1

### СМР – регистр сравнения

Смещение: + 0Ch

Сброс: 0h

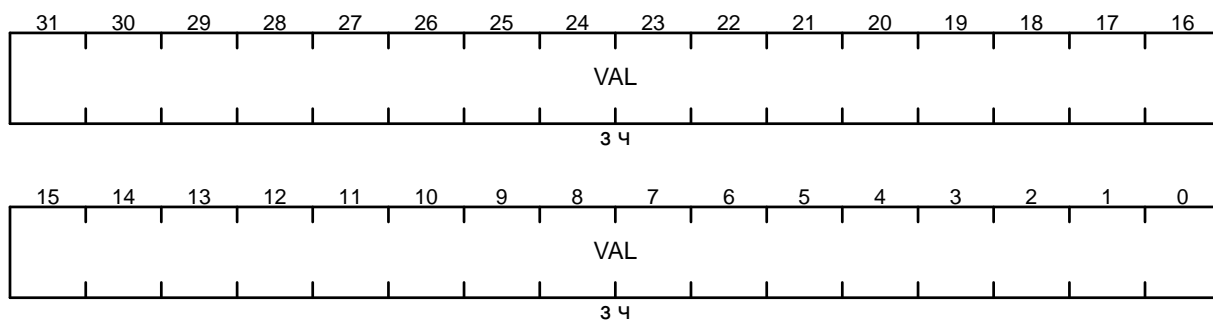


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме APWM и содержит значение сравнения. Величина может быть обновлена из регистра отложенной загрузки CMPSHDW

## САР2 – регистр захвата 2

Смещение: + 10h

Сброс: 0h

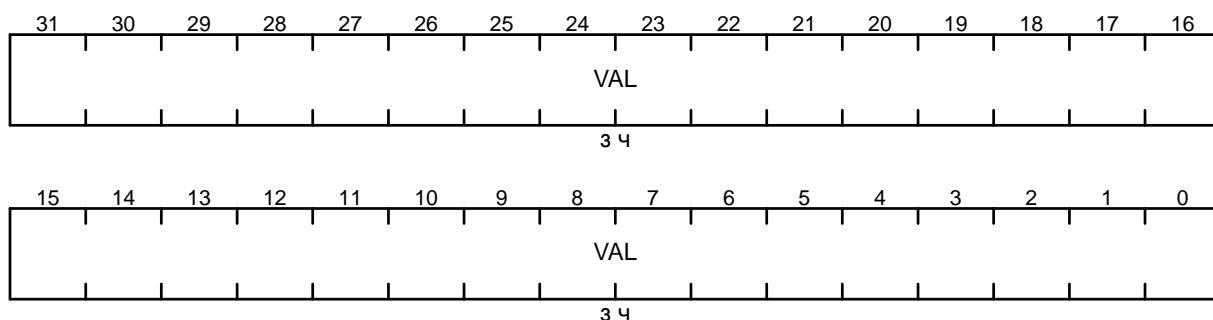


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме захвата и содержит значение таймера, сохраненное по событию EV2

## PRDSHDW – регистр отложенной загрузки периода

Смещение: + 10h

Сброс: 0h

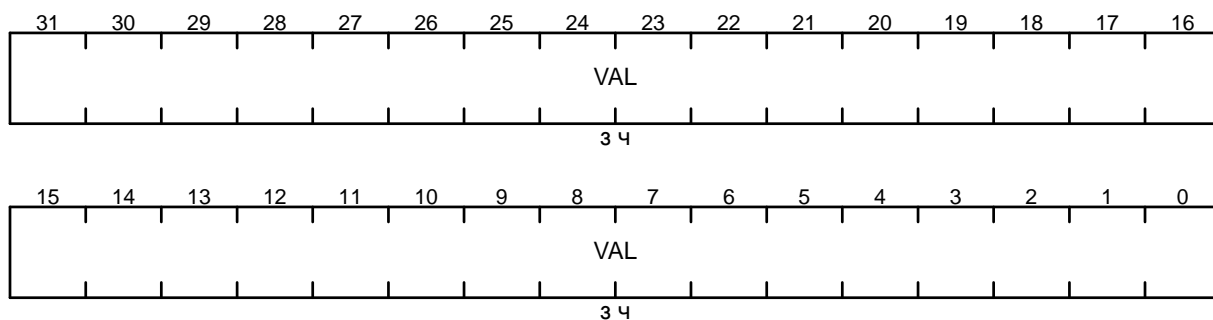


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме APWM и содержит значение отложенной загрузки периода. По событию CTR = PRD значение будет перенесено в регистр PRD

### САРЗ – регистр захвата 3

Смещение: + 14h

Сброс: 0h

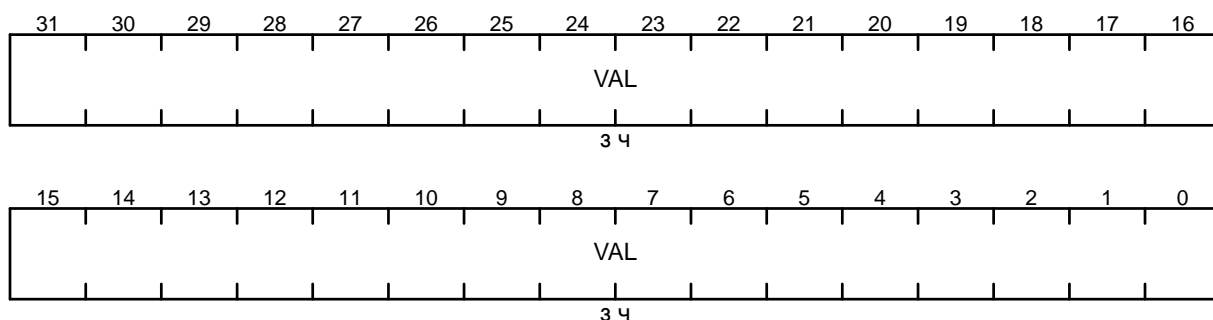


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме захвата и содержит значение таймера, сохраненное по событию EV3

### СМPSHDW – регистр отложенной загрузки сравнения

Смещение: + 14h

Сброс: 0h

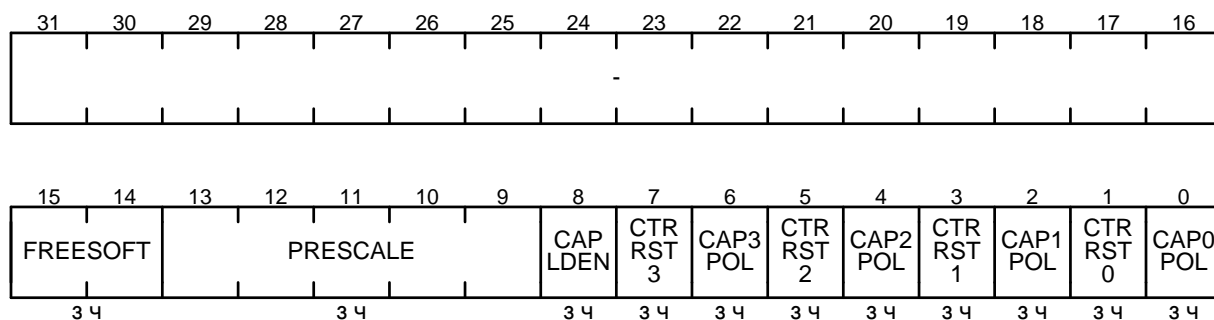


Поле	Биты	Описание
VAL	31-0	Регистр используется только в режиме APWM и содержит значение отложенной загрузки сравнения. По событию CTR = PRD значение будет перенесено в регистр CMP.

## ЕСCTL0 – регистр контроля 0

Смещение: + 28h

Сброс: 0h

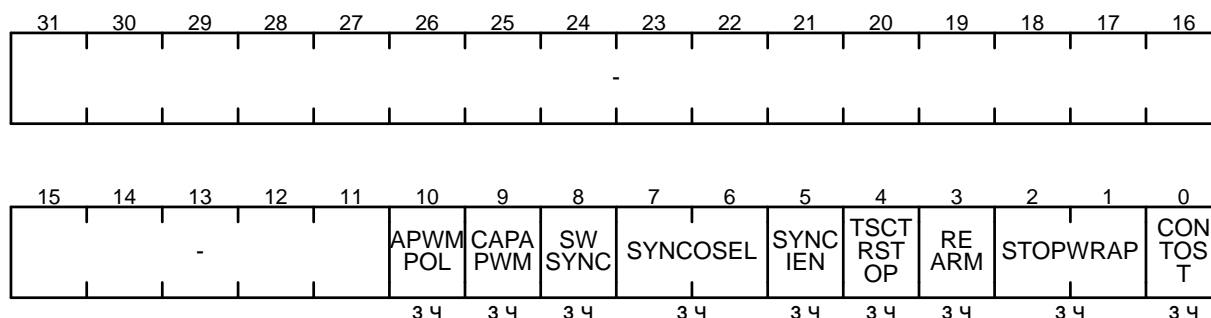


Поле	Биты	Описание	
FREESOFT	15-14	Управление остановкой таймера в режиме отладки	
		00b	Моментальная остановка таймера
		01b	Остановка таймера по переполнению
		10b, 11b	Таймер не останавливается
PRESCALE	13-9	Предварительный делитель. Если записано значение 00h – делитель выключен.	
CAPLDEN	8	Бит разрешения захвата регистрами CAP0 – CAP3	
		0	Запрещено
CTRRSTn	7, 5, 3, 1	Бит сброса таймера после события n (n от 0 до 3)	
		0	Нет действий
CAPnPOL	6, 4, 2, 0	Бит выбора фронта захвата события n (n от 0 до 3)	
		0	Захват по переднему фронту
		1	Захват по заднему фронту
		–	31-16

## ЕСCTL1 – регистр контроля 1

Смещение: + 2Ch

Сброс: 0h



Поле	Биты	Описание	
APWMPOL	10	Бит задания активного уровня в режиме APWM	
		0	Высокий
		1	Низкий
CAPAPWM	9	Бит выбора режима	
		0	Работа в режиме захвата. Блокирование сброса таймера по событию CTR=PRD. Блокирование отложенной загрузки PRD, CMP. Разрешение захвата CAP0–CAP3. Внешний порт работает на вход.
		1	Работа в режиме APWM. Разрешение сброса таймера по CTR = PRD. Разрешение теневой загрузки PRD, CMP. Блокирование захвата CAP0–CAP3. Внешний порт работает на выход.
SWSYNC	8	Межблочная синхронизация таймеров	
		0	Нет действий
		1	Запись единицы: - загружает значение таймера из отложенного регистра при условии, что установлен бит SYNCIEN; - генерирует выходной сигнал синхронизации SYNCO при условии, что в поле SYNCOSSEL записано 00b.
		Примечание – В режиме APWM синхронизация также происходит автоматически по событию CTR = PRD.	
SYNCOSSEL	7-6	Выбор источника выходного синхросигнала SYNCO	
		00b	Пропуск сигнала синхронизации с SYNCI или SWSYNC
		01b	Передача события CTR = PRD в качестве выходного сигнала синхронизации
		10b, 11b	Запрет выходного сигнала синхронизации
SYNCIEN	5	Бит разрешения синхронизации	
		0	Запрещено
		1	Разрешено
TSCTRSTOP	4	Бит управления работой таймера	
		0	Остановлен
		1	Запущен
REARM	3	Запись единицы запускает следующую последовательность действий: сброс управляющего контроллера, разрешение работы управляющего контроллера и загрузку регистров захвата	

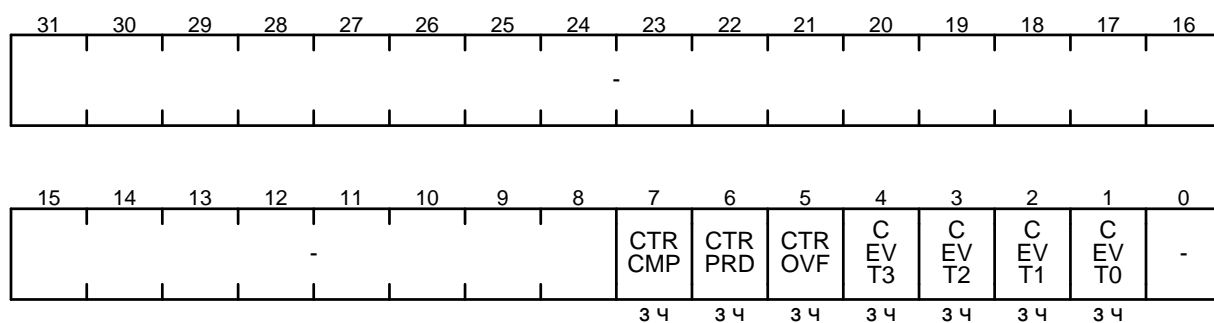


Поле	Биты	Описание
STOPWRAP	2-1	Значение компаратора остановки в режимах захвата
		00   Останов при значении счетчика 00
		01   Останов при значении счетчика 01
		10   Останов при значении счетчика 10
		11   Останов при значении счетчика 11
		Примечание – Остановка управляющего контроллера приводит также к блокировке загрузки регистров захвата.
CONTOST	0	Режим работы захвата
		0   Циклический
		1   Однократный
–	31-16	

### ЕСЕINT – регистр маски прерываний

Смещение: + 30h

Сброс: 0h



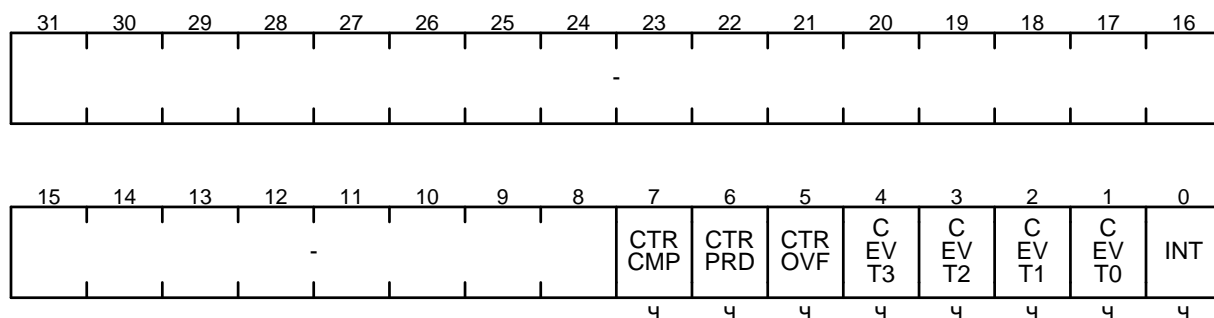
Поле	Биты	Описание
CTRCMP	7	Бит разрешения генерации прерывания по событию CTR = CMP
CTRPRD	6	Бит разрешения генерации прерывания по событию CTR = PRD
CTROVF	5	Бит разрешения генерации прерывания по событию CTROVF
CEVTn	4-1	Бит разрешения генерации прерывания по событию CEVTn (n от 0 до 3)
–	31-8, 0	Зарезервировано

Примечание – Установленный бит разрешает прерывание, сброшенный – запрещает

## ECFLG –регистр статуса прерываний

Смещение: + 34h

Сброс: 0h



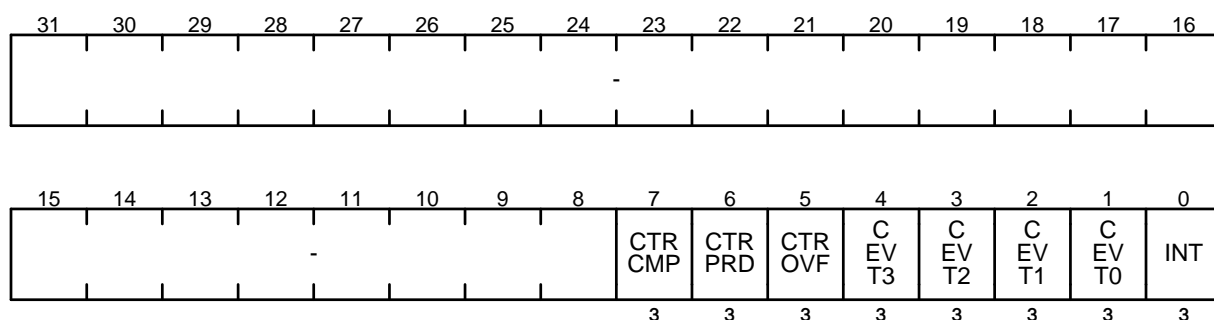
Поле	Биты	Описание	
CTRCMP	7	Флаг прерывания по событию CTR = CMP	
		0	Событие не произошло
		1	Событие произошло
CTRPRD	6	Флаг прерывания по событию CTR = PRD	
		0	Событие не произошло
		1	Событие произошло
CTROVF	5	Флаг прерывания по событию CTROVF	
		0	Событие не произошло
		1	Событие произошло
CEVTn	4-1	Флаг прерывания по событию CEVTn (n от 0 до 3)	
		0	Событие не произошло
		1	Событие произошло
INT	0	Флаг прерывания	
-	31-8	Зарезервировано	

Примечание – Все флаги сбрасываются записью единиц в биты регистра ECCLR.

## ECCLR – регистр сброса прерываний

Смещение: + 38h

Сброс: 0h

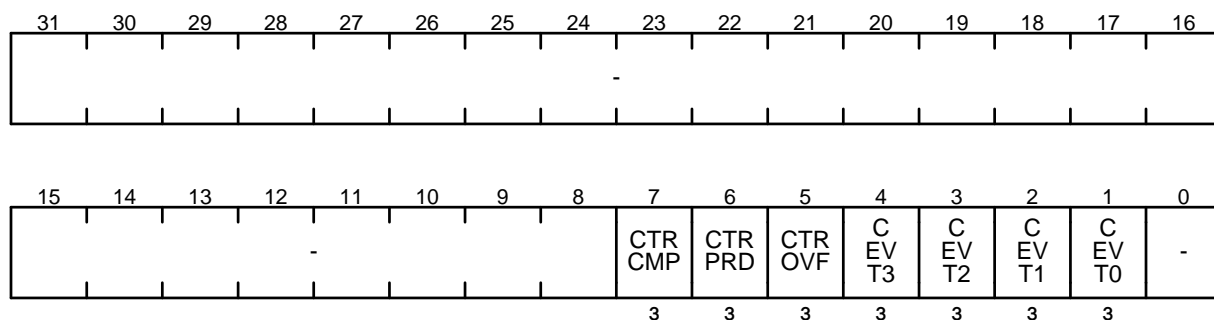


Поле	Биты	Описание
CTRCMP	7	Запись единицы сбрасывает флаг прерывания по событию CTR = CMP
CTRPRD	6	Запись единицы сбрасывает флаг прерывания по событию CTR = PRD
CTROVF	5	Запись единицы сбрасывает флаг прерывания по событию CTROVF
CEVTn	4-1	Запись единицы сбрасывает флаг прерывания по событию CEVTn (n от 0 до 3)
INT	0	Запись единицы сбрасывает флаг прерывания
–	31-8	Зарезервировано

## ECFRC – регистр программных прерываний

Смещение: + 3Ch

Сброс: 0h

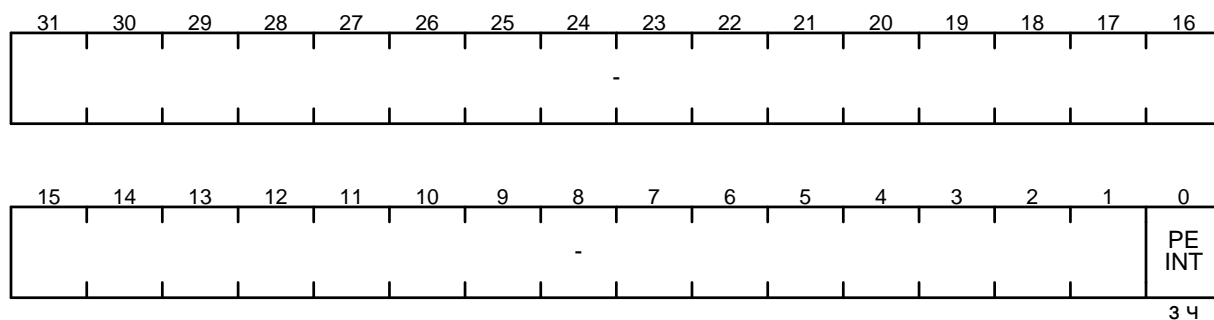


Поле	Биты	Описание
CTRCMP	7	Запись единицы генерирует прерывание по событию CTR = CMP
CTRPRD	6	Запись единицы генерирует прерывание по событию CTR = PRD
CTROVF	5	Запись единицы генерирует прерывание по событию CTROVF
CEVTn	4-1	Запись единицы генерирует прерывание по событию CEVTn
–	31-8, 0	Зарезервировано

## PEINT – регистр активного прерывания

Смещение: + 40h

Сброс: 0h



Поле	Биты	Описание
PEINT	0	Флаг активного прерывания. Устанавливается при возникновении прерывания блока. Сбрасывается только программно, записью единицы
–	31-1	Зарезервировано

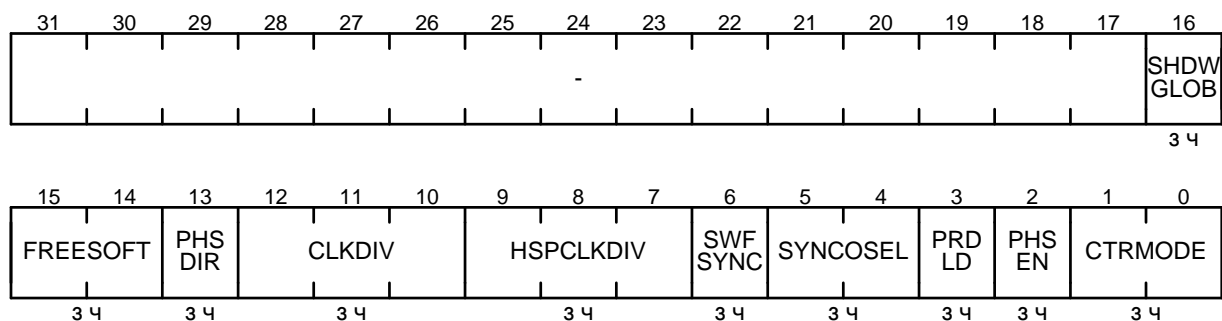
## А.17 Регистры блоков ШИМ

<b>Базовый адрес:</b>	400A_0000h	Регистры блока ШИМ 0
	400A_1000h	Регистры блока ШИМ 1
	400A_2000h	Регистры блока ШИМ 2
	400A_3000h	Регистры блока ШИМ 3
	400A_4000h	Регистры блока ШИМ 4
	400A_5000h	Регистры блока ШИМ 5
	400A_6000h	Регистры блока ШИМ 6
	400A_7000h	Регистры блока ШИМ 7
	400A_8000h	Регистры блока ШИМ 8
	400A_9000h	Регистры блока ШИМ 9

### ТВCTL – регистр управления таймером

Смещение: + 00h

Сброс: 0001\_8000h



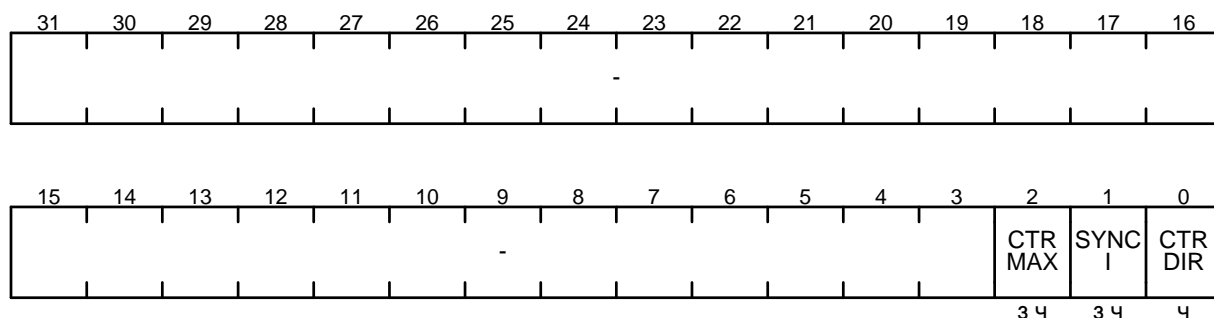
Поле	Биты	Описание	
SHDWGLOB	16	Глобальное разрешение всех теневых загрузок	
		0	Все теневые регистры пишутся, но события перезаписи в активные блокируются
		1	Теневая загрузка происходит согласно настройкам соответствующих полей
FREESOFT	15-14	Поле задания поведения счетчика ШИМ после перехода в режим останова во время отладки	
		00	Счетчик будет остановлен на следующий такт ТВCLK
		01	Счетчик будет остановлен по достижении события: - ТВCTR = ТВPRD (при счете вверх); - ТВCTR = 0000h (при счете вниз или вверх-вниз).
		1x	Счетчик продолжит работу
PHSDIR	13	Бит задания фазового направления (используется только при двунаправленном счете). Задаёт направление счета после синхронизации. Загружается вместе с регистром фазы ТВPHS	
		0	Вниз
		1	Вверх

Поле	Биты	Описание	
CLKDIV	12-10	Поле задания первого делителя тактовой частоты	
		000	1
		001	1/2
		010	1/4
		011	1/8
		100	1/16
		101	1/32
		110	1/64
HSPCLKDIV	9-7	Поле задания второго делителя тактовой частоты. Конечное значение делителя является произведением значений делителей, задаваемых полями CLKDIV и HSPCLKDIV	
		000	1
		001	1/2
		010	1/4
		011	1/6
		100	1/8
		101	1/10
		110	1/12
SWFSYNC	6	Бит программной эмуляции появления синхроимпульса	
		0	Нет действий
		1	Запись единицы вызывает появление синхроимпульса в цепи PWM_SYNCI
SYNCOSEL	5-4	Поле выбора источника для выходного сигнала синхронизации PWM_SYNCO	
		00	PWM_SYNCI
		01	CTR = 0000h
		10	CTR = CMPB
PRDL	3	Бит управления загрузкой регистра TBPRD	
		0	Режим отложенной загрузки регистра TBPRD разрешен
		1	Запись в TBPRD будет произведена сразу в активный регистр
PHSEN	2	Бит разрешения загрузки счетчика таймера	
		0	Запрещено
		1	Разрешена загрузка счетчика TBCTR значением регистра фазы TBPHS при получении события синхронизации (импульс на входе PWM_SYNCI или запись в бит SWFSYNC)
CTRMODE	1-0	Поле задания направления счета	
		00	Вверх
		01	Вниз
		10	Вверх-вниз
-	31-17	Зарезервировано	

## TBSTS – регистр статуса таймера

Смещение: + 04h

Сброс: 0h

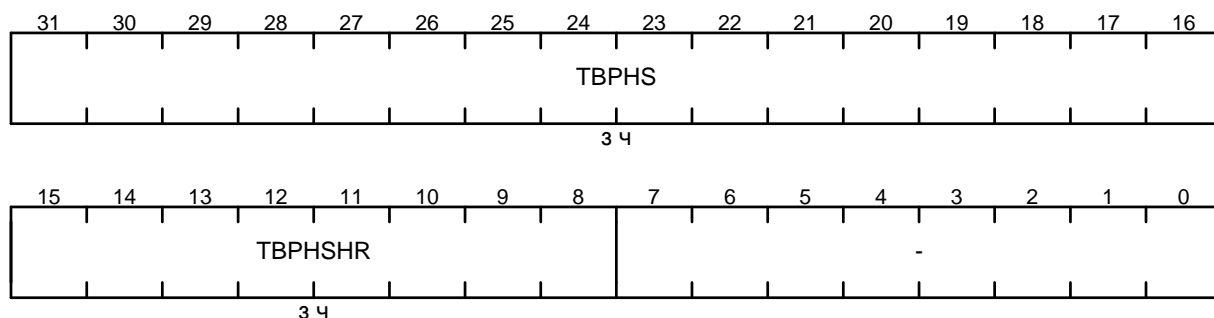


Поле	Бит	Описание
CTRMAX	2	Флаг достижения счетчиком таймера своего максимального значения FFFFh
		0   Значение не достигнуто или флаг был сброшен
		1   Значение было достигнуто
		Запись единицы сбрасывает флаг
SYNCI	1	Флаг синхронизации
		0   Синхронизация не достигнута или флаг был сброшен
		1   Синхронизация произошла
		Запись единицы сбрасывает флаг
CTRDIR	0	Флаг направление счета таймера
		0   Вниз
		1   Вверх
–	31-3	Зарезервировано

## ТВРНС – регистр фазы таймера

Смещение: + 08h

Сброс: 0h

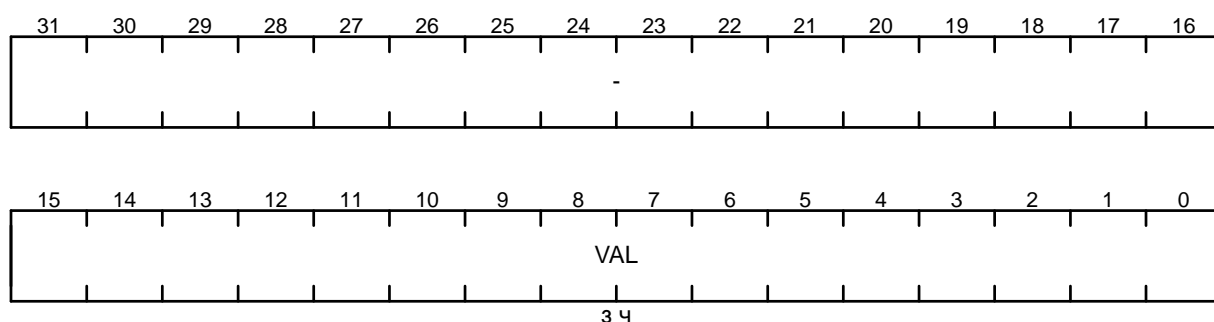


Поле	Биты	Описание
ТВРНС	31-16	Поле задания начальной фазы таймера при получении сигнала синхронизации
ТВРНСНР	15-8	Поле дополнительных разрядов начальной фазы таймера. Доступен в блоке ШИМ высокого разрешения. Если бит РНSEN сброшен, то синхронизация отключена, и таймер не будет загружаться значением ТВРНС. Если бит РНSEN установлен, то по получению события синхронизации в счетчик таймера ТВСТР будет загружено значение ТВРНС
–	7-0	Зарезервировано

## ТВСТР – регистр текущего значения таймера

Смещение: + 0Ch

Сброс: 0h



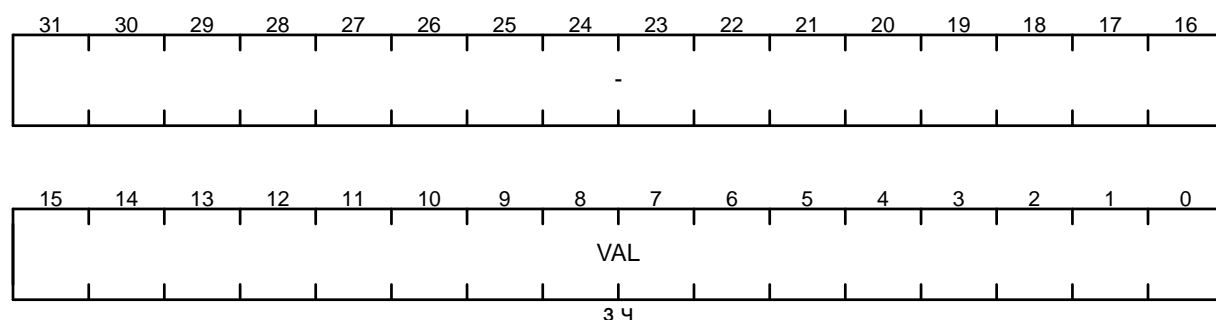
Поле	Биты	Описание
VAL	15-0	Текущее значение счетчика таймера. Запись в регистр изменяет значение таймера. Запись происходит асинхронно с TBLK и не использует отложенный механизм загрузки
–	31-16	Зарезервировано



## ТВPRD – регистр периода таймера

Смещение: + 10h

Сброс: 0h

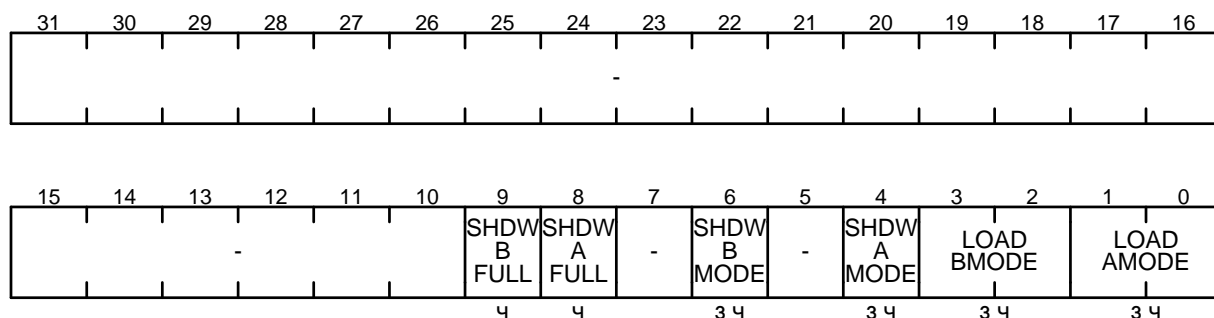


Поле	Биты	Описание
VAL	15-0	Период таймера (максимальное значение счета таймера). Отложенная загрузка в этот регистр программируется битом PRDLД регистра ТВCTL. По умолчанию бит PRDLД сброшен и запись в регистр ТВPRD приводит к записи в теневой регистр. Активный регистр будет загружен по событию ТВCTR = Zero. Если бит PRDLД установлен, то запись выполняется напрямую в активный регистр
–	31-16	Зарезервировано

## CMPCTL – регистр управления компаратором

Смещение: + 14h

Сброс: 0h

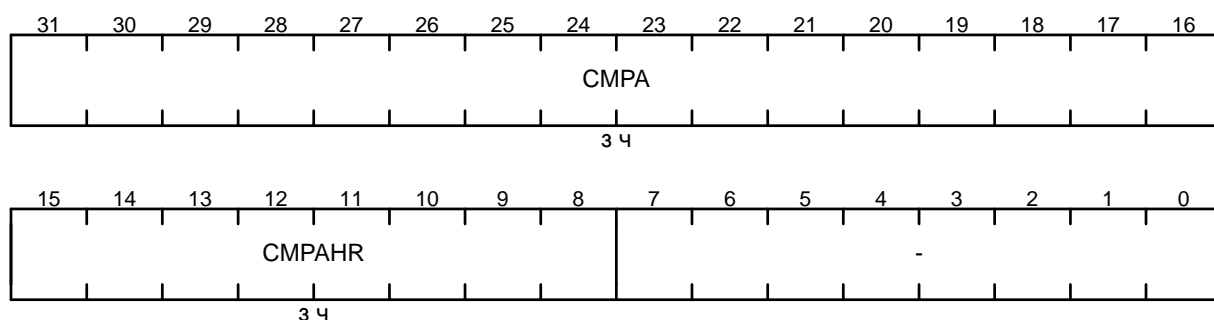


Поле	Биты	Описание
SHDWBFULL	9	Флаг отложенной загрузки в регистр CMPB
		0   Нет действий
SHDWAFULL	8	Флаг отложенной загрузки в регистр CMPA
		0   Нет действий
SHDWBMODE	6	Бит управления загрузкой регистра CMPB
		0   Значение, записываемое в регистр CMPB, размещается в теновом регистре (отложенная загрузка)
SHDWA MODE	4	Бит управления загрузкой регистра CMPA
		0   Значение, записываемое в регистр CMPA, размещается в теновом регистре (отложенная загрузка)
LOADBMODE	3-2	Поле задания события загрузки отложенного значения в регистр CMPB (при условии, что бит SHDWBMODE сброшен)
		00   CTR = Zero
		01   CTR = PRD
		10   CTR = Zero или CTR = PRD
LOADAMODE	1-0	Поле задания события загрузки отложенного значения в регистр CMPA (при условии, что бит SHDWA MODE сброшен)
		00   CTR = Zero
		01   CTR = PRD
		10   CTR = Zero или CTR = PRD
–	31-10, 7, 5	Зарезервировано

## СМРА – регистр порога срабатывания А

Смещение: + 18h

Сброс: 0h

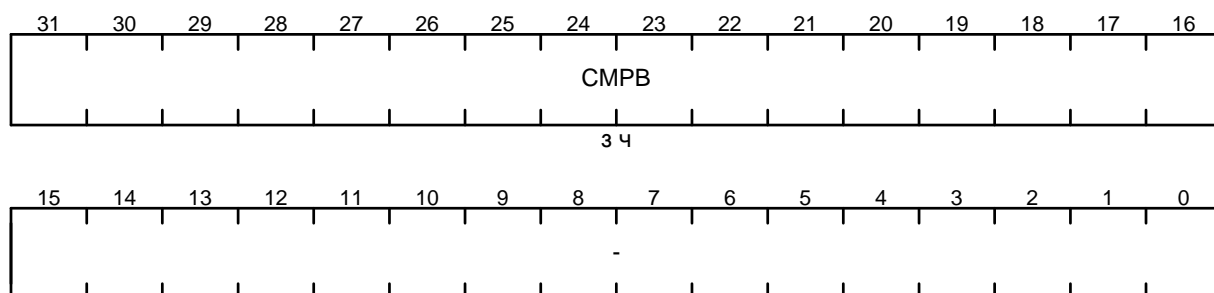


Поле	Биты	Описание
СМРА	31-16	Активное значение порога срабатывания канала А, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие $CTR = СМРА$ , которое влияет на поведение сигналов на линиях $PWMx\_A$ и $PWMx\_B$
СМРАНР	15-8	Дополнительные младшие биты значения порога срабатывания канала А (используются только для блока ШИМ высокого разрешения). Отложенная загрузка включается и работает также как и для поля СМРА
–	7-0	Зарезервировано

## СМРВ – регистр порога срабатывания В

Смещение: + 1Ch

Сброс: 0h

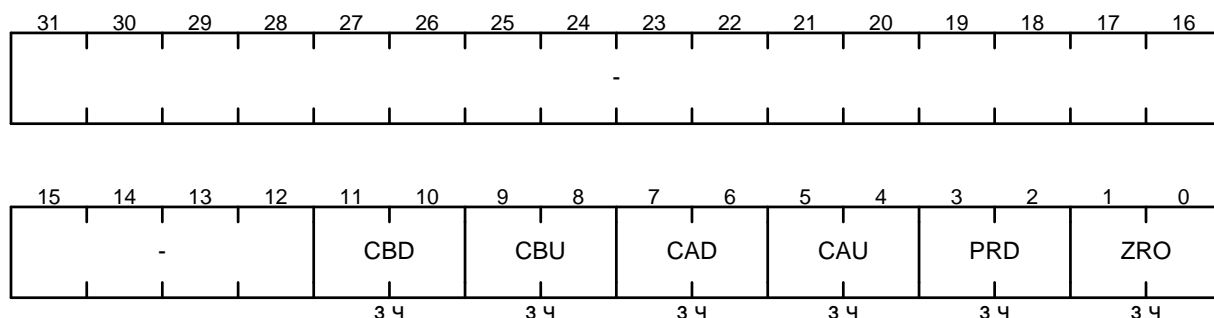


Поле	Биты	Описание
СМРВ	31-16	Активное значение порога срабатывания канала В, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие $CTR = СМРВ$ , которое влияет на поведение сигналов на линиях $PWMx\_A$ и $PWMx\_B$
–	15-0	Зарезервировано

## AQCTLA – регистр обработчика для выхода А

Смещение: + 20h

Сброс: 0h



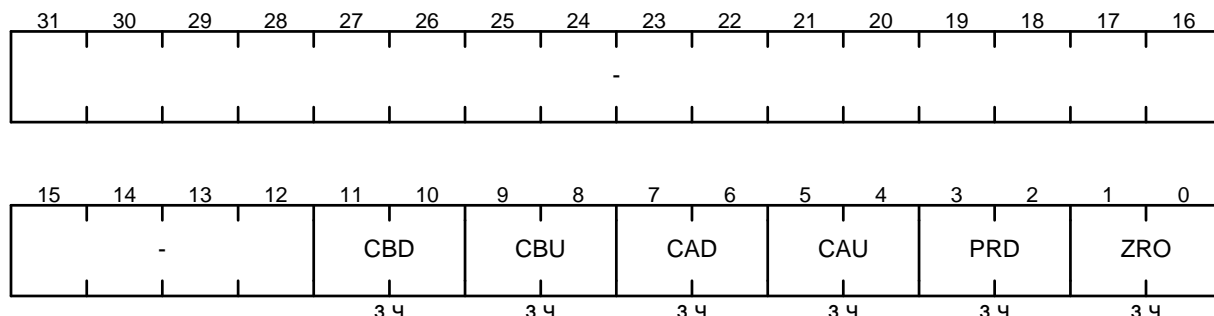
Поле	Биты	Описание
CBD	11-10	Действие на выводе PWMx_A при CTR = CMPB при счете вниз
CBU	9-8	Действие на выводе PWMx_A при CTR = CMPB при счете вверх
CAD	7-6	Действие на выводе PWMx_A при CTR = CMPA при счете вниз
CAU	5-4	Действие на выводе PWMx_A при CTR = CMPA при счете вверх
PRD	3-2	Действие на выводе PWMx_A при CTR = PRD
ZRO	1-0	Действие на выводе PWMx_A при CTR = Zero
–	31-12	Зарезервировано

Примечание – Для каждого события может быть задано одно из четырех действий:  
 00 – нет реакции;  
 01 – переключение вывода PWMx\_A в ноль;  
 10 – переключение вывода PWMx\_A в единицу;  
 11 – инверсия вывода PWMx\_A.

## AQCTLB – регистр обработчика для выхода В

Смещение: + 24h

Сброс: 0h

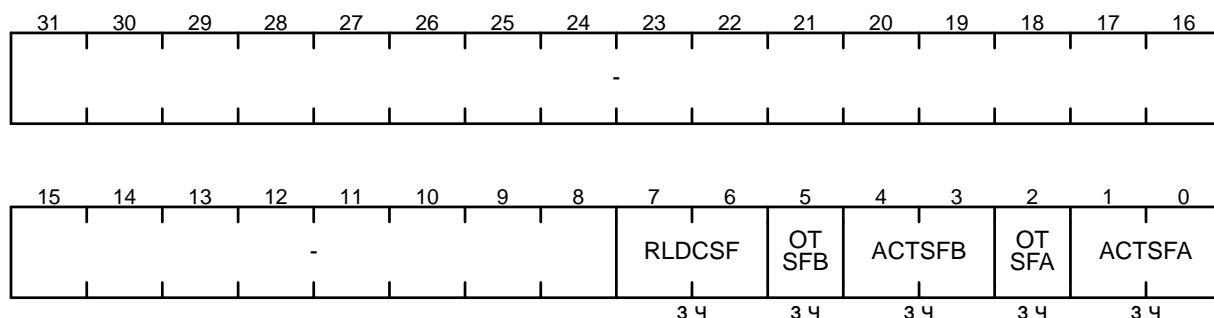


Примечание – Назначения битовых полей аналогичны назначению полей регистра AQCTLA, с той разницей, что относятся они к выводу PWMx\_B.

## AQSFRС – регистр программного управления однократным действием

Смещение: + 28h

Сброс: 0h

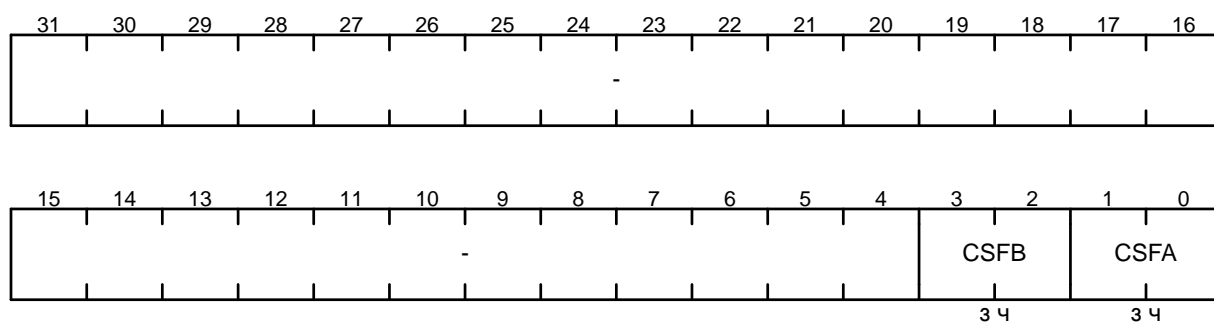


Поле	Биты	Описание	
RLDCSF	7-6	Управление событием теневой загрузки полей OTSFA, OTSFB, а также полей CSFA, CSFB регистра AQSFRС	
		00	CTR = Zero
		01	CTR = PRD
		10	CTR = Zero или CTR = PRD
	11	Без теневой загрузки, прямая запись в регистр	
OTSFB	5	Запись единицы приводит к однократному переключению вывода в состояние, согласно ACTSFB	
ACTSFB	4-3	Выбор действия с выходным сигналом на выводе	
		00	Нет действий
		01	PWMx_B = 0
		10	PWMx_B = 1
	11	Инверсия вывода PWMx_B	
OTSFA	2	Запись единицы приводит к однократному переключению вывода в состояние, согласно ACTSFA	
ACTSFA	1-0	Выбор действия с выходным сигналом на выводе	
		00	Нет действий
		01	PWMx_A = 0
		10	PWMx_A = 1
	11	Инверсия вывода PWMx_A	
–	31-8	Зарезервировано	

## AQCSFRC – регистр обработчика для циклического программного управления

Смещение: + 2Ch

Сброс: 0h



Поле	Биты	Описание
CSFB/ CSFA	3-2/ 1-0	Поле задания циклического воздействия на выход PWM <sub>x</sub> _B/PWM <sub>x</sub> _A
–	31-4	Зарезервировано

Примечание – Может быть задано одно из четырех воздействий:

00, 11 – нет реакции;

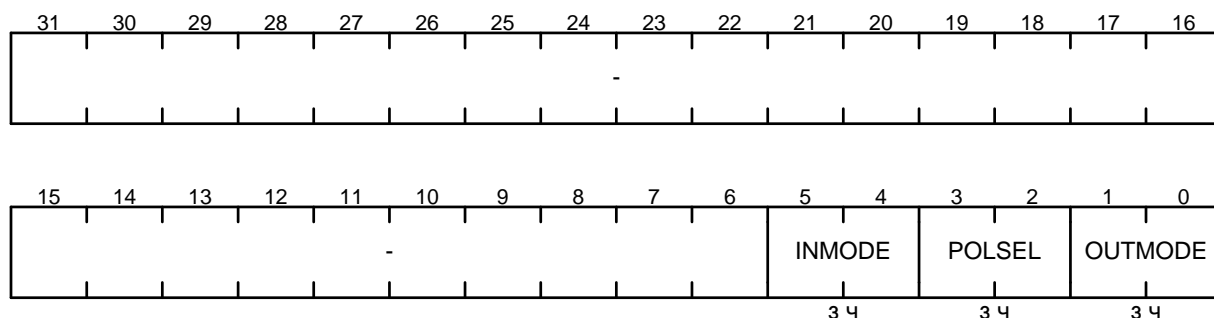
01 – значение 0 на выходе PWM<sub>x</sub>\_B/PWM<sub>x</sub>\_A;

10 – значение 1 на выходе PWM<sub>x</sub>\_B/PWM<sub>x</sub>\_A;

## DBCTL – регистр управления генератором «мертвого» времени ШИМ

Смещение: + 30h

Сброс: 0h

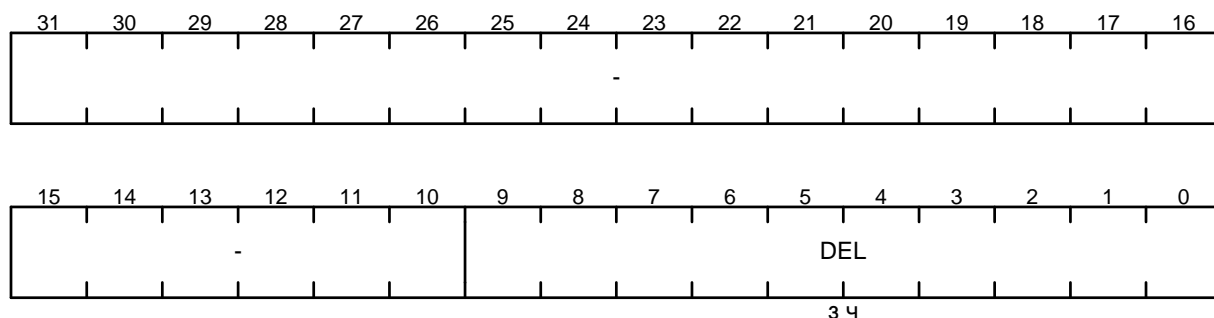


Поле	Биты	Описание
INMODE	5-4	Поле выбора источника для контроля по фронту и срезу. Старший бит поля управляет ключом S5, младший – ключом S4, см. рисунок 14.12
	00b	Входной сигнал PWMx_A используется для контроля по переднему и заднему фронтам
	01b	Входной сигнал PWMx_A используется для контроля по заднему фронту, а сигнал PWMx_B – по переднему
	10b	Входной сигнал PWMx_A используется для контроля по переднему фронту, а сигнал PWMx_B – по заднему
	11b	Входной сигнал PWMx_B используется для контроля по переднему и заднему фронтам
POLSEL	3-2	Поле задания полярности сигнала на выходе. Старший бит поля управляет ключом S3, а младший – ключом S2, см. рисунок 14.12
	00b	Инверсия запрещена
	01b	Инверсия только на выходе PWMx_A
	10b	Инверсия только на выходе PWMx_B
	11b	Инверсия на выходах PWMx_A и PWMx_B
OUTMODE	1-0	Поле выбора фронта, для которого включена задержка («мертвое» время). Старший бит поля управляет ключом S1, а младший – ключом S0, см. рисунок 14.12
	00b	Не задано
	01b	Задний фронт выхода PWMx_B
	10b	Передний фронт выхода PWMx_A
	11b	Передний фронт выхода PWMx_A и задний фронт выхода PWMx_B
–	31-6	Зарезервировано

### DBRED – регистр управления «мертвым» временем переднего фронта

Смещение: + 34h

Сброс: 0h

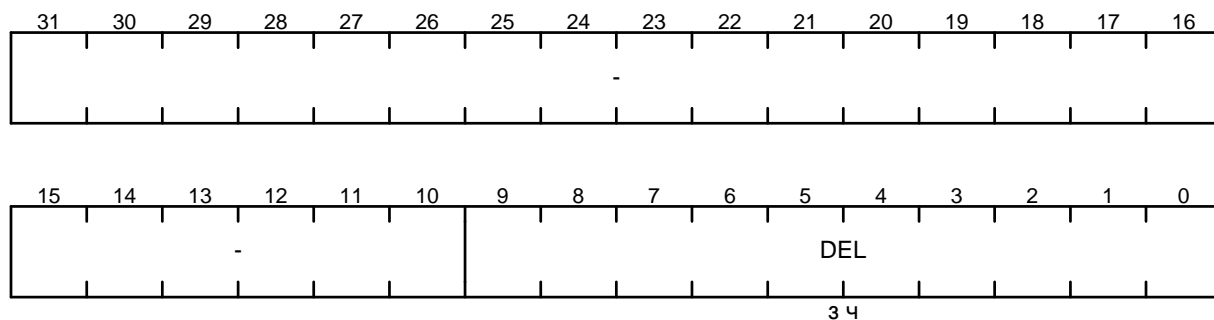


Поле	Биты	Описание
DEL	9-0	Величина задержки переднего фронта для генератора «мертвого» времени ШИМ (в периодах тактового сигнала TBCLK)
–	31-10	Зарезервировано

### DBFED – регистр управления «мертвым» временем заднего фронта

Смещение: + 38h

Сброс: 0h



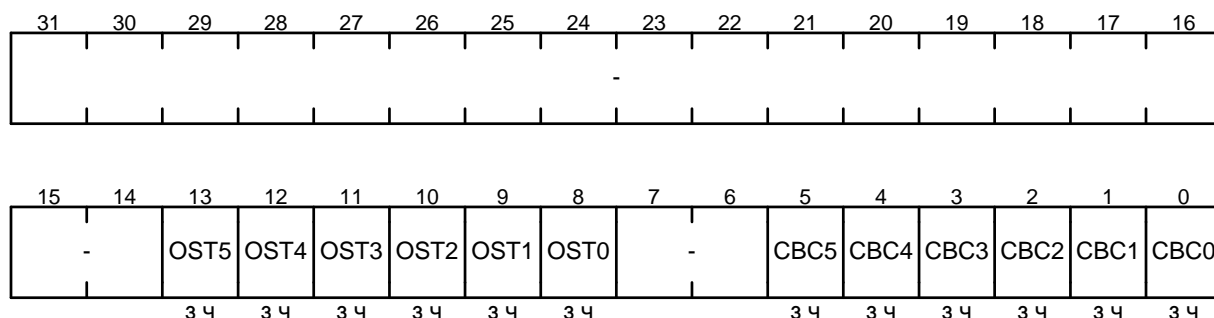
Поле	Биты	Описание
DEL	9-0	Величина задержки заднего фронта для генератора «мертвого» времени ШИМ (в периодах тактового сигнала TBCLK)
–	31-10	Зарезервировано



### TZSEL – регистр источника сигнала аварии

Смещение: + 3Ch

Сброс: 0h

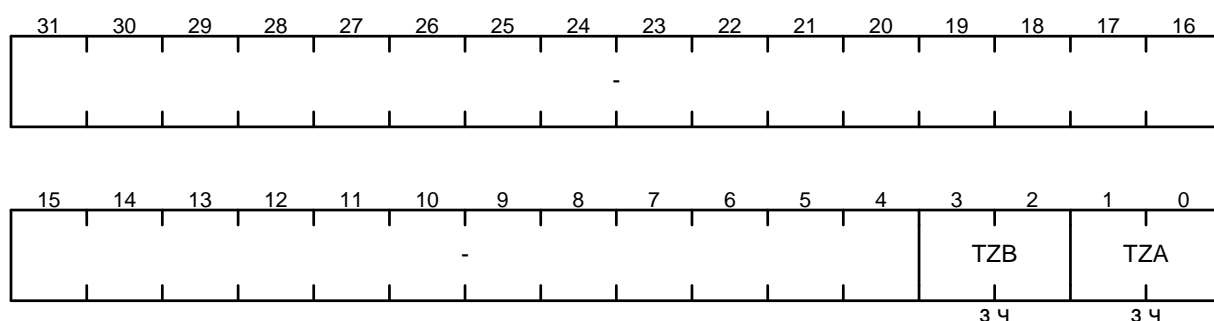


Поле	Биты	Описание
OSTn	13-8	Бит разрешения источника сигнала аварии с вывода PWM_TZn (n от 0 до 5) в однократном режиме
		0   Запрещено
		1   Разрешено
CBCn	5-0	Бит разрешения источника сигнала аварии с вывода PWM_TZn (номер вывода от 0 до 5) в циклическом режиме
		0   Запрещено
		1   Разрешено
-	31-14, 7-6	Зарезервировано

### TZCTL – регистр управления детектором сигнала аварии

Смещение: + 40h

Сброс: 0h

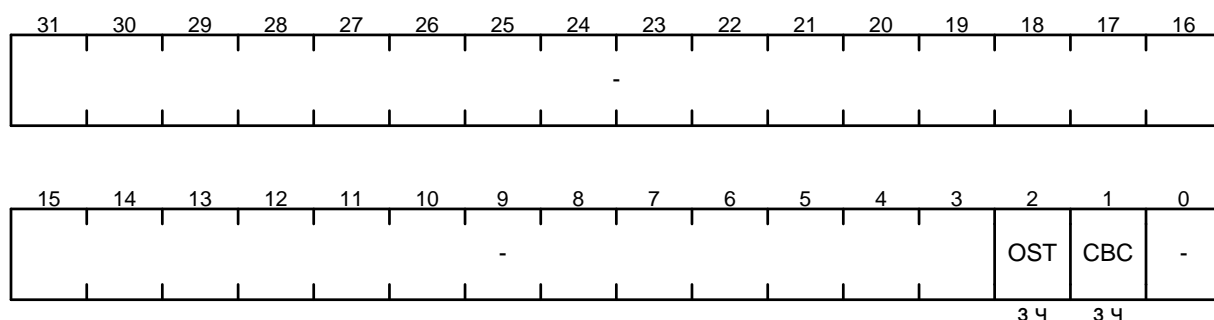


Поле	Биты	Описание
TZB/ TZA	3-2/ 1-0	Поле задания поведения вывода PWMx_B/PWMx_A в случае получения сигнала аварии. Источник сигнала аварии при этом определяется регистром TZSEL
		00b   Переключение в третье состояние
		01b   Переключение в единицу
		10b   Переключение в ноль
		11b   Нет действий
-	31-4	Зарезервировано

## TZEINT – регистр маски прерывания детектора сигнала аварии

Смещение: + 44h

Сброс: 0h

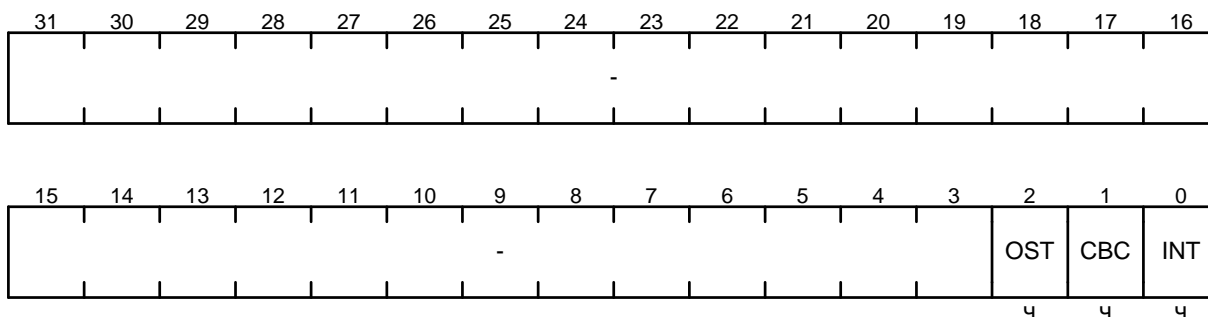


Поле	Биты	Описание
OST	2	Бит разрешения генерации прерывания в однократном режиме обработки аварии
		0   Запрещено
		1   Разрешено
CBC	1	Бит разрешения генерации прерывания в циклическом режиме обработки аварии
		0   Запрещено
		1   Разрешено
-	31-3, 0	Зарезервировано

## TZFLG – регистр флагов прерывания детектора сигнала аварии

Смещение: + 48h

Сброс: 0h

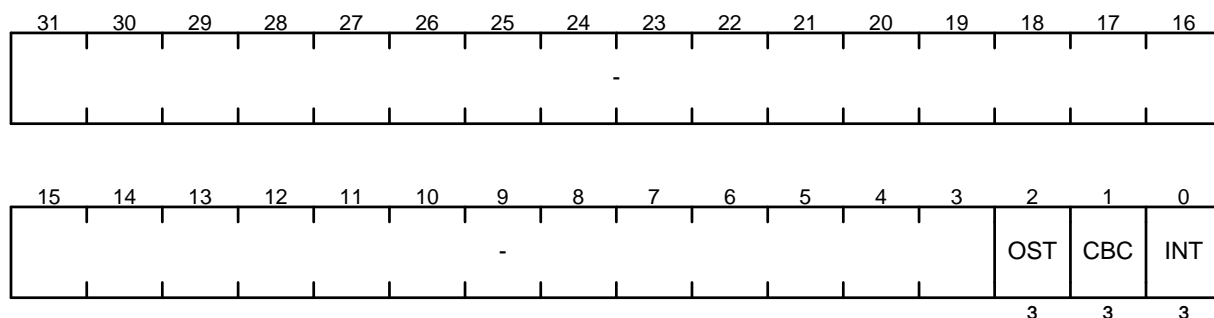


Поле	Биты	Описание
OST	2	Флаг прерывания в однократном режиме обработки аварии
		0   Нет прерывания
		1   Запрос на прерывание
При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова		
CBC	1	Флаг прерывания в циклическом режиме обработки аварии
		0   Нет прерывания
		1   Запрос на прерывание
При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова		
INT	0	Флаг внешнего прерывания NVIC
		0   Нет прерывания
		1   Запрос на прерывание
Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг установится снова		
–	31-3	Зарезервировано

### TZCLR – регистр сброса флагов прерываний детектора сигнала аварии

Смещение: + 4Ch

Сброс: 0h

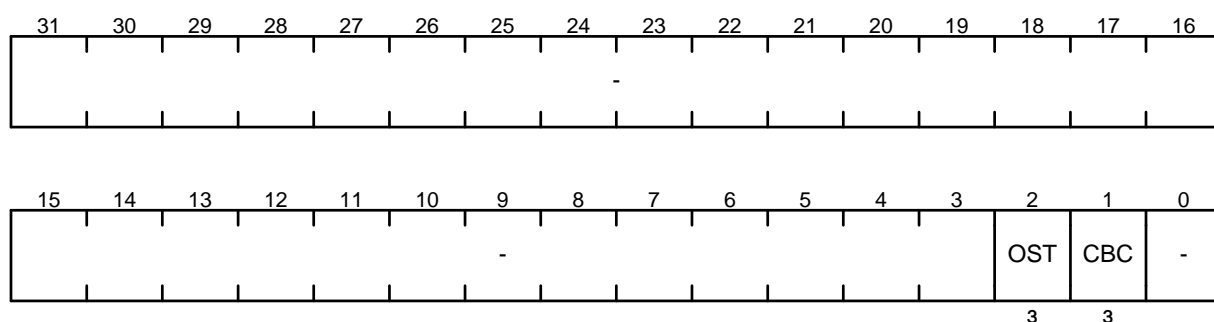


Поле	Биты	Описание
OST	2	Бит сброса флага прерывания в однократном режиме обработки аварии. Запись единицы сбрасывает бит OST в регистре TZFLG
CBC	1	Бит сброса флага прерывания в циклическом режиме обработки аварии. Запись единицы сбрасывает бит CBC в регистре TZFLG
INT	0	Бит сброса флага внешнего прерывания NVIC. Запись единицы сбрасывает бит INT в регистре TZFLG
–	31-3	Зарезервировано

### TZFRC – регистр программной эмуляции сигнала аварии

Смещение: + 50h

Сброс: 0h

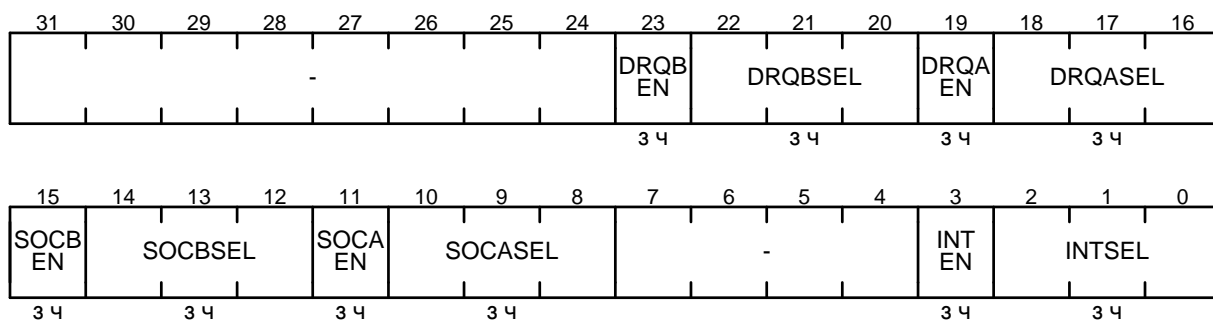


Поле	Биты	Описание
OST	2	Бит программной генерации сигнала аварии в однократном режиме. Запись единицы устанавливает бит OST в регистре TZFLG Чтение бита возвращает ноль
CBC	1	Бит программной генерации сигнала аварии в циклическом режиме. Запись единицы устанавливает бит CBC в регистре TZFLG Чтение бита возвращает ноль
–	31-3, 0	Зарезервировано

## ETSEL – регистр источника триггера событий

Смещение: + 54h

Сброс: 0h

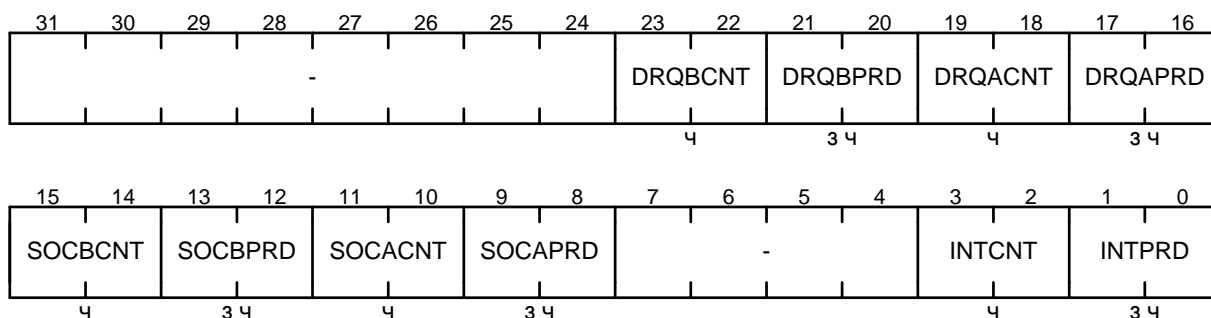


Поле	Биты	Описание
DRQBEN/ DRQAEN	23/ 19	Бит разрешения генерации запроса DMA
		0   Запрещено
		1   Разрешено
SOCBEN/ SOCAEN	15/ 11	Бит разрешения генерации внешнего сигнала PWM_SOCB/ PWM_SOCA для запуска АЦП
		0   Запрещено
		1   Разрешено
DRQBSEL/ DRQASEL/ SOCBSEL/ SOCASEL/ INTSEL	22-20/ 18-16/ 14-12/ 10-8/ 2-0	Поле выбора события, по которому будет сформирован импульс DMA_REQB/DMA_REQA/PWM_SOCB/PWM_SOCA/PWM_INT
		000   Зарезервировано
		001   CTR = Zero
		010   CTR = PRD
		011   Зарезервировано
		100   CTR = CMPA при счете вверх
		101   CTR = CMPA при счете вниз
		110   CTR = CMPB при счете вверх
111   CTR = CMPB при счете вниз		
INTEN	3	Бит разрешения генерации внешнего прерывания PWM_INT
		0   Запрещено
		1   Разрешено
–	31-24, 7-4	Зарезервировано

## ETPS – регистр делителя триггера событий

Смещение: + 58h

Сброс: 0h



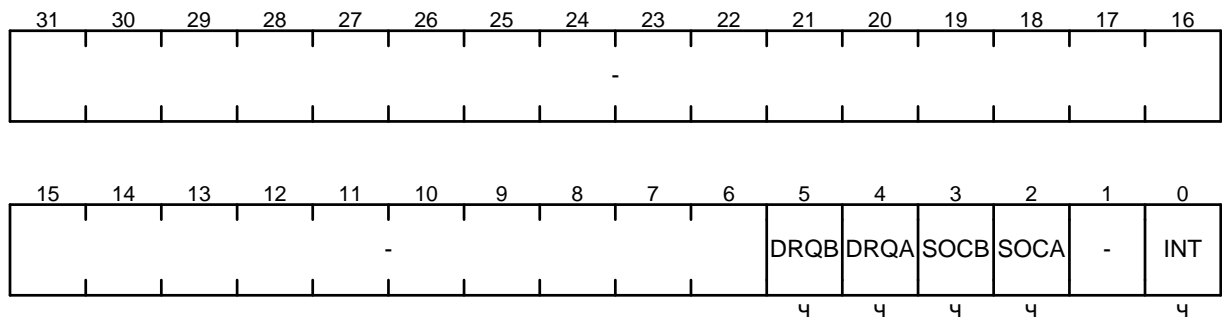
Поле	Биты	Описание
DRQBCNT/ DRQACNT/ SOCBCNT/ SOCACNT	23-22/ 19-18/ 15-14/ 11-10	Счетчик событий сигнала DMA_REQB/DMA_REQA/PWM_SOCB/ PWM_SOCA. Счетчик автоматически сбрасывается, когда сформировано прерывание и перестает считать, когда достигает значения DRQBPRD/ DRQAPRD/SOCBPRD/SOCAPRD соответственно.
DRQBPRD/ DRQAPRD	21-20/ 17-16	Поле задания количества событий, выбранных полем DRQBSEL/ DRQASEL регистра ETSEL, по которым будет сформирован сигнал запуска DMA_REQB/DMA_REQA. Для разрешения генерации сигнала нужно установить бит DRQBEN/ DRQAEN регистра ETSEL. Сигнал будет сформирован, даже если флаг DRQB/DRQA (регистр ETFLG) предыдущего сигнала не был сброшен. Как только сигнал DMA_REQB/DMA_REQA отправлен, счетчик DRQBCNT/DRQACNT автоматически сбрасывается.
		00      Выдача сигнала по каждому событию
		01      Выдача сигнала каждые два события
		10      Выдача сигнала каждые три события
		11      Выдача сигнала каждые четыре события
SOCBPRD/ SOCAPRD	13-12/ 9-8	Поле задания количества событий, заданных полем SOCBSSEL/ SOCASEL регистра ETSEL, по которым будет сформирован сигнал запуска АЦП PWM_SOCB/PWM_SOCA. Для разрешения генерации сигнала нужно установить бит SOC BEN/ SOC AEN регистра ETSEL. Сигнал будет сформирован, даже если флаг SOC B/SOC A (регистр ETFLG) предыдущего сигнала не был сброшен. Как только сигнал PWM_SOCB/PWM_SOCA отправлен, счетчик SOCBCNT/SOCACNT автоматически сбрасывается.
		00      Выдача сигнала по каждому событию
		01      Выдача сигнала каждые два события
		10      Выдача сигнала каждые три события
		11      Выдача сигнала каждые четыре события
INTCNT	3-2	Значение счетчика событий прерываний. Счетчик автоматически сбрасывается, когда сформировано прерывание и перестает считать, когда достигает значения INTPRD.

Поле	Биты	Описание	
INTPRD	1-0	Поле задания количества событий, выбранных полем INTSEL регистра ETSEL, по которым будет сформировано внешнее прерывание PWM_INT. Для разрешения генерации прерывания нужно установить бит INTEN в регистре ETSEL. Если флаг прерывания INT (регистр ETFLG) установлен от предыдущего прерывания, то текущее прерывание не будет активировано до сброса этого флага (сбрасывается записью единицы в бит INT регистра ETCLR). Такой механизм позволяет обрабатывать одно прерывание, в то время как другое ждет своей очереди	
		00	Прерывания по каждому событию (INTCNT = 00b)
		01	Прерывания каждые два события (INTCNT = 01b)
		10	Прерывания каждые три события (INTCNT = 10b)
		11	Прерывания каждые четыре события (INTCNT = 11b)
–	31-24, 7-4	Зарезервировано	

### ETFLG – регистр флагов триггера событий

Смещение: + 5Ch

Сброс: 0h

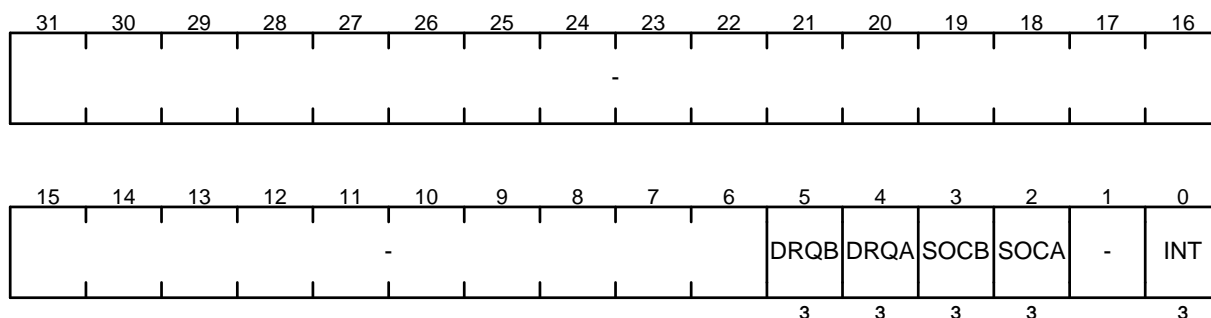


Поле	Биты	Описание
DRQB/ DRQA	5/ 4	Флаг запроса DMA_REQB/DMA_REQA
		0 Не установлен или сброшен
		1 Установлен
SOCB/ SOCA/ INT	3/ 2/ 0	Флаг внешнего сигнала АЦП PWM_SOCB/PWM_SOCA/PWM_INT
		0 Не установлен или сброшен
		1 Установлен
–	31-6, 1	Зарезервировано

## ETCLR – регистр сброса флагов триггера событий

Смещение: + 60h

Сброс: 0h

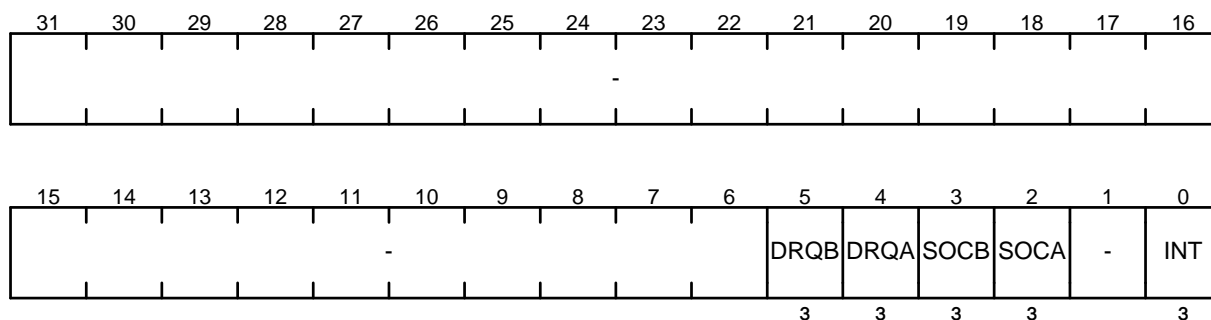


Поле	Биты	Описание	
DRQB/ DRQA	5/	Бит сброса флага DRQB/DRQA в регистре ETFLG	
	4	0	Нет действий
		1	Запись единицы сбрасывает флаг
SOCB/ SOCA/ INT	3/	Бит сброса флага SOCB/SOCA/INT в регистре ETFLG	
	2/	0	Нет действий
		0	1
–	31-6, 1	Зарезервировано	

## ETFRC – регистр программной эмуляции флагов триггера событий

Смещение: + 64h

Сброс: 0h



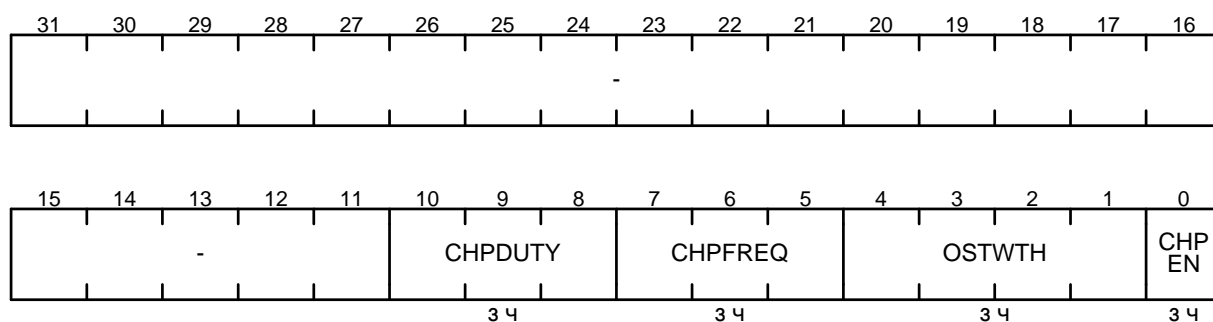
Поле	Биты	Описание	
DRQB/ DRQA	5/	Бит программной установки флага DRQB/DRQA в регистре ETFLG	
	4	0	Нет действий
		1	Запись единицы устанавливает флаг
SOCB/ SOCA/ INT	3/	Бит программной установки флага SOCB/SOCA/INT в регистре ETFLG	
	2/	0	Нет действий
		0	1
–	31-4, 1	Зарезервировано	



## PCCTL - регистр управления модулятором

Смещение: + 68h

Сброс: 0h

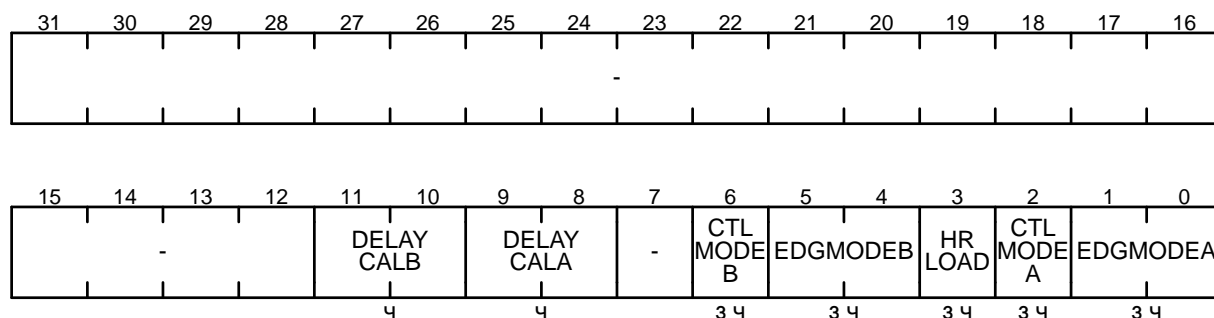


Поле	Биты	Описание
CHPDUTY	10-8	Поле задания коэффициента заполнения второго и последующих импульсов
		000   1/8 (13,5 %)
		001   2/8 (25,0 %)
		...   ...
		110   7/8 (87,5 %)
		111   Зарезервировано
CHPFREQ	7-5	Поле выбора делителя частоты синхронизации для задания частоты второго и последующих импульсов
		000   1
		001   1/2
		...   ...
		110   1/7
		111   1/8
OSTWTH	4-1	Поле задания ширины первого импульса
		0h   1 × T <sub>PCLK</sub> ×8
		1h   2 × T <sub>PCLK</sub> ×8
		...   ...
		Eh   15 × T <sub>PCLK</sub> ×8
		Fh   16 × T <sub>PCLK</sub> ×8
CHPEN	0	Бит разрешения работы модулятора
		0   Запрещено
		1   Разрешено
–	31-11	Зарезервировано

## HRCTL – регистр управления блоком ШИМ высокого разрешения

Смещение: + 6Ch

Сброс: 0h

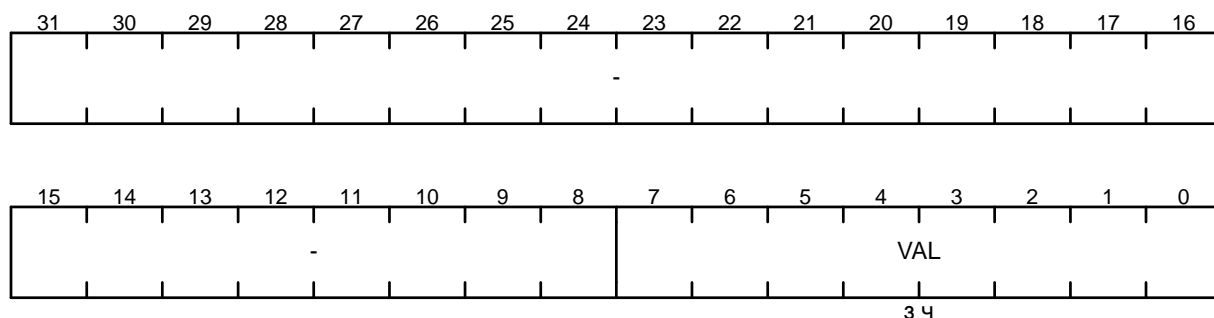


Поле	Биты	Описание	
DELAYCALB/ DELAYCALA	11-10/ 9-8	Флаги для оценки реального времени работы линии задержки на выводе PWMx_B/PWMx_A	
		00	Величина выбранной задержки меньше одного периода системной частоты
		01	Величина выбранной задержки больше или равна одному периоду системной частоты
		10	Зарезервировано
		11	Величина выбранной задержки больше или равна двум периодам системной частоты
CTLMODEB/ CTLMODEA	6/ 2	Бит выбора регистра для задания значение задержки на выводе PWMx_B/PWMx_A	
		0	Поле SMPANR
		1	Поле TVPHSHR
EDGEMODEB/ EDGEMODEA	5-4/ 1-0	Поле выбора фронта сигнала ШИМ, который сдвигается линией задержки на выводе PWMx_B/PWMx_A	
		00b	Зарезервировано
		01b	Передний
		10b	Задний
HRLOAD	3	Бит выбора события, по которому производится загрузка отложенного значения в активный регистр SMPANR	
		0	Событие CTR = Zero
		1	Событие CTR = PRD. При этом режим отложенной загрузки доступен, только если бит CTLMODE сброшен, а загрузка поля SMPA осуществляется аналогичным образом (в регистре SMPCTL поле LOADMODE = 00/01)
–	31-12, 7	Зарезервировано	

## FWDTH – регистр ширины фильтрации

Смещение: + 70h

Сброс: 0h

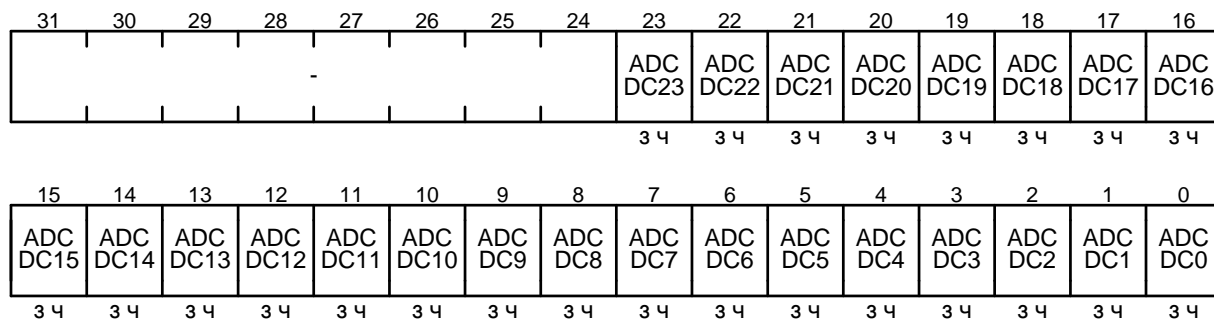


Поле	Биты	Описание
VAL	7-0	Поле задания ширины фильтрации коротких импульсов в тактах PCLK. Если $f_{PCLK} = 100$ МГц, то фильтруются импульсы до 2,55 мкс с шагом 0,01 мкс.
		0h   Фильтр выключен
		1h   1 такт PCLK
		...   ...
FFh   255 тактов PCLK		
–	31-8	Зарезервировано

## HDSEL0 – регистр источника сигнала события удержания 0

Смещение: + 74h

Сброс: 0h

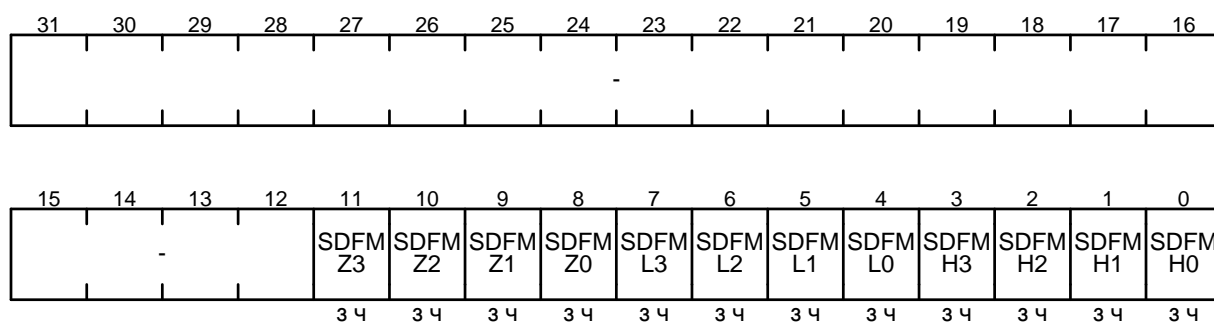


Поле	Биты	Описание
ADCDCm	23-0	Бит выбора цифрового компаратора m (m от 0 до 23) блока АЦП, с выхода которого берется сигнал для формирования события удержания
		0   Не выбран
		1   Выбран
–	31-24	Зарезервировано

## HDSEL1 – регистр источника сигнала события удержания 1

Смещение: + 78h

Сброс: 0h

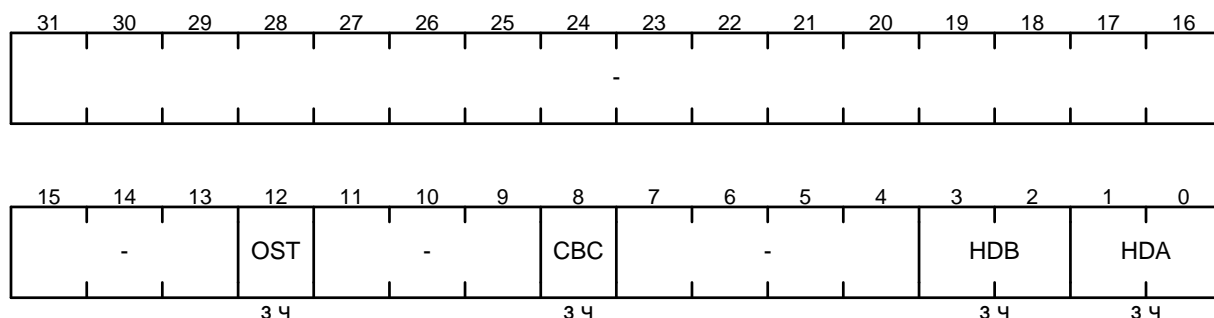


Поле	Биты	Описание
SDFMZ3– SDFMZ0	11-8	Бит выбора сигналов SDFM_CMPHZ0 – SDFM_CMPHZ3 компараторов блока SDFM для формирования события удержания
		0   Не выбран 1   Выбран
SDFML3– SDFML0	7-4	Бит выбора сигналов SDFM_CMPL0 – SDFM_CMPL3 компараторов блока SDFM для формирования события удержания
		0   Не выбран 1   Выбран
SDFMH3– SDFMH0	3-0	Бит выбора сигналов SDFM_CMPH0 – SDFM_CMPH3 компараторов блока SDFM для формирования события удержания
		0   Не выбран 1   Выбран
–	31-12	Зарезервировано

## HDCTL – регистр управления детектором событий удержания

Смещение: + 7Ch

Сброс: 0h

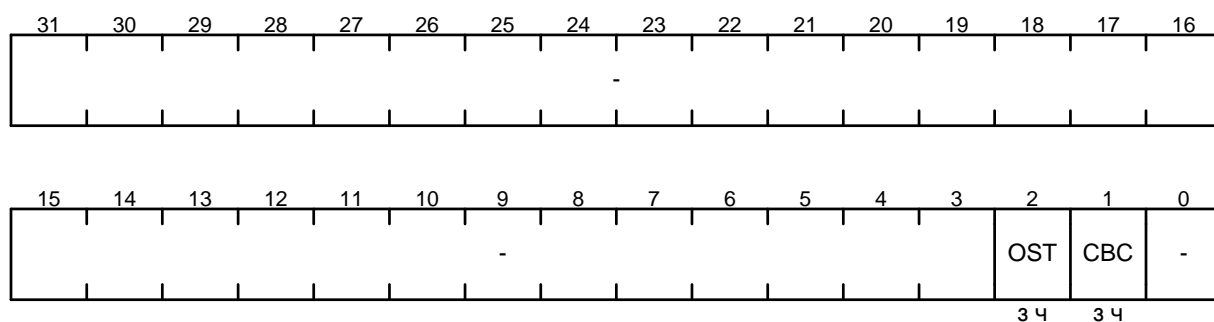


Поле	Биты	Описание
OST	12	Бит разрешения события по источнику ADCDC/SDFM в однократном режиме обработки аварии
		0   Запрещено 1   Разрешено
CBC	8	Бит разрешения события по источнику ADCDC/SDFM в циклическом режиме обработки аварии
		0   Запрещено 1   Разрешено
HDB/ HDA	3-2/ 1-0	Поле задания поведения сигнала PWMx_B/PWMx_A в случае сбоя (аварии). (Источник сбоя определяется регистрами HDSEL0 и HDSEL1)
		00b   Зарезервировано
		01b   Переключается в состояние единицы
		10b   Переключается в состояние нуля 11b   Остается без изменений
–	31-13, 11-9, 7-4	Зарезервировано

## HDEINT – регистр маски прерывания порогового выключателя

Смещение: + 80h

Сброс: 0h

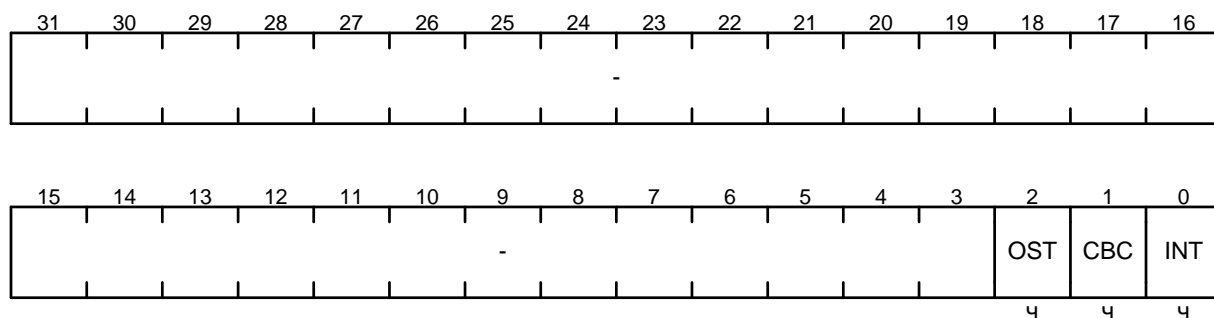


Поле	Биты	Описание	
OST	2	Бит разрешения генерации прерывания в однократном режиме	
		0	Запрещено
		1	Разрешено
CBC	1	Бит разрешения генерации прерывания в циклическом режиме	
		0	Запрещено
		1	Разрешено
–	31-3, 0	Зарезервировано	

## HDFLG – регистр флагов прерывания порогового выключателя

Смещение: + 84h

Сброс: 0h

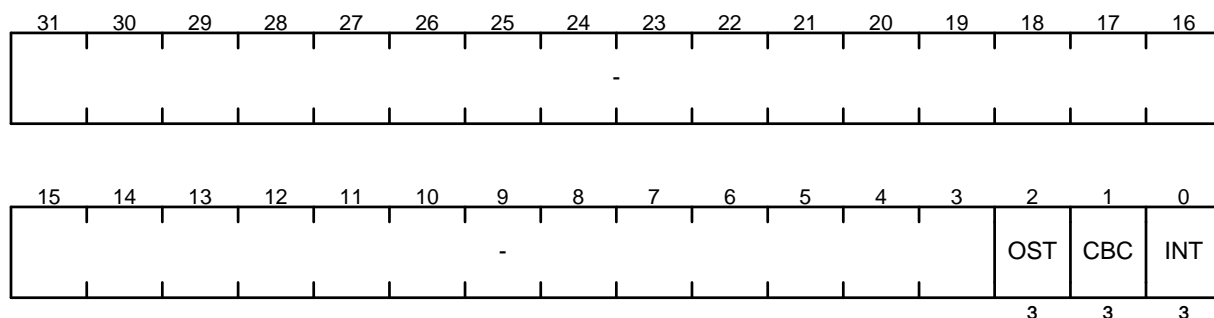


Поле	Биты	Описание
OST	2	Флаг прерывания в однократном режиме. Примечание – При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова
CBC	1	Флаг прерывания в циклическом режиме. Примечание – При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова
INT	0	Флаг внешнего прерывания NVIC. Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг INT установится снова
–	31-3	Зарезервировано

## HDCLR – регистр сброса флагов порогового выключателя

Смещение: + 88h

Сброс: 0h



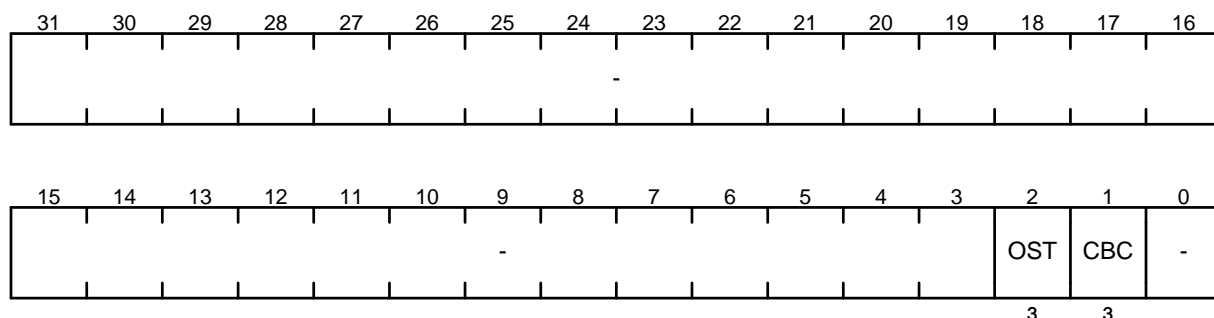
Поле	Бит	Описание
OST	2	Бит сброса флага прерывания в однократном режиме
CBC	1	Бит сброса флага прерывания в циклическом режиме
INT	0	Бит сброса флага внешнего прерывания NVIC
–	31-3	Зарезервировано

Примечание – Запись единицы в бит регистра сбрасывает соответствующий бит в регистре HDFLG

## HDFRC – регистр программной активации порогового выключателя

Смещение: + 8Ch

Сброс: 0h



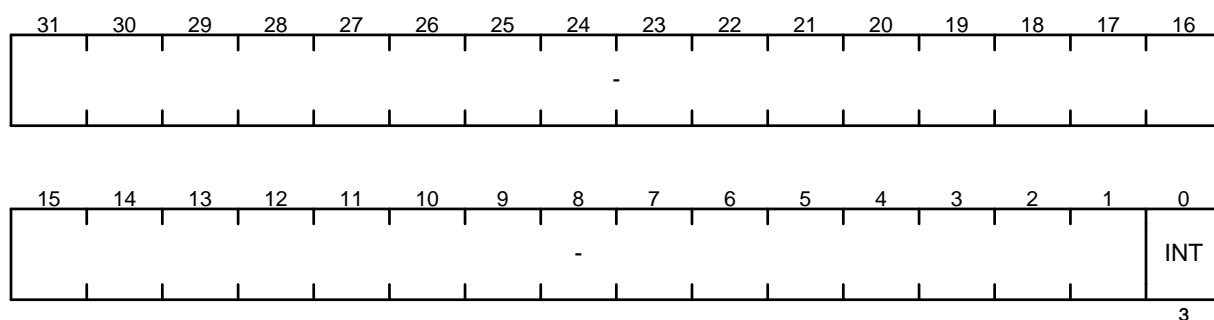
Поле	Бит	Описание
OST	2	Бит активации порогового выключателя в однократном режиме обработки аварии
		0   Нет действий
		1   Запись единицы активирует выключатель и устанавливает флаг OST в регистре HDFLG
CBC	1	Бит активации порогового выключателя в циклическом режиме обработки аварии
		0   Нет действий
		1   Запись единицы активирует выключатель и устанавливает флаг CBC в регистре HDFLG
–	31-3, 0	Зарезервировано



### HDINTCLR – регистр сброса прерывания порогового выключателя

Смещение: + 90h

Сброс: 0h

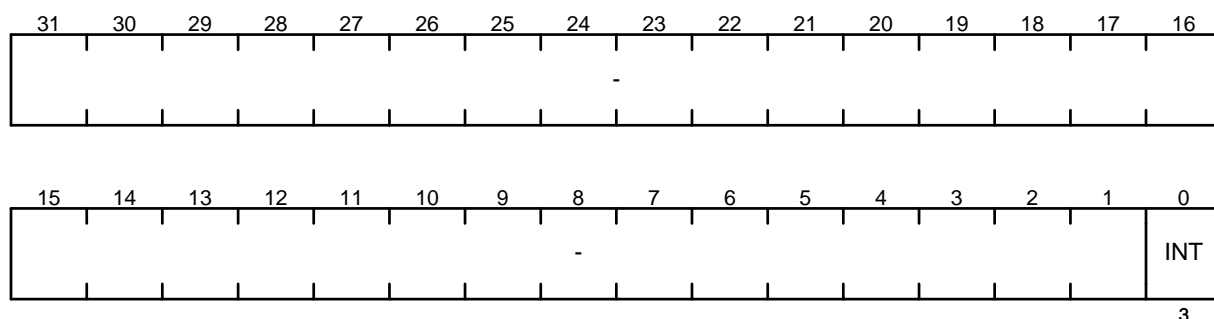


Поле	Биты	Описание
INT	0	Бит сброса прерывания. Сброс запроса на прерывание должен производиться программой обработки прерывания, во избежание повторного запуска обработчика
–	31-1	Зарезервировано

### TZINTCLR – регистр сброса прерывания детектора событий аварии

Смещение: + 94h

Сброс: 0h

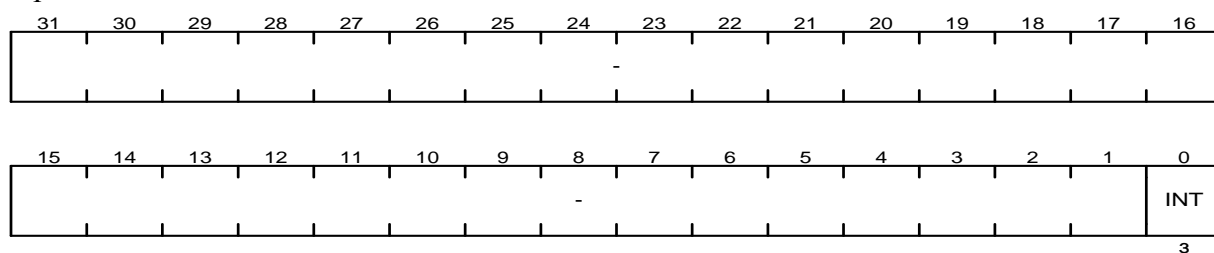


Примечание – Назначение и функционирование аналогично регистру HDINTCLR

### INTCLR – регистр сброса прерывания таймера блока ШИМ

Смещение: + 98h

Сброс: 0h



Примечание – Назначение и функционирование аналогично регистру HDINTCLR.

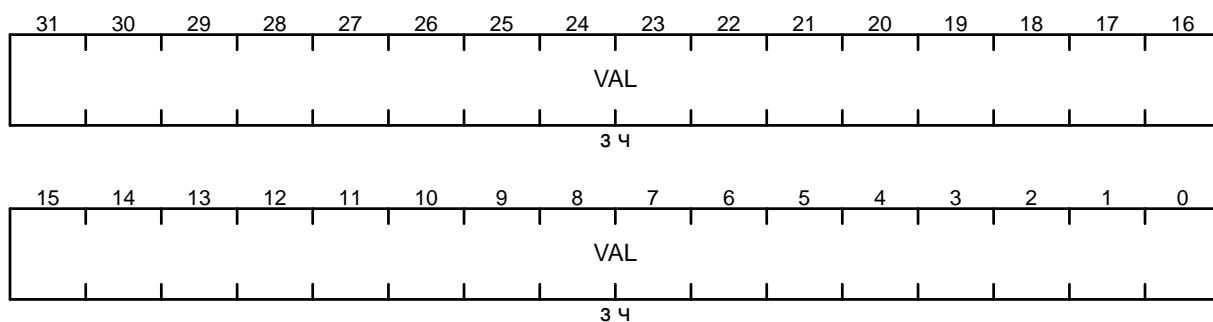
## A.18 Регистры модуля квадратурного декодера QEP

<b>Базовый адрес:</b>	400A_A000h	Регистры квадратурного декодера 0
	400A_B000h	Регистры квадратурного декодера 1
	400A_C000h	Регистры квадратурного декодера 2
	400A_D000h	Регистры квадратурного декодера 3

### QPOSCNT – регистр счета счетчика позиции

Смещение: + 00h

Сброс: 0h

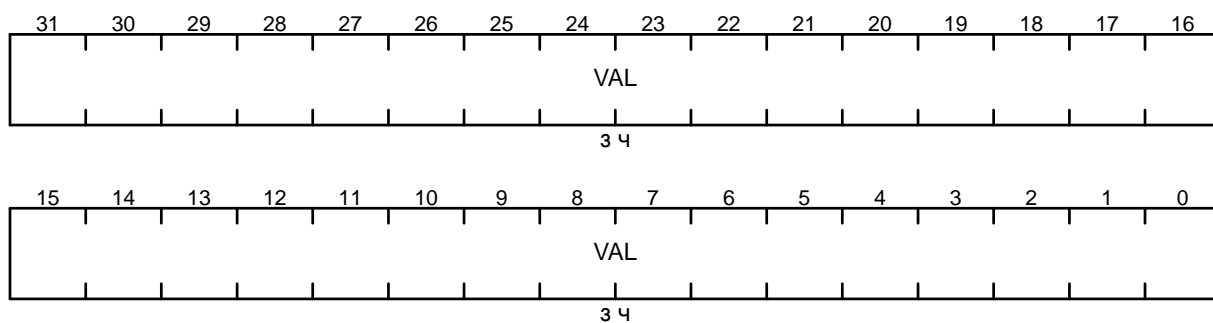


Поле	Биты	Описание
VAL	31-0	Значение счета счетчика позиции

### QPOSINIT – регистр инициализации счетчика позиции

Смещение: + 04h

Сброс: 0h

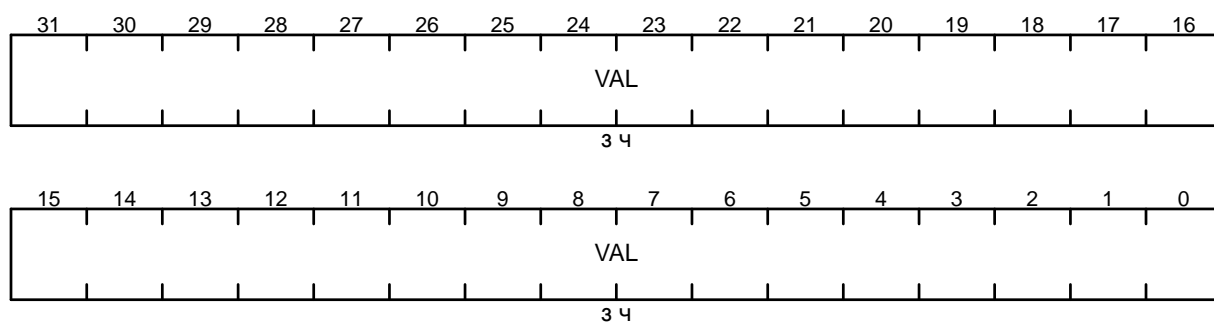


Поле	Биты	Описание
VAL	31-0	Значение инициализации счетчика позиции

### QPOSMAX – регистр максимального значения счетчика позиции

Смещение: + 08h

Сброс: 0h

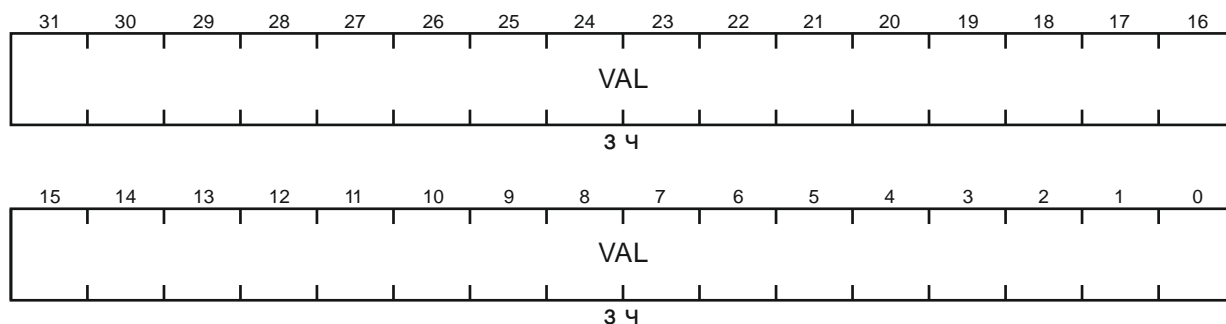


Поле	Биты	Описание
VAL	31-0	Значение максимального значения счетчика позиции

### QPOSCMP – регистр сравнения счетчика позиции

Смещение: + 0Ch

Сброс: 0h

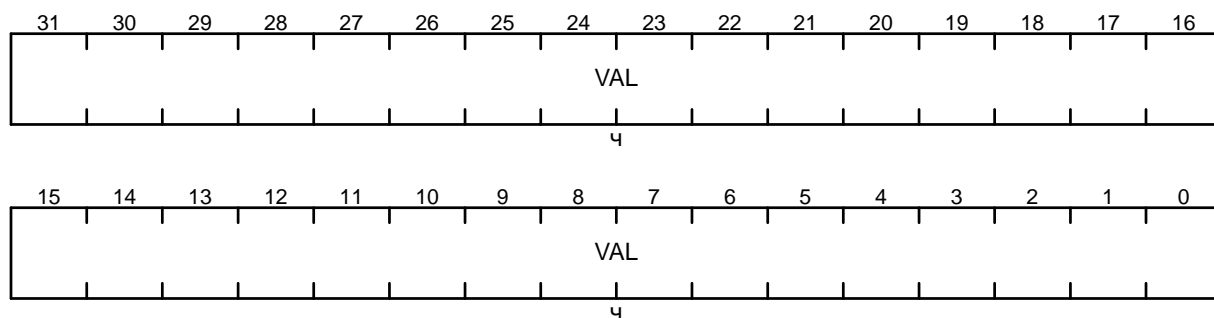


Поле	Биты	Описание
VAL	31-0	Значение сравнения счетчика позиции

### QPOSILAT – регистр хранения позиции по индексации

Смещение: + 10h

Сброс: 0h

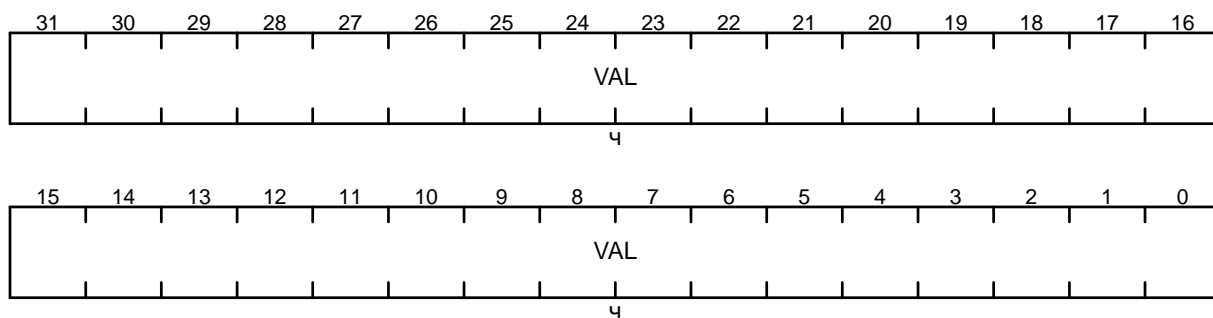


Поле	Биты	Описание
VAL	31-0	Значение хранения позиции по индексации

### QPOSSLAT – регистр хранения позиции по стробу

Смещение: + 14h

Сброс: 0h

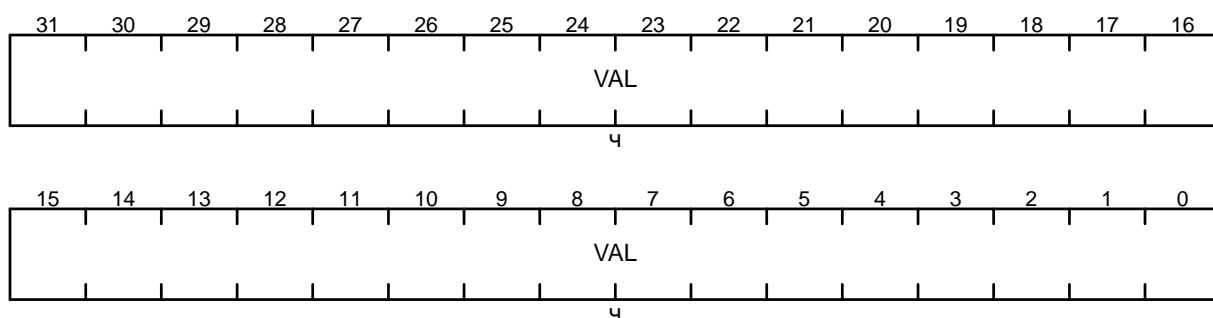


Поле	Биты	Описание
VAL	31-0	Значение хранения позиции по стробу

### QPOSLAT – регистр хранения позиции по таймеру временных отсчетов

Смещение: + 18h

Сброс: 0h

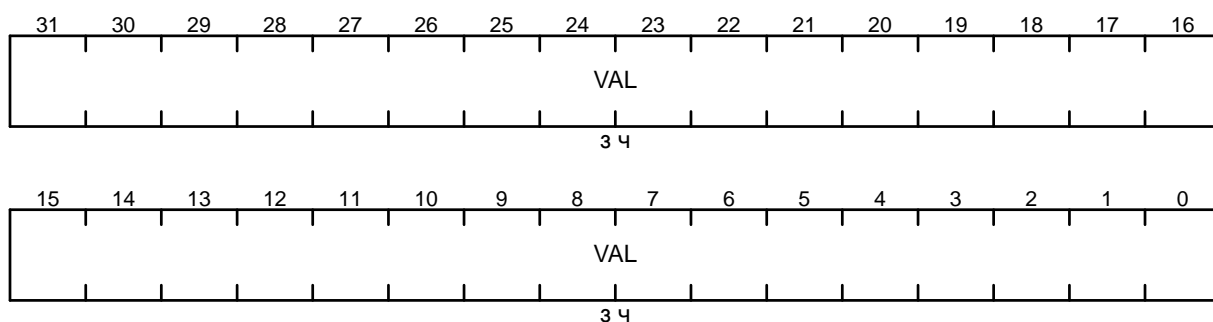


Поле	Биты	Описание
VAL	31-0	Значение хранения позиции по таймеру временных отсчетов

### QUTMR – регистр таймера временных отсчетов

Смещение: + 1Ch

Сброс: 0h

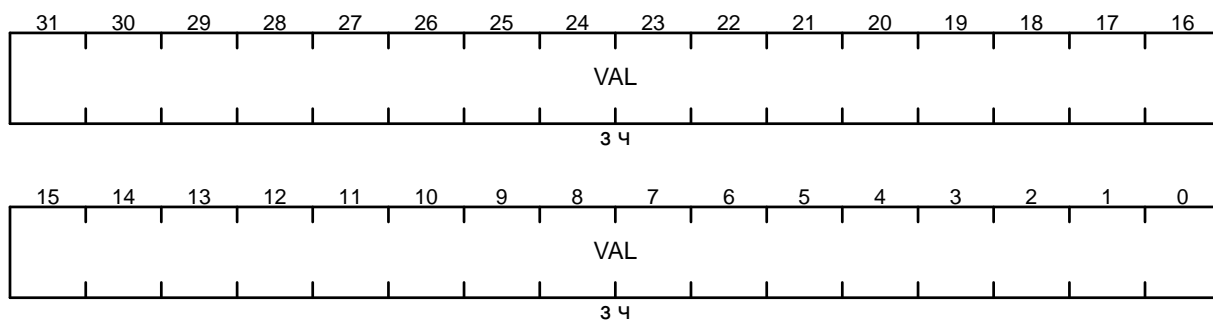


Поле	Биты	Описание
VAL	31-0	Значение таймера временных отсчетов

### QUPRD – регистр длительности счета таймера временных отсчетов

Смещение: + 20h

Сброс: 0h

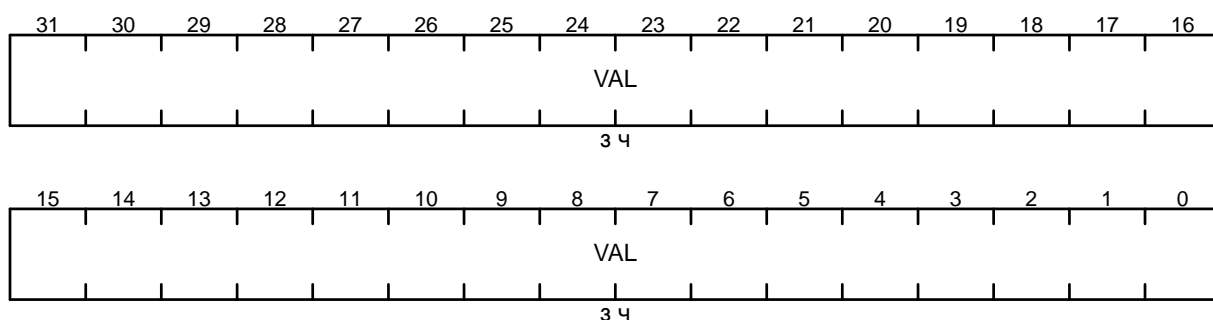


Поле	Биты	Описание
VAL	31-0	Значение длительности счета таймера временных отсчетов

### QWDTMR – регистр счета сторожевого таймера

Смещение: + 24h

Сброс: 0h

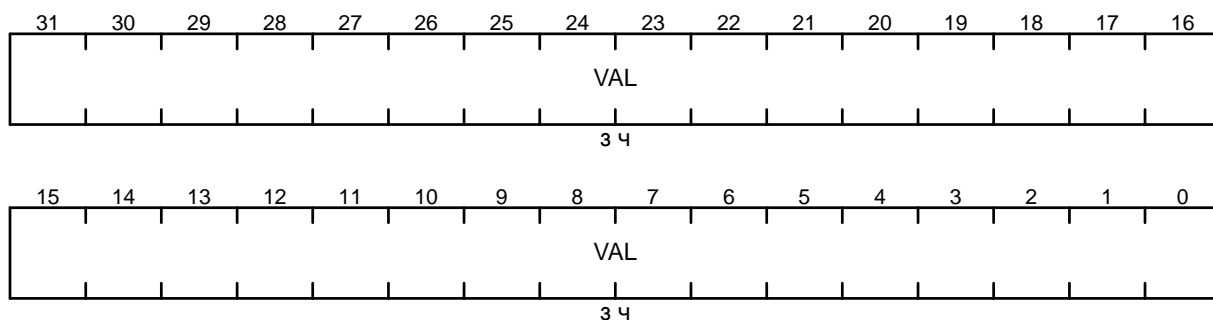


Поле	Биты	Описание
VAL	31-0	Значение счета сторожевого таймера

### QWDPRD – регистр длительности счета сторожевого таймера

Смещение: + 28h

Сброс: 0h

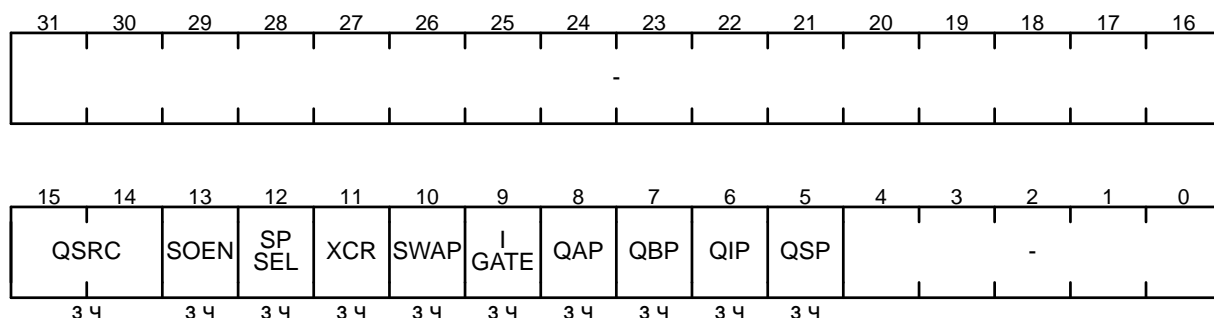


Поле	Биты	Описание
VAL	31-0	Значение длительности счета сторожевого таймера

## QDECCTL – регистр управления входами

Смещение: + 2Ch

Сброс: 0h



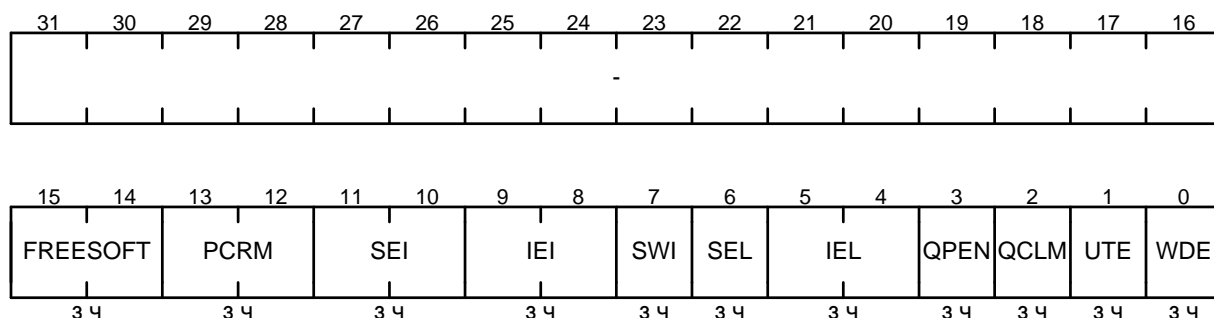
Поле	Биты	Описание
QSRC	15-14	Режим работы
		00   Квадратурный
		01   Счета/направления
		10   Счет вверх (QCLK = xCLK, QDIR = 1), 11   Счет вниз (QCLK = xCLK, QDIR = 0).
SOEN	13	Бит разрешения выдачи выходного сигнала компаратора
		0   Запрещено 1   Разрешено
SPSEL	12	Бит выбора вывода для выдачи выходного сигнала компаратора
		0   Стробующий вывод 1   Индексный вывод
XCR	11	Бит выбора фронта квадратурного входа
		0   Передний фронт 1   Передний и задний фронты
SWAP	10	Бит обмена входов QEPx_A и QEPx_B
		0   Нет действий 1   Входы QEPx_A и QEPx_B меняются местами
IGATE	9	Бит включения стробирования входного сигнала индексации
QAP	8	Бит включения инвертирования входного сигнала с QEPx_A
QBP	7	Бит включения инвертирования входного сигнала с QEPx_B
QIP	6	Бит включения инвертирования входного сигнала с QEPx_I
QSP	5	Бит включения инвертирования входного сигнала с QEPx_S
–	31-16, 4-0	Зарезервировано

Примечание – Для битов с 9 по 5 справедливо: 0 – выключено, 1 – включено.

## QEPCTL – регистр управления квадратурного декодера

Смещение: + 30h

Сброс: 0h



Поле	Биты	Описание	
FREESOFT	15-14	Поле управления счетчиками QPOSCNT, QWDTMR, QUTMR, QCTMR в режиме отладки	
		00	Принудительная блокировка счета
		01	Счет до переполнения
		10, 11	Разблокирование счета
PCRM	13-12	Поле задания события для сброса счетчика позиции	
		00	Событие индексации
		01	Достижение максимальной позиции
		10	Первое событие индексации
		11	Окончание временного отсчета
SEI	11-10	Поле задания события стробирования для инициализации счетчика позиции (QPOSCNT = QPOSINIT)	
		00, 01	Работа без инициализации
		10	Передний фронт сигнала QEPx_S
		11	Передний фронт QEPx_S при вращении по часовой стрелке или задний фронт QEPx_S при вращении против часовой стрелки
IEI	9-8	Поле задания события индексации для инициализации счетчика позиции (QPOSCNT = QPOSINIT)	
		00, 01	Работа без инициализации
		10	По переднему фронту сигнала QEPx_I
		11	По заднему фронту сигнала QEPx_I
SWI	7	Бит программной инициализации счетчика позиции. Не сбрасывается аппаратно	
		0	Нет действий
		1	Запись единицы загружает счетчик позиции QPOSCNT значением QPOSINIT
SEL	6	Бит задания события стробирования для сохранения значения счетчика позиции (QPOSSLAT = POSCNT)	
		0	По переднему фронту QEPx_S
		1	По переднему фронту QEPx_S при вращении по часовой стрелке или по заднему фронту QEPx_S при вращении против часовой стрелки

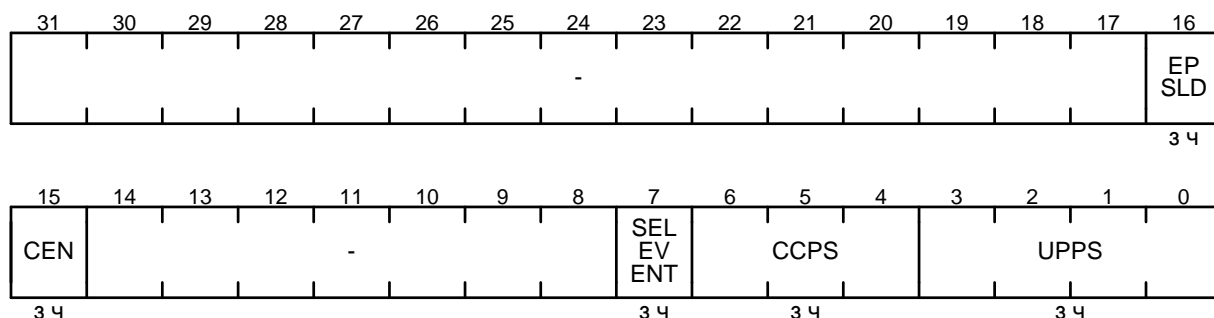
Поле	Биты	Описание
IEL	5-4	Поле задания события индексации для сохранения значения счетчика позиции (QPOSILAT = POSCNT)
		00   Без сохранения
		01   По переднему фронту сигнала индексации
		10   По заднему фронту сигнала индексации
		11   По маркеру индексации
QPEN	3	Бит разрешения работы счетчика позиции
		0   Запись нуля останавливает счетчик и сбрасывает его
		1   Работа разрешена
QCLM	2	Бит задания события сохранения значения регистров модуля захвата
		0   По чтению QPOSCNT регистры QCTMR и QCPRD сохраняются в регистры QCTMRLAT и QCPRDLAT, соответственно.
		1   По окончании временного отсчета регистры QPOSCNT, QCTMR и QCPRD сохраняются в регистры QPOSLAT, QCTMRLAT и QCPRDLAT, соответственно
UTE	1	Бит разрешения работы таймера временных отсчетов
		0   Запрещено
		1   Разрешено
WDE	0	Бит разрешения работы сторожевого таймера
		0   Запрещено
		1   Разрешено
–	31-16	Зарезервировано



## QCAPCTL – регистр управления захватом

Смещение: + 34h

Сброс: 0h

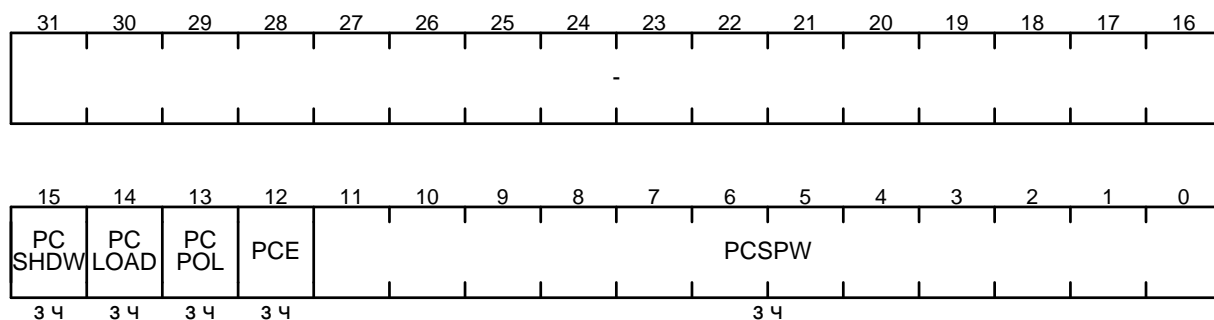


Поле	Биты	Описание
EP SLD	16	Бит включения улучшенного режима теневой загрузки UPPS/CCPS
		0   Запрещено
		1   Разрешено
CEN	15	Бит разрешения работы модуля захвата времени
		0   Запрещено
		1   Разрешено
SELEVENT	7	Бит сброса таймера
		0   По деленному квадратурному событию
		1   По получении сигнала PCSOUT от компаратора
CCPS	6-4	Поле задания делителя системного такта
		000   Нет деления
		001   1/2
		010   1/4
		011   1/8
		100   1/16
		101   1/32
		110   1/64
111   1/128		
UPPS	3-0	Поле задания делителя квадратурного сигнала
		0h   Нет деления
		1h   1/2
		2h   1/4
		3h   1/8
		4h   1/16
		5h   1/32
		6h   1/64
		7h   1/128
		8h   1/256
		9h   1/512
		Ah   1/1024
Bh   1/2048		
Ch-Fh   Зарезервировано		
–	31-16, 14-8	Зарезервировано

## QPOSCTL – регистр управления счетчиком позиции

Смещение: + 38h

Сброс: 0h

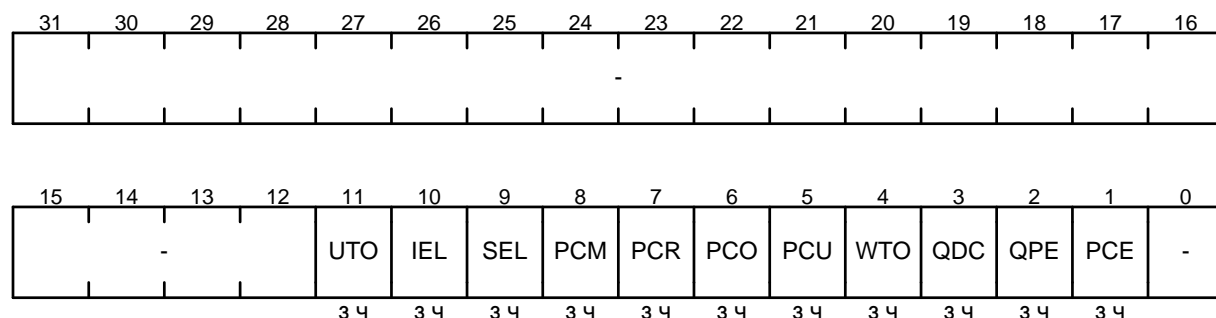


Поле	Биты	Описание
PCSHDW	15	Бит разрешения режима отложенной загрузки
		0   Запрещено
		1   Разрешено
PCLOAD	14	Бит выбора события загрузки в режиме отложенной записи
		0   Загрузка отложенного значения в активный регистр по событию QPOSCNT = 0.
		1   Загрузка по QPOSCNT = QPOSCMP
PCPOL	13	Бит выбора полярности выхода синхронизации
		0   Активная единица
		1   Активный ноль
PCE	12	Бит разрешения работы компаратора
		0   Запрещено
		1   Разрешено
PCSPW	11-0	Поле задания ширины импульса выхода синхронизации
		000h   Отсутствие импульса
		001h   $2 \times T$
		...
		007h   $8 \times T$
		...
		FFFh   $4096 \times T$
		T – период тактового сигнала PCLK
–	31-16	Зарезервировано

## QEINT – регистр масок прерываний

Смещение: + 3Ch

Сброс: 0h



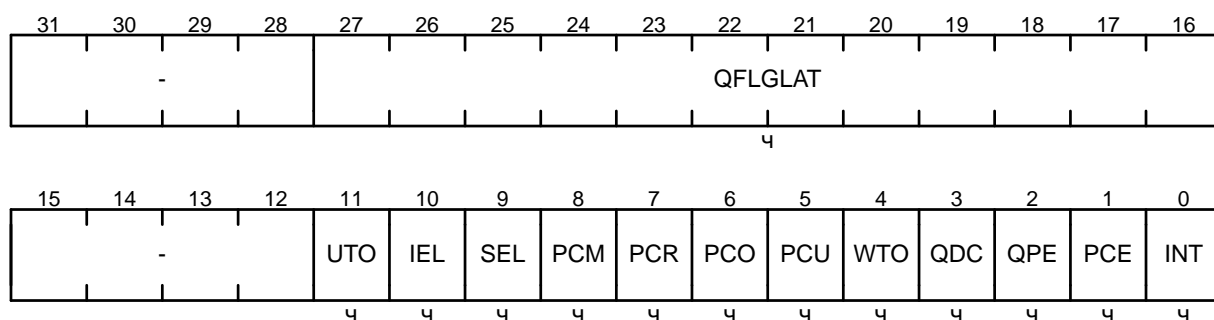
Поле	Биты	Описание
UTO	11	Бит разрешения прерывания по срабатыванию таймера временных отсчетов
IEL	10	Бит разрешения прерывания по событию индексации
SEL	9	Бит разрешения прерывания по событию стробирования
PCM	8	Бит разрешения прерывания по срабатыванию компаратора
PCR	7	Бит разрешения прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра
PCO	6	Бит разрешения прерывания по переполнению счетчиком позиции QPOSMAX при счете вверх
PCU	5	Бит разрешения прерывания по переходу счетчиком позиции через минимальное значение при счете вниз
WTO	4	Бит разрешения прерывания при срабатывании сторожевого таймера
QDC	3	Бит разрешения прерывания при смене направления вращения
QPE	2	Бит разрешения прерывания по ошибке фазы на квадратурном входе
PCE	1	Бит разрешения прерывания счетчика позиции
–	31-12, 0	Зарезервировано

Примечание – Установленный бит разрешает генерирование соответствующего прерывания, сброшенный – запрещает.

## QFLG – регистр флагов прерываний

Смещение: + 40h

Сброс: 0h



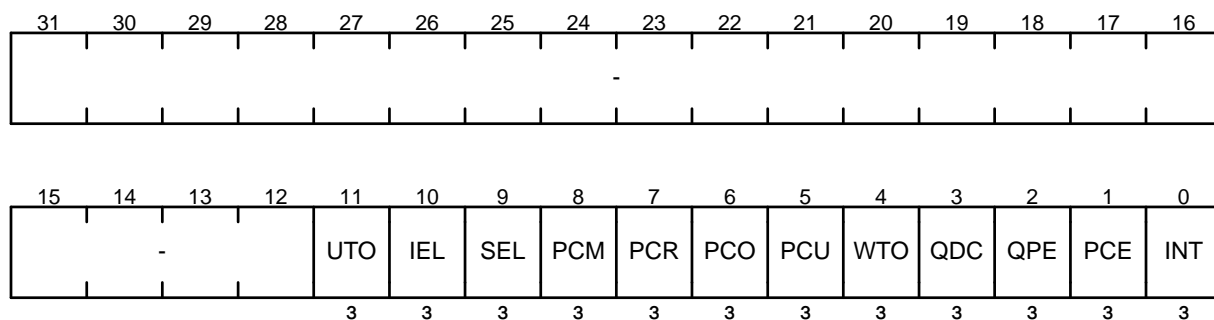
Поле	Биты	Описание
QFLGLAT	27-16	Защелкивает биты 11-0 регистра QFLG по событию чтения регистра QPOSCNT. Поле доступно только для чтения
UTO	11	Флаг прерывания по срабатыванию таймера временных отсчетов
IEL	10	Флаг прерывания по событию индексации
SEL	9	Флаг прерывания по событию стробирования
PCM	8	Флаг прерывания по срабатыванию компаратора
PCR	7	Флаг прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра
PCO	6	Флаг прерывания по переполнению счетчиком позиции QPOSMAX при счете вверх
PCU	5	Флаг прерывания по переходу счетчиком позиции через минимальное значение при счете вниз
WTO	4	Флаг прерывания при срабатывании сторожевого таймера
QDC	3	Флаг прерывания при смене направления вращения
QPE	2	Флаг прерывания по ошибке фазы на квадратурном входе
PCE	1	Флаг прерывания ошибки счетчика позиции
INT	0	Флаг выходного прерывания блока квадратурного декодера
–	31-12	Зарезервировано

Примечание – Установленный бит является индикатором запроса соответствующего прерывания. Сброс флагов прерываний осуществляется посредством регистра QCLR.

## QCLR – регистр сброса флагов прерываний

Смещение: + 44h

Сброс: 0h

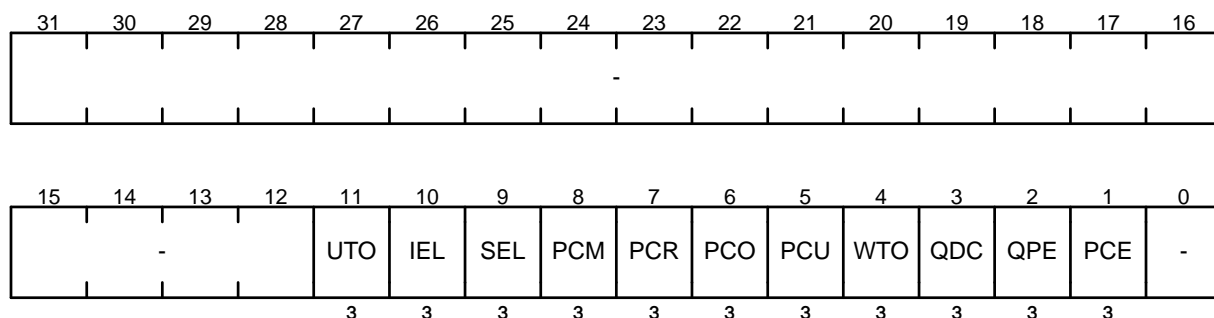


Поле	Биты	Описание
UTO	11	Запись единицы сбрасывает флаг прерывания по срабатыванию таймера временных отсчетов
IEL	10	Запись единицы сбрасывает флаг прерывания по событию индексации
SEL	9	Запись единицы сбрасывает флаг прерывания по событию стробирования
PCM	8	Запись единицы сбрасывает флаг прерывания по срабатыванию компаратора
PCR	7	Запись единицы сбрасывает флаг прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра
PCO	6	Запись единицы сбрасывает флаг прерывания по переполнению счетчиком позиции QPOSMAX при счете вверх
PCU	5	Запись единицы сбрасывает флаг прерывания по переходу счетчиком позиции через минимальное значение при счете вниз
WTO	4	Запись единицы сбрасывает флаг прерывания при срабатывании сторожевого таймера
QDC	3	Запись единицы сбрасывает флаг прерывания при смене направления вращения
QPE	2	Запись единицы сбрасывает флаг прерывания по ошибке фазы на квадратурном входе
PCE	1	Запись единицы сбрасывает флаг прерывания ошибки счетчика позиции
INT	0	Запись единицы сбрасывает флаг выходного прерывания блока квадратурного декодера
–	31-12	Зарезервировано

## QFRC – регистр эмуляции прерываний

Смещение: + 48h

Сброс: 0h

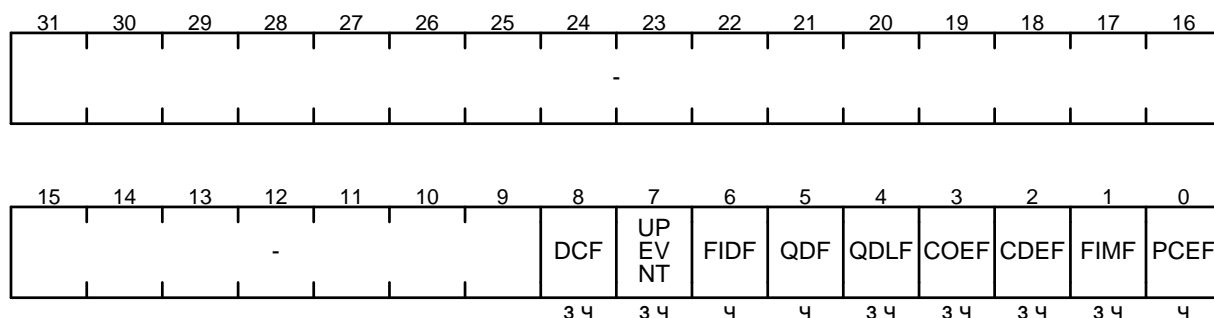


Поле	Биты	Описание
UTO	11	Запись единицы сбрасывает флаг прерывания по срабатыванию таймера временных отсчетов
IEL	10	Запись единицы сбрасывает флаг прерывания по событию индексации
SEL	9	Запись единицы сбрасывает флаг прерывания по событию стробирования
PCM	8	Запись единицы сбрасывает флаг прерывания по срабатыванию компаратора
PCR	7	Запись единицы сбрасывает флаг прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра
PCO	6	Запись единицы сбрасывает флаг прерывания по переполнению счетчиком позиции QPOSMAX при счете вверх
PCU	5	Запись единицы сбрасывает флаг прерывания по переходу счетчиком позиции через минимальное значение при счете вниз
WTO	4	Запись единицы сбрасывает флаг прерывания при срабатывании сторожевого таймера
QDC	3	Запись единицы сбрасывает флаг прерывания при смене направления вращения
QPE	2	Запись единицы сбрасывает флаг прерывания по ошибке фазы на квадратурном входе
PCE	1	Запись единицы сбрасывает флаг прерывания ошибки счетчика позиции
–	31-12, 0	Зарезервировано

## QEPSTS – регистр статуса

Смещение: + 4Ch

Сброс: 0h

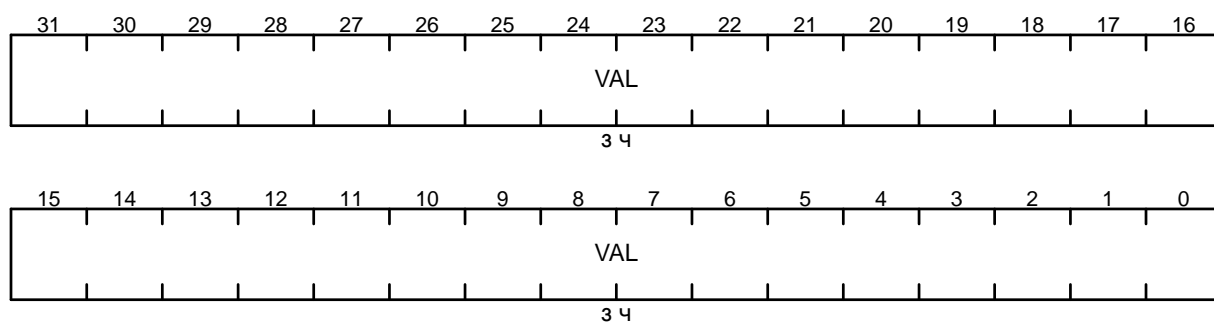


Поле	Биты	Описание
DCF	8	Флаг изменения направления вращения вала ротора. Сброс записью 1
		0   Направление не изменялось
		1   Произошло изменение направления вращения
UPEVNT	7	Флаг сброса QCTMR и обновления QCPRD. Сброс записью 1
		0   Нет событий
		1   Зафиксировано событие сброса и обновления
FIDF	6	Индикатор направления вращения по событию первого импульса индексации. Только чтение
		0   Против часовой стрелки (счет вниз)
		1   По часовой стрелке (счет вверх)
QDF	5	Флаг направления вращения. Обновляется по каждому событию на входах квадратур. Только чтение
		0   Вращение вала ротора против часовой стрелки
		1   Вращение вала ротора по часовой стрелке.
QDLF	4	Флаг направления вращения. Обновляется по каждому сигналу индексации. Только чтение
		0   Вращение вала ротора против часовой стрелки
		1   Вращение вала ротора по часовой стрелке.
COEF	3	Флаг ошибки переполнения счетчика QCTMR модуля захвата. Сброс записью 1
		0   Ошибка отсутствует
		1   Произошло переполнение
CDEF	2	Флаг ошибки изменения направления вращения вала ротора между двумя событиями UPEVNT. Сброс записью 1
		0   Ошибка отсутствует
		1   Произошло изменение направления вращения во время измерения
FIMF	1	Флаг приема первого импульса сигнала индексации. Сброс записью 1
		0   Импульсов нет, либо первый импульс уже был принят
		1   Принят первый импульс сигнала индексации
PCEF	0	Флаг ошибки счетчика позиции. Обновляется по каждому сигналу индексации. Только чтение
		0   Во время последнего сигнала индексации ошибки не возникло
		1   Ошибка счетчика позиции
–	31-9	Зарезервировано

### QCTMR – регистр таймера блока захвата

Смещение: + 50h

Сброс: 0h

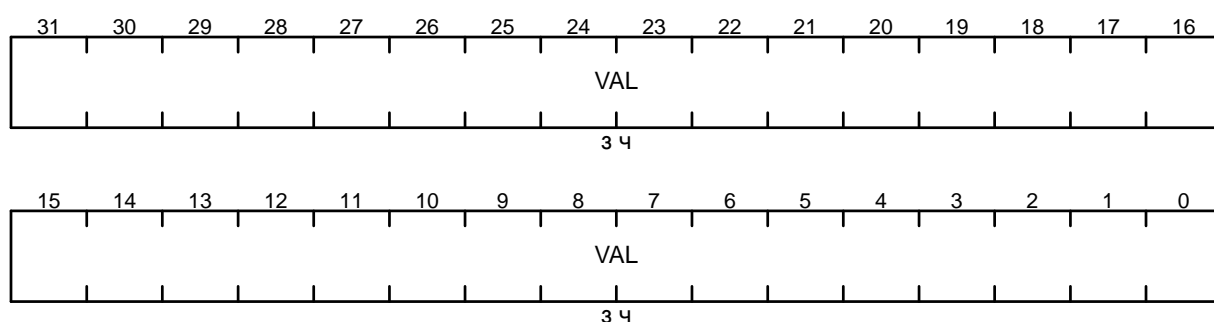


Поле	Биты	Описание
VAL	31-0	Значение таймера блока захвата

### QCPRD – регистр длительности измерения блока захвата

Смещение: + 54h

Сброс: 0h

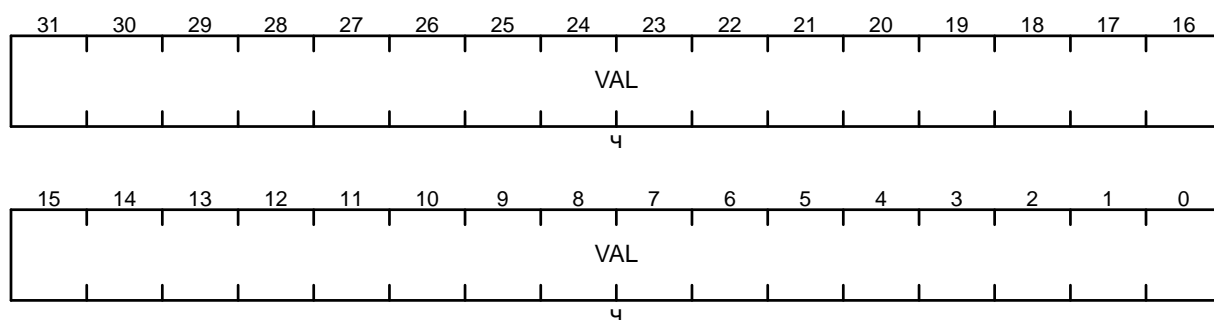


Поле	Биты	Описание
VAL	31-0	Значение длительности измерения блока захвата

### QCTMRLAT – регистр хранения таймера блока захвата

Смещение: + 58h

Сброс: 0h



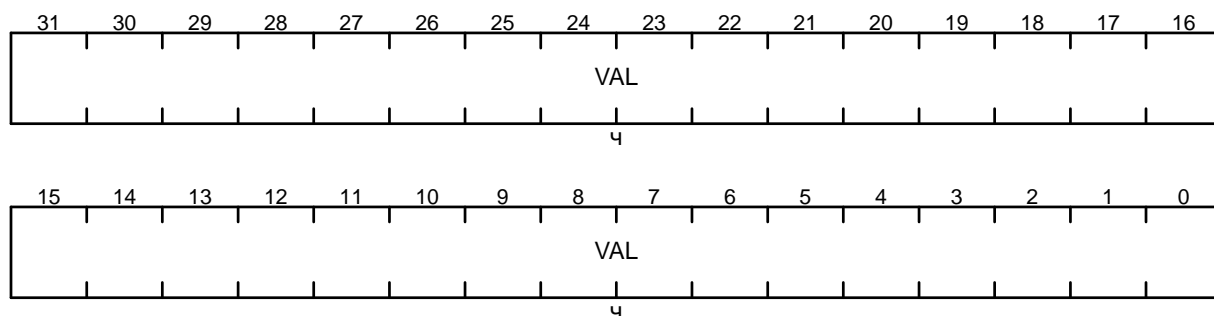
Поле	Биты	Описание
VAL	31-0	Значение хранения таймера блока захвата



### QCPRDLAT – регистр хранения длительности измерения блока захвата

Смещение: + 5Ch

Сброс: 0h

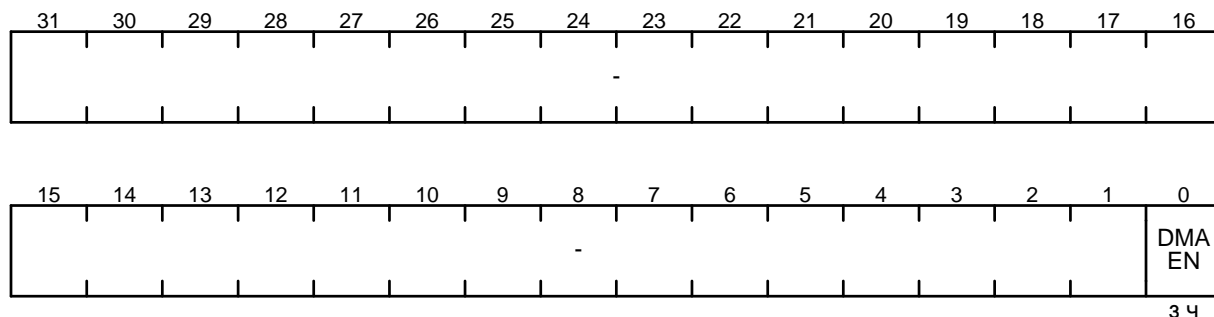


Поле	Биты	Описание
VAL	31-0	Значение хранения длительности измерения блока захвата

### DMAREQ – регистр управления запросом DMA

Смещение: + 60h

Сброс: 0h

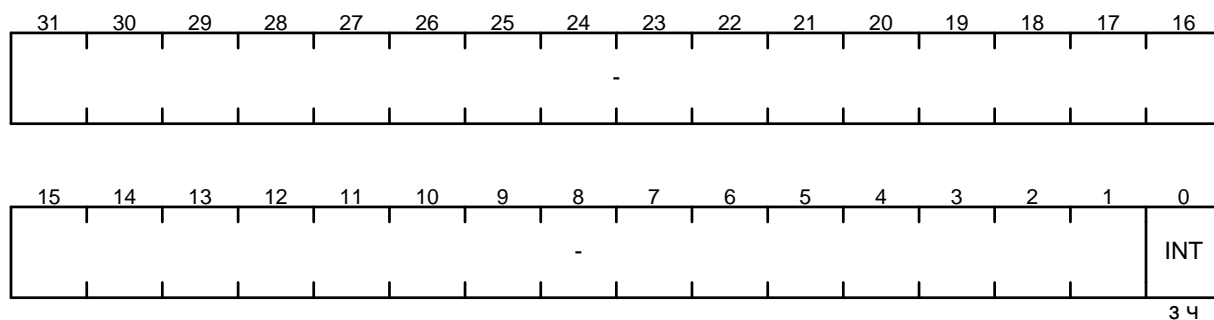


Поле	Биты	Описание	
DMAEN	0	Разрешение генерации запроса DMA. Запрос генерируется каждый раз, когда флаг UPEVNT (регистра EPSTS) переходит из 0 в 1. Если UPEVNT будет оставаться несброшенным, то запросы генерироваться не будут. При этом когда происходит чтение QCPRD при DMAEN = 1, флаг UPEVNT сбрасывается автоматически	
		0	Нет запроса
		1	Запрос разрешен
–	31-1	Зарезервировано	

## INTCLR – регистр сброса прерывания

Смещение: + 70h

Сброс: 0h



Поле	Биты	Описание
INT	0	Бит сброса прерывания. Чтение возвращает текущий статус активности прерывания. Запись единицы в бит сбрасывает запрос прерывания. Активный запрос способен вызывать повторные запуски обработчика прерывания.
–	31-1	Зарезервировано

## A.19 Регистры сигма-дельта демодулятора SDFM

**Базовый адрес:** 4007\_5000h

**Смещение:** + 18h (CH0)                      Смещение регистров канала 0  
 + 30h (CH1)                      Смещение регистров канала 1  
 + 48h (CH2)                      Смещение регистров канала 2  
 + 60h (CH3)                      Смещение регистров канала 3

**Мнемоника:** CHx

Примечание – x – номер канала 0 - 3

### IFLG – регистр флагов прерываний

Смещение: + 00h

Сброс: 0h

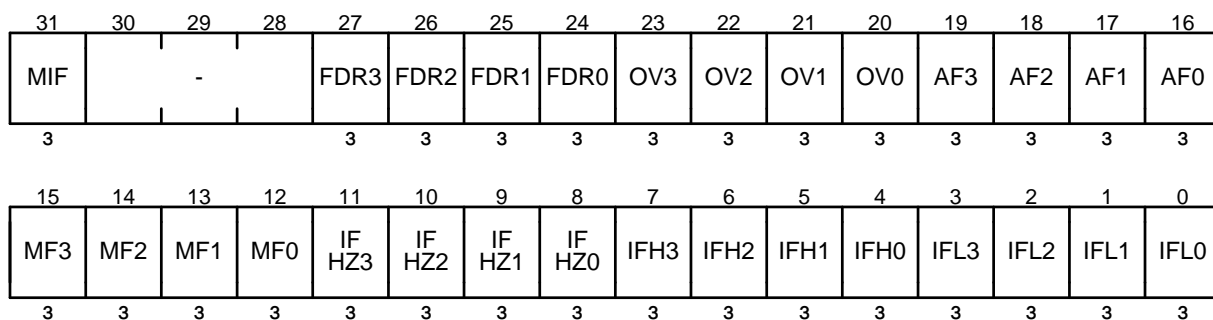
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIF		-		FDR3	FDR2	FDR1	FDR0	OV3	OV2	OV1	OV0	AF3	AF2	AF1	AF0
ч				ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MF3	MF2	MF1	MF0	IF HZ3	IF HZ2	IF HZ1	IF HZ0	IFH3	IFH2	IFH1	IFH0	IFL3	IFL2	IFL1	IFL0
ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч	ч

Поле	Биты	Описание
MIF	31	Флаг, объединяющий по ИЛИ флаги: OVx, MFx, IFHx, IFLx
FDRx	27-24	Флаг прерывания по уровню заполнения FIFO фильтра данных x
OVx	23-20	Флаг прерывания по переполнению FIFO фильтра данных x
AFx	19-16	Флаг прерывания по появлению новых результатов фильтра данных x
MFx	15-12	Флаг прерывания по сбою модулятора на входном декодере x
IFHZx	11-8	Флаг прерывания компаратора x по условию «Результат ≥ CMPHZ»
IFHx	7-4	Флаг прерывания компаратора x по условию «Результат ≥ CMPH»
IFLx	3-0	Флаг прерывания компаратора x по условию «Результат ≤ CMPL»
–	30-28	Зарезервировано

## IFLGCLR – регистр очистки флагов прерываний

Смещение: + 04h

Сброс: 0h

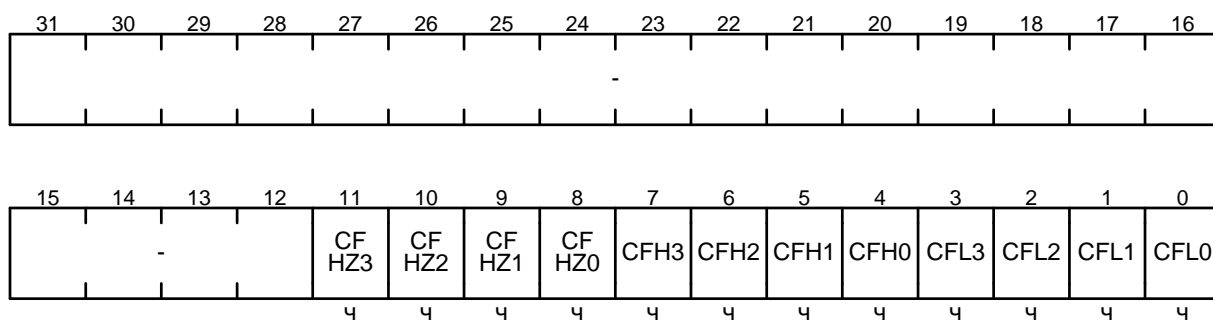


Поле	Биты	Описание
MIF	31	Установка бита очистит флаг MIF в регистре IFLG
FDR <sub>x</sub>	27-24	Установка бита очистит флаг FDR <sub>x</sub> в регистре IFLG
OV <sub>x</sub>	23-20	Установка бита очистит флаг OV <sub>x</sub> в регистре IFLG
AF <sub>x</sub>	19-16	Установка бита очистит флаг AF <sub>x</sub> в регистре IFLG
MF <sub>x</sub>	15-12	Установка бита очистит флаг MF <sub>x</sub> в регистре IFLG
IFHZ <sub>x</sub>	11-8	Установка бита очистит флаг IFHZ <sub>x</sub> в регистре IFLG
IFH <sub>x</sub>	7-4	Установка бита очистит флаг IFH <sub>x</sub> в регистре IFLG
IFL <sub>x</sub>	3-0	Установка бита очистит флаг IFL <sub>x</sub> в регистре IFLG
–	30-28	Зарезервировано

## CFLG - регистр флагов компараторов

Смещение: + 08h

Сброс: 0h

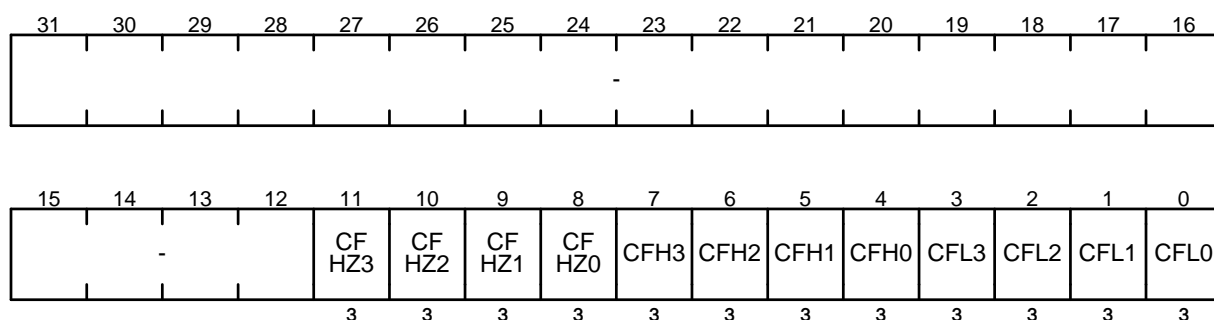


Поле	Биты	Описание
CFHZ <sub>x</sub>	11-8	Флаг срабатывания компаратора x по условию «Результат ≥ CMPHZ»
CFH <sub>x</sub>	7-4	Флаг срабатывания компаратора x по условию «Результат ≥ CMPH»
CFL <sub>x</sub>	3-0	Флаг срабатывания компаратора x по условию «Результат ≤ CMPL»
–	31-12	Зарезервировано

## CFLGCLR - регистр очистки флагов компараторов

Смещение: + 0Ch

Сброс: 0h

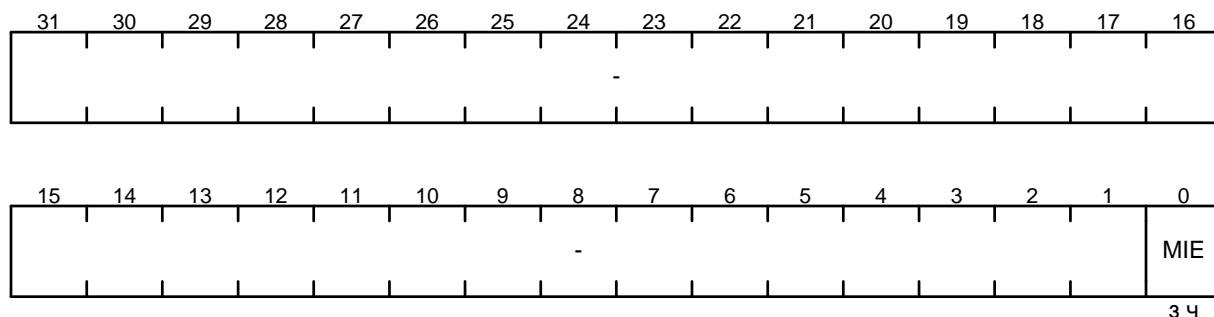


Поле	Биты	Описание
CFHZx	11-8	Установка бита очистит флаг CFHZx в регистре CFLG
CFHx	7-4	Установка бита очистит флаг CFHx в регистре CFLG
CFLx	3-0	Установка бита очистит флаг CFLx в регистре CFLG
–	31-12	Зарезервировано

## CTL - регистр управления

Смещение: +10h

Сброс: 0h

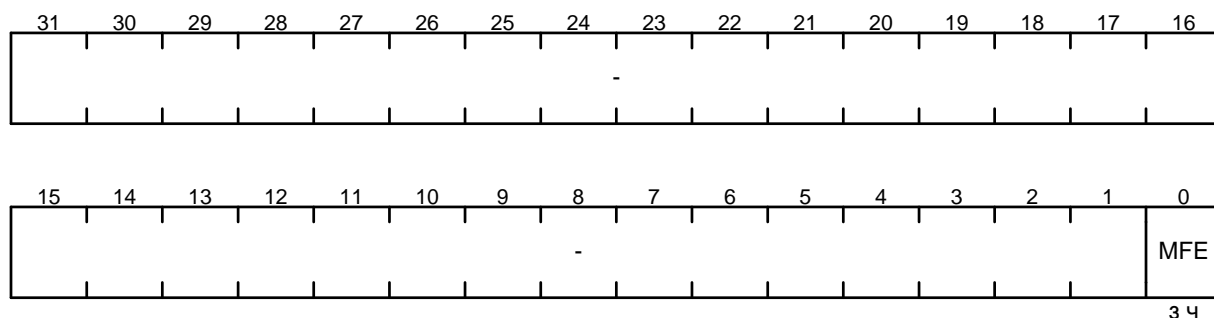


Поле	Биты	Описание
MIE	0	Глобальное разрешение генерации прерываний сигма-дельта демодулятора
–	31-1	Зарезервировано

## MFILEN – регистр разрешения работы фильтров

Смещение: + 14h

Сброс: 0h

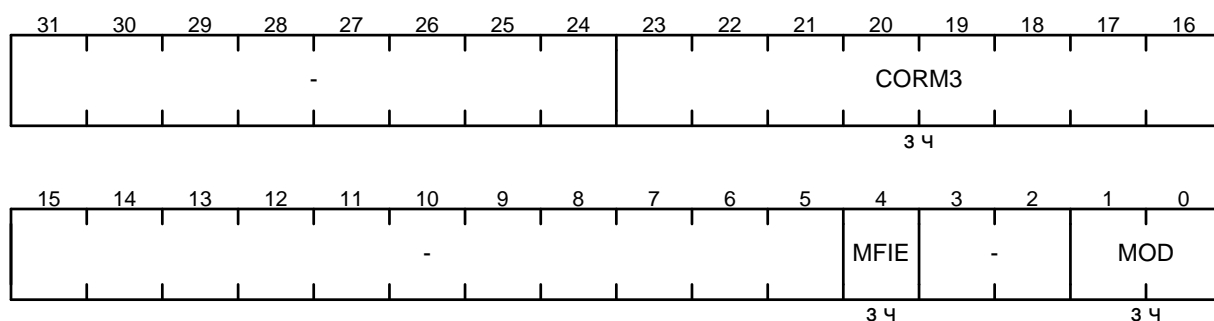


Поле	Биты	Описание
MFE	0	Глобальное разрешение работы каналов фильтрации
–	31-1	Зарезервировано

## CTLPARM - регистр настройки входного декодера канала

Смещение: CHx + 00h

Сброс: 0h

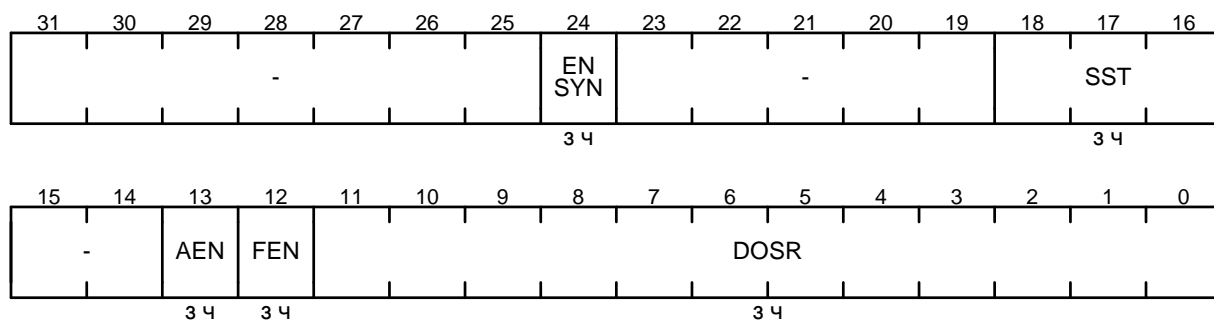


Поле	Биты	Описание
CORM3	23-16	Делитель, определяющий частоту генерации для режима 3 модулятора (MOD=11b). Делитель принимает значения от 0 до 255, и результирующий коэффициент деления тактового сигнала АНВ определяется по формуле $2 \times (CORM3 + 1)$ .
MFIE	4	Бит разрешения прерывания по сбю модулятора
		0   Прерывание отключено
		1   Прерывание включено
MOD	1-0	Выбор режима декодирования
		00b   Режим 0. Частота SDFMx_CLK совпадает со скоростью выставления данных SDFMx_DATA.
		01b   Режим 1. Частота SDFMx_CLK в два раза меньше скорости выставления данных SDFMx_DATA.
		10b   Режим 2. Манчестерское кодирование данных.
		11b   Режим 3. Тактирование от демодулятора.
–	31-24, 15-5, 3-2	Зарезервировано

## DFPARM – регистр настройки фильтра данных канала

Смещение: CHx + 04h

Сброс: 0h

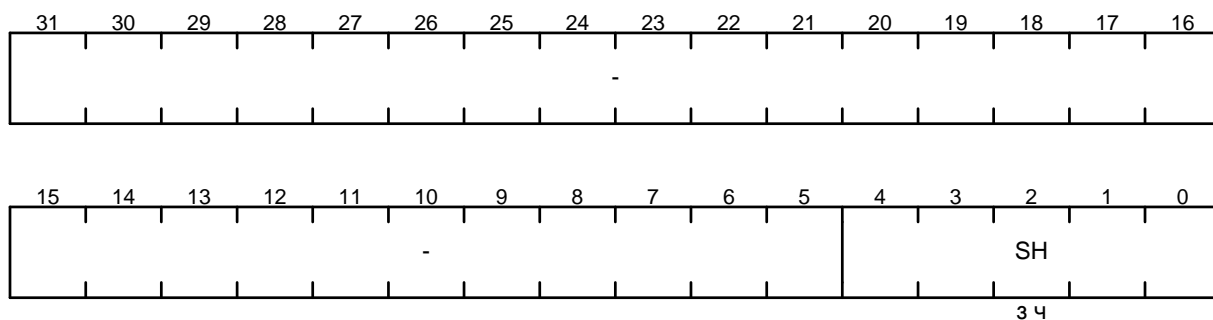


Поле	Биты	Описание	
ENSYN	24	Бит разрешения синхронизации фильтра данных	
		0	Синхронизация отключена
		1	Синхронизация включена
SST	18-16	Выбор структуры фильтра данных	
		0h	Fastsinc
		1h	Sinc1
		2h	Sinc2
		3h	Sinc3
		4h	Sinc4
		5h	Sinc5
AEN	13	Бит разрешения прерывания по появлению новых данных	
		0	Прерывание отключено
		1	Прерывание включено
FEN	12	Бит разрешения работы фильтра	
		0	Выключен
		1	Включен
DOSR	11-0	Поле, задающее величину коэффициента децимации от 0 до 4095. Реально действующее значение на единицу больше.	
–	31-25, 23-19, 15-14	Зарезервировано	

### DPARM – регистр настройки сдвига фильтра данных канала

Смещение: CHx + 08h

Сброс: 0h

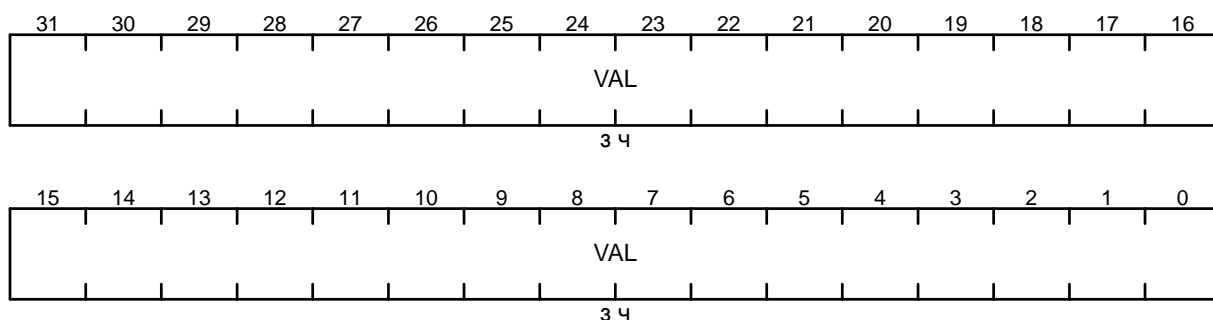


Поле	Биты	Описание
SH	4-0	Поле, задающее величину сдвига данных вправо от 1 до 23. При значении 0 сдвиг отключен.
–	31-5	Зарезервировано

### СМРН – регистр настройки верхнего порога компаратора канала

Смещение: CHx + 0Ch

Сброс: 0h



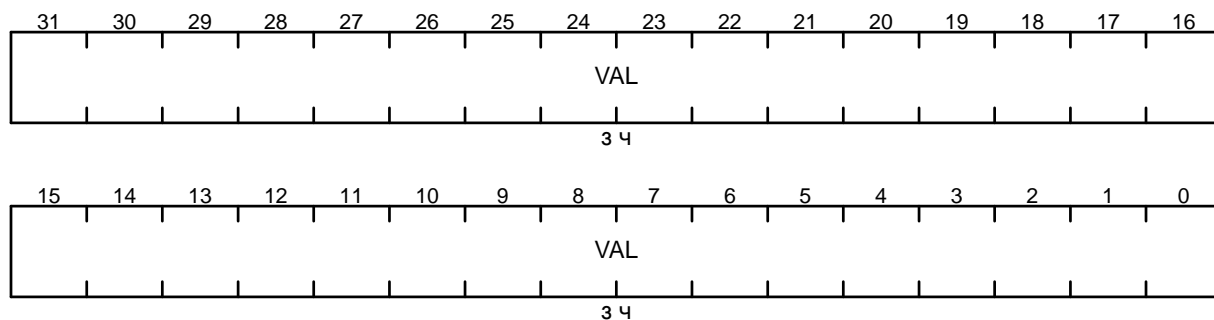
Поле	Биты	Описание
VAL	31-0	Значение верхнего порога компаратора



### СМРHZ – регистр настройки порога нуля компаратора канала

Смещение: CHx + 10h

Сброс: 0h

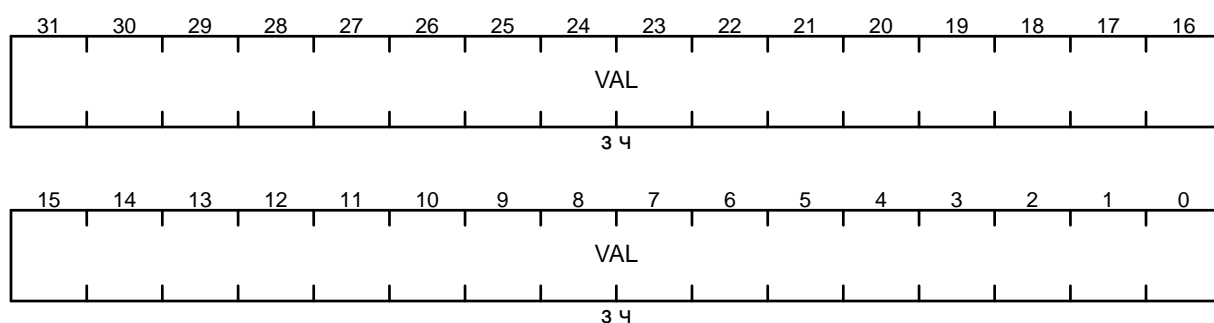


Поле	Биты	Описание
VAL	31-0	Значение порога нуля компаратора

### СМРPL – регистр настройки нижнего порога компаратора канала

Смещение: CHx + 14h

Сброс: 0h

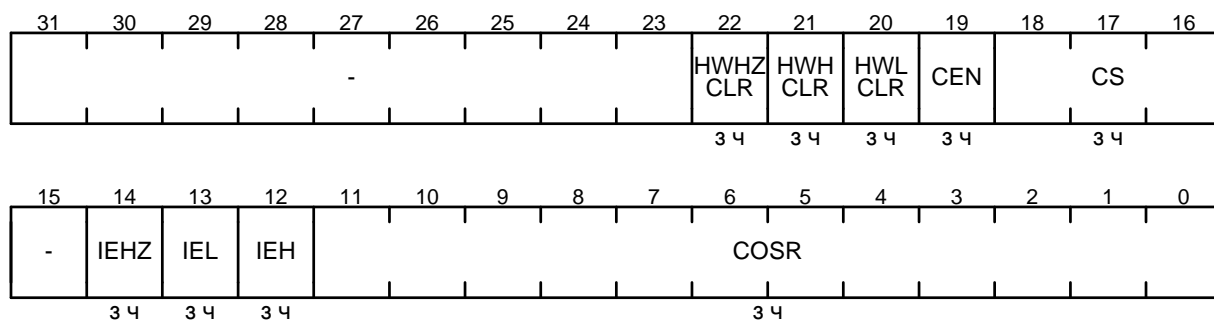


Поле	Биты	Описание
VAL	31-0	Значение нижнего порога компаратора

## CPARM – регистр настройки компаратора канала

Смещение: CHx + 18h

Сброс: 0h



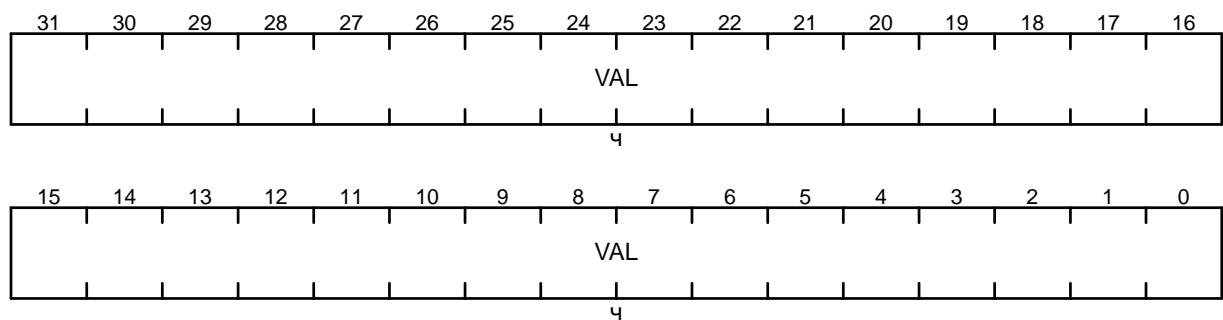
Поле	Биты	Описание	
HWHZCLR	22	Бит разрешения аппаратной очистки флага CFHZx регистра CFLG по ближайшему отрицательному сравнению	
		0	Очистка отключена
		1	Очистка включена
HWHCLR	21	Бит разрешения аппаратной очистки флага CFHx регистра CFLG по ближайшему отрицательному сравнению	
		0	Очистка отключена
		1	Очистка включена
HWLCLR	20	Бит разрешения аппаратной очистки флага CFLx регистра CFLG по ближайшему отрицательному сравнению	
		0	Очистка отключена
		1	Очистка включена
CEN	19	Бит разрешения работы компаратора	
		0	Выключен
		1	Включен
CS	18-16	Выбор структуры фильтра компаратора	
		0h	Fastsinc
		1h	Sinc1
		2h	Sinc2
		3h	Sinc3
		4h	Sinc4
		5h	Sinc5
IENZ	14	Бит разрешения прерывания компаратора по условию «Результат ≥ CMPHZ»	
		0	Прерывание отключено
		1	Прерывание включено
IEN	13	Бит разрешения прерывания компаратора по условию «Результат ≥ CMPH»	
		0	Прерывание отключено
		1	Прерывание включено
IEL	12	Бит разрешения прерывания компаратора по условию «Результат ≤ CMPL»	
		0	Прерывание отключено
		1	Прерывание включено

Поле	Биты	Описание
COSR	11-0	Поле, задающее величину коэффициента децимации от 0 до 4095. Реально действующее значение на единицу больше.
–	31-23, 15	Зарезервировано

### DATA – регистр FIFO фильтра данных канала

Смещение: CHx + 1Ch

Сброс: 0h

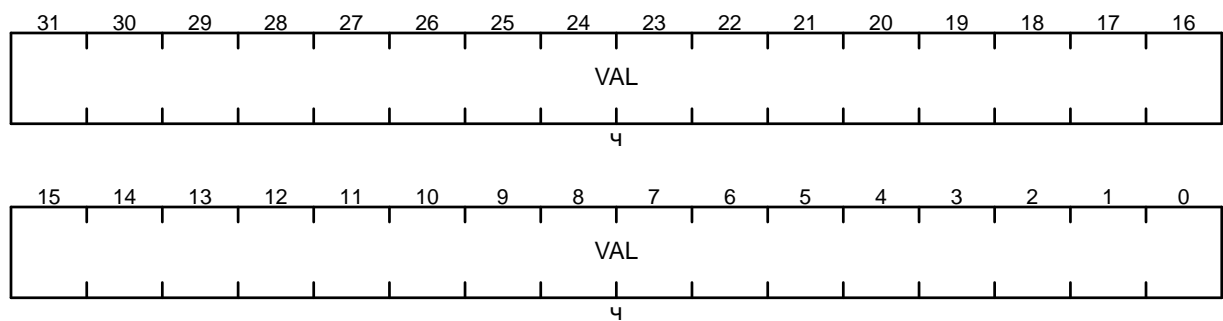


Поле	Биты	Описание
VAL	31-0	Значение результата фильтрации

### CDATA – регистр данных компаратора канала

Смещение: CHx + 20h

Сброс: 0h

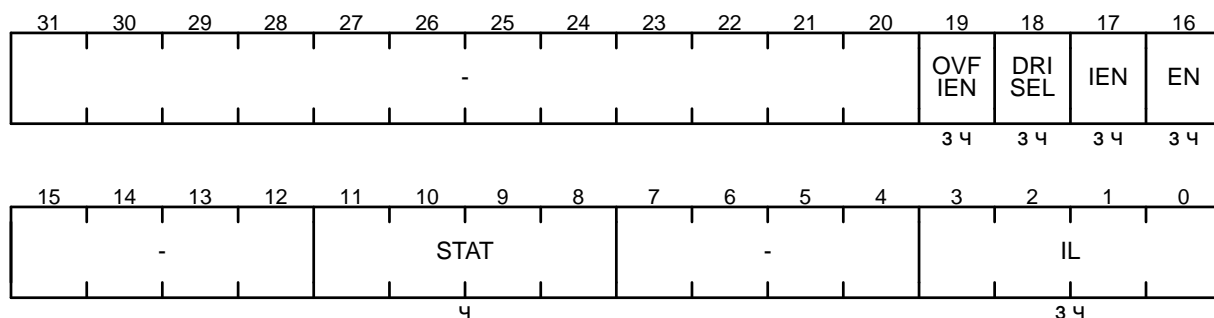


Поле	Биты	Описание
VAL	31-0	Значение данных, использованных в последнем сравнении

## FFCTL – регистр настройки FIFO фильтра данных канала

Смещение: CHx + 24h

Сброс: 0h

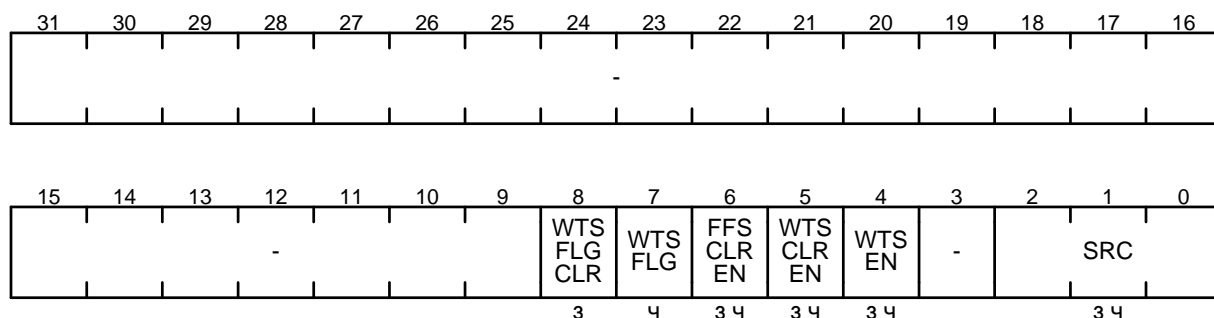


Поле	Биты	Описание	
OVFIEN	19	Бит разрешения прерывания по переполнению FIFO	
		0	Прерывание отключено
		1	Прерывание включено
DRISEL	18	Бит выбора источника прерываний фильтра данных	
		0	Появление новых результатов на выходе фильтра
		1	Уровень заполнения FIFO
IEN	17	Бит разрешения прерывания фильтра данных	
		0	Прерывание отключено
		1	Прерывание включено
EN	16	Бит разрешения работы FIFO	
		0	Отключено
		1	Включено
STAT	11-8	Состояние наполненности FIFO. Чтение поля возвращает значения от 0 до 8.	
IL	3-0	Настройка уровня наполнения FIFO для генерации прерывания. Возможна запись значений от 0 до 8. Когда $STAT \geq IL$ , будет вызвано прерывание.	
–	31-20, 15-12, 7-4	Зарезервировано	

## SYNC – регистр настройки синхронизации фильтра данных канала

Смещение: CHx + 28h

Сброс: 0000\_0020h

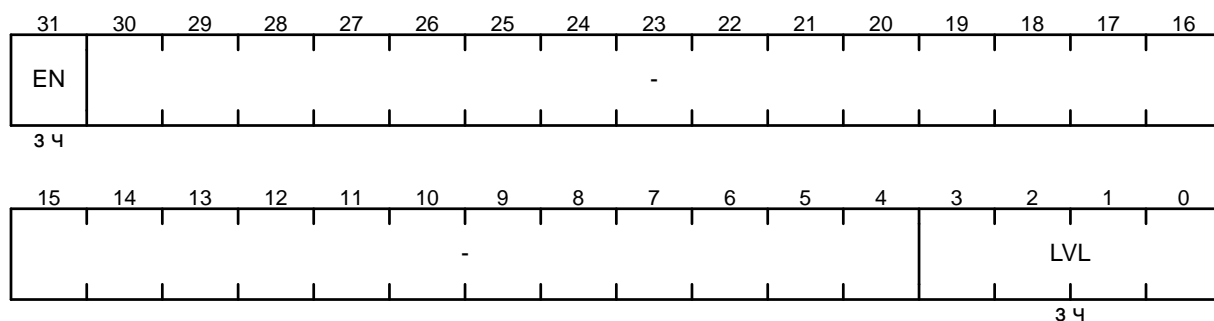


Поле	Биты	Описание
WTSFLGCLR	8	Установка бита приводит к сбросу флага WTSFLG
WTSFLG	7	Флаг события синхронизации SDSYNC
		0   Флаг сброшен. Данные в буфер не поступают. 1   Произошло событие синхронизации. Данные сохраняются в FIFO до тех пор, пока флаг установлен.
FFSCLREN	6	Бит разрешения автоматической очистке FIFO по событию синхронизации SDSYNC
		0   Очистка отключена 1   Очистка включена
WTSCLREN	5	Бит разрешения автоматической очистки флага WTSFLG по установке соответствующего флага прерывания FDRx в регистре IFLG. Включено по умолчанию.
		0   Отключено 1   Включено
WTS EN	4	Бит разрешения функции ожидания синхронизации SDSYNC
		0   Отключена 1   Включена
SRC	2-0	Настройка источника события синхронизации
		0h   Сигналы SOC от блоков PWM0, PWM1, PWM2 – канал А
		1h   Сигналы SOC от блоков PWM0, PWM1, PWM2 – канал В
		2h   Сигналы SOC от блоков PWM3, PWM4, PWM5 – канал А
		3h   Сигналы SOC от блоков PWM3, PWM4, PWM5 – канал В
		4h   Сигналы SOC от блоков PWM6, PWM7, PWM8, PWM9 – канал А
		5h   Сигналы SOC от блоков PWM6, PWM7, PWM8, PWM9 – канал В
		6h   Сигнал TMR0_ADCSOC
7h   Сигнал TMR1_ADCSOC		
–	31-9, 3	Зарезервировано

## DMACTL - регистр настройки запросов DMA фильтра данных канала

Смещение: CHx + 2Ch

Сброс: 0h



Поле	Биты	Описание	
EN	31	Бит разрешения генерации запросов DMA	
		0	Отключено
		1	Включено
LVL	3-0	Настройка уровня наполнения FIFO для генерации запроса DMA. Возможна запись значений от 1 до 8. Когда $STAT \geq LVL$ , начнется передача LVL слов данных.	
–	30-4	Зарезервировано	

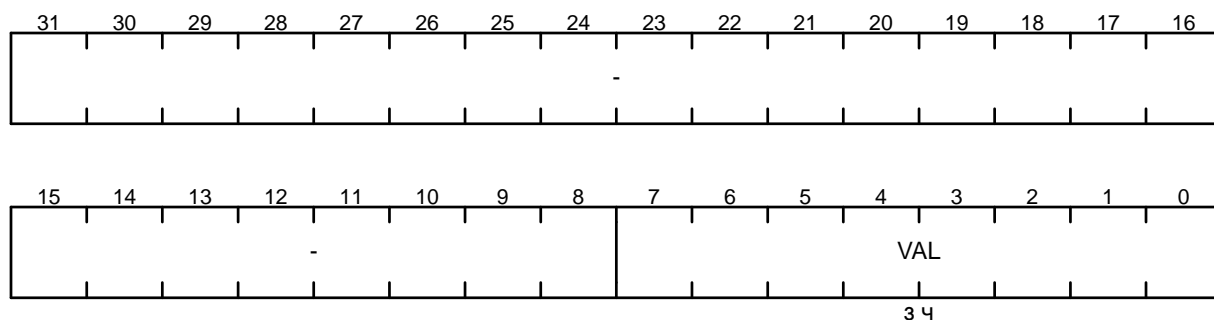
## A.20 Регистры приемопередатчика UART

Базовый адрес: 4008\_9000h      Регистры блока UART 0  
 4008\_A000h      Регистры блока UART 1

### DATA – регистр данных

Смещение: + 00h

Сброс: 0h

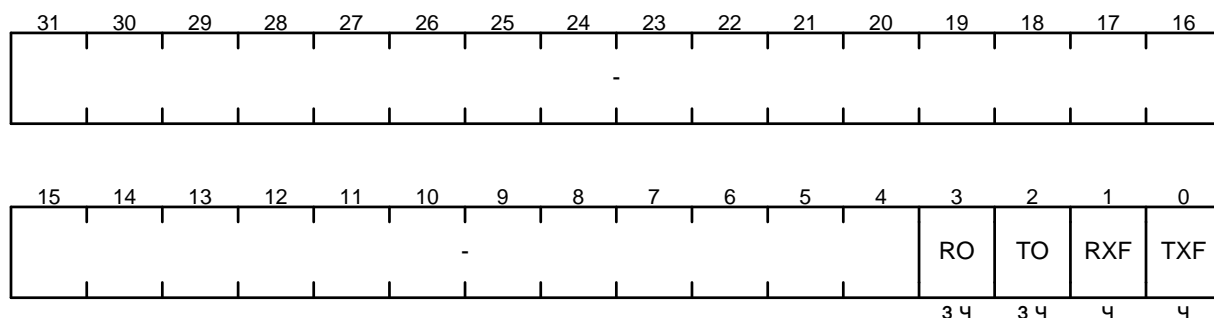


Поле	Биты	Описание
VAL	7-0	Поле данных. Результатом записи в поле DATA является размещение байта в буфере передатчика, а результатом чтения – считывание байта из буфера приемника
–	31-8	Зарезервировано

## STATE – регистр статуса

Смещение: + 04h

Сброс: 0h



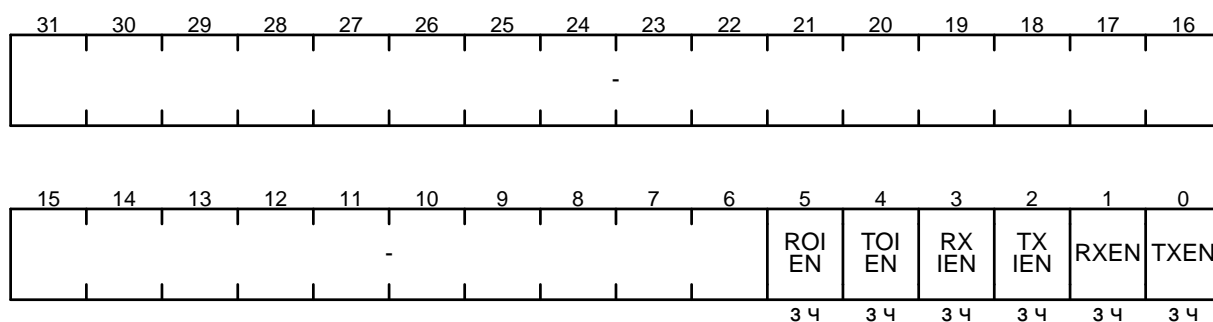
Поле	Биты	Описание
RO	3	Флаг переполнения буфера приемника
		0   Буфер не переполнен
		1   Произошла попытка сохранить данные в полный буфер Сбрасывается записью единицы
TO	2	Флаг переполнения буфера передатчика
		0   Буфер не переполнен
		1   Произошла попытка записать данные в полный буфер Сбрасывается записью единицы
RXF	1	Флаг заполнения буфера приемника
		0   Буфер пуст 1   Буфер заполнен
TXF	0	Флаг заполнения буфера передатчика
		0   Буфер пуст 1   Буфер заполнен
–	31-4	Зарезервировано



## CTRL – регистр управления

Смещение: + 08h

Сброс: 0h

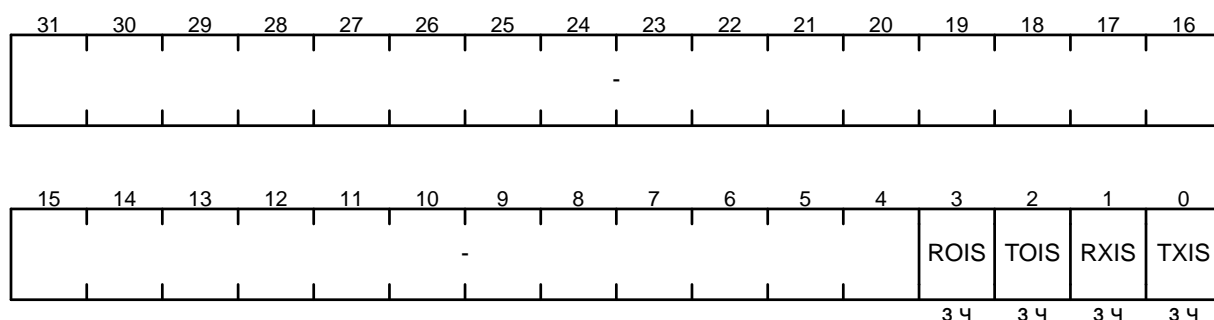


Поле	Биты	Описание
ROIEN	5	Бит включения прерывания по переполнению буфера приемника
		0 Прерывание выключено
		1 Прерывание включено
TOIEN	4	Бит включения прерывания по переполнению буфера передатчика
		0 Прерывание выключено
		1 Прерывание включено
RXIEN	3	Бит включения прерывания по приему
		0 Прерывание выключено
		1 Прерывание включено
TXIEN	2	Бит включения прерывания по передаче
		0 Прерывание выключено
		1 Прерывание включено
RXEN	1	Бит включения приемника
		0 Приемник выключен
		1 Приемник включен
TXEN	0	Бит включения передатчика
		0 Передатчик выключен
		1 Передатчик включен
–	31-6	Зарезервировано

## INTSTATUS – регистр флагов прерываний

Смещение: + 0Ch

Сброс: 0h



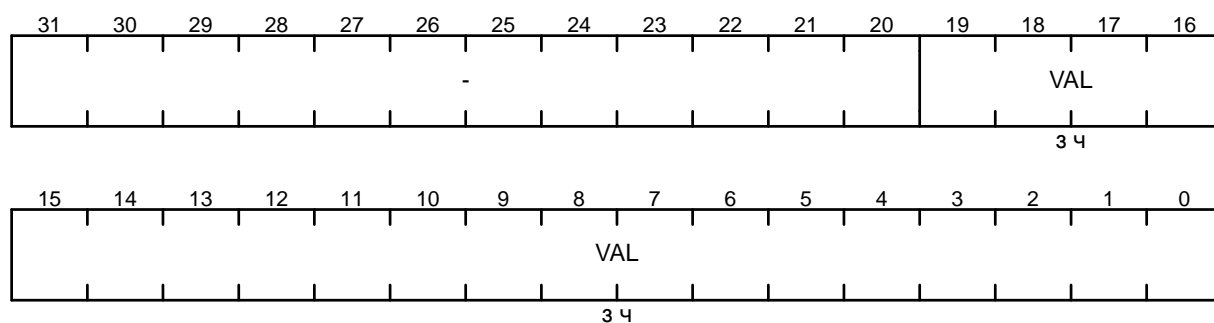
Поле	Биты	Описание
RO	3	Флаг прерывания по переполнению буфера приемника
TO	2	Флаг прерывания по переполнению буфера передатчика
RXF	1	Флаг прерывания по приему
TXF	0	Флаг прерывания по передаче
–	31-4	Зарезервировано

Примечание – Флаги сбрасываются записью единицы

## BAUDDIV – регистр настройки делителя тактового сигнала

Смещение: + 10h

Сброс: 0h



Поле	Биты	Описание
VAL	19-0	Установка делителя тактового сигнала для формирования необходимой скорости передачи. Минимальное значение - 16.
–	31-20	Зарезервировано

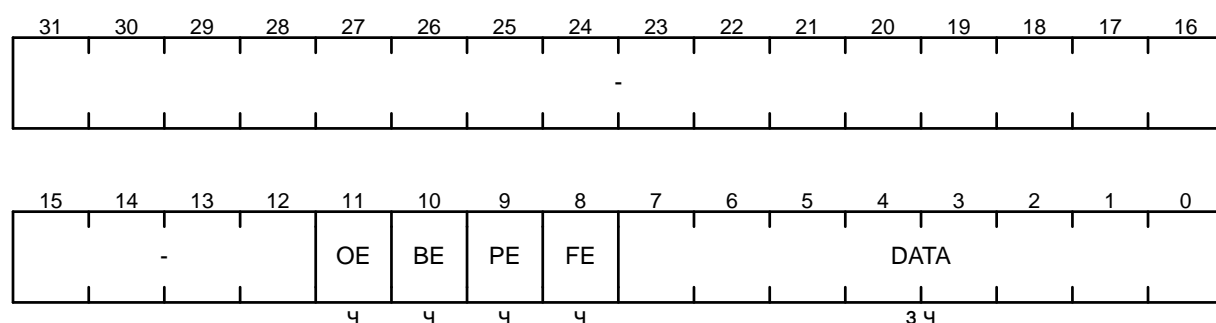
## A.21 Регистры приемопередатчика UART

<b>Базовый адрес:</b> 4008_5000h	Регистры приемопередатчика UART0
4008_6000h	Регистры приемопередатчика UART1
4008_7000h	Регистры приемопередатчика UART1
4008_8000h	Регистры приемопередатчика UART1

### DR – регистр данных

Смещение: + 00h

Сброс: 0h

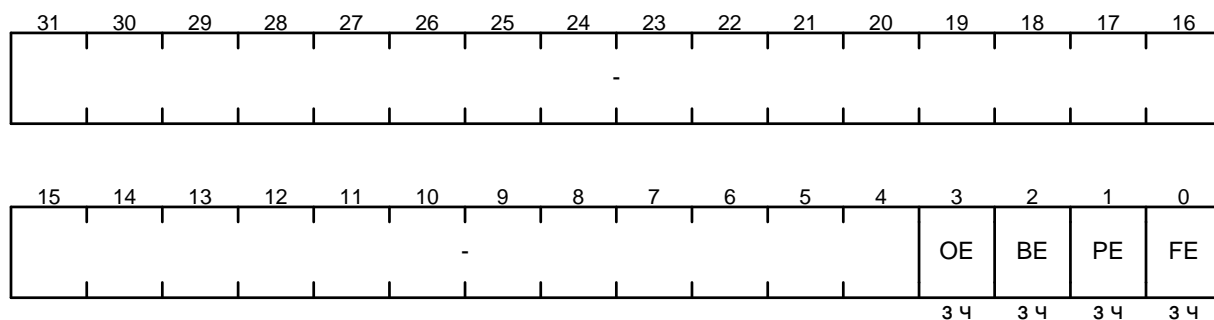


Поле	Биты	Описание
OE, BE, PE, FE	11, 10, 9, 8	См. описание бит в регистре RSR
DATA	7-0	Поле данных. Результатом записи в поле DATA является размещение байта в буфере передатчика, а результатом чтения – считывание байта из буфера приемника
–	31-12	Зарезервировано

## RSR – регистр состояния приемника и сброса ошибки приемника

Смещение: + 04h

Сброс: 0h



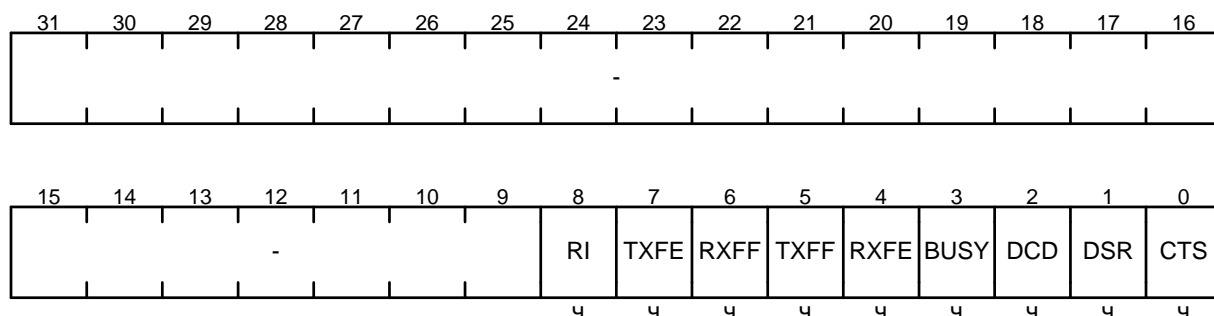
Поле	Биты	Описание		
OE	3	<b>Флаг переполнения буфера приемника</b>		
		<table border="1"> <tr> <td>0</td> <td>В буфере есть свободное место или бит был сброшен после записи в регистр RSR. Содержимое буфера остается верным, так как перезаписан был только сдвиговый регистр. Центральный процессор должен считать данные для того, чтобы освободить буфер</td> </tr> <tr> <td>1</td> <td>Буфер заполнен, а данные продолжают поступать</td> </tr> </table>	0	В буфере есть свободное место или бит был сброшен после записи в регистр RSR. Содержимое буфера остается верным, так как перезаписан был только сдвиговый регистр. Центральный процессор должен считать данные для того, чтобы освободить буфер
0	В буфере есть свободное место или бит был сброшен после записи в регистр RSR. Содержимое буфера остается верным, так как перезаписан был только сдвиговый регистр. Центральный процессор должен считать данные для того, чтобы освободить буфер			
1	Буфер заполнен, а данные продолжают поступать			
BE	2	<b>Флаг разрыва линии</b>		
		<table border="1"> <tr> <td>0</td> <td>Нормальная работа или бит был сброшен после записи в регистр RSR</td> </tr> <tr> <td>1</td> <td>Обнаружен признак разрыва линии, то есть наличие низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного кадра данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном режиме FIFO данная ошибка ассоциируется с последним байтом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой кадр. Прием данных возобновляется только после перехода линии в логическую единицу и последующего обнаружения корректного стартового бита</td> </tr> </table>	0	Нормальная работа или бит был сброшен после записи в регистр RSR
0	Нормальная работа или бит был сброшен после записи в регистр RSR			
1	Обнаружен признак разрыва линии, то есть наличие низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного кадра данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном режиме FIFO данная ошибка ассоциируется с последним байтом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой кадр. Прием данных возобновляется только после перехода линии в логическую единицу и последующего обнаружения корректного стартового бита			
PE	1	<b>Флаг ошибки контроля четности</b>		
		<table border="1"> <tr> <td>0</td> <td>Нормальная работа или бит был сброшен после записи в регистр RSR</td> </tr> <tr> <td>1</td> <td>Четность принятого кадра данных не соответствует установкам битов EPS и SPS в регистре управления линией LCRH. При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера</td> </tr> </table>	0	Нормальная работа или бит был сброшен после записи в регистр RSR
0	Нормальная работа или бит был сброшен после записи в регистр RSR			
1	Четность принятого кадра данных не соответствует установкам битов EPS и SPS в регистре управления линией LCRH. При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера			
FE	0	<b>Флаг ошибки в структуре кадра</b>		
		<table border="1"> <tr> <td>0</td> <td>Нормальная работа или бит был сброшен после записи в регистр RSR</td> </tr> <tr> <td>1</td> <td>В принятом символе не обнаружен корректный стоповый бит (единица). При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера</td> </tr> </table>	0	Нормальная работа или бит был сброшен после записи в регистр RSR
0	Нормальная работа или бит был сброшен после записи в регистр RSR			
1	В принятом символе не обнаружен корректный стоповый бит (единица). При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера			
–	31-4	Зарезервировано		

Примечание – Все флаги сбрасываются записью единицы.

## FR – регистр флагов

Смещение: + 18h

Сброс: 1h

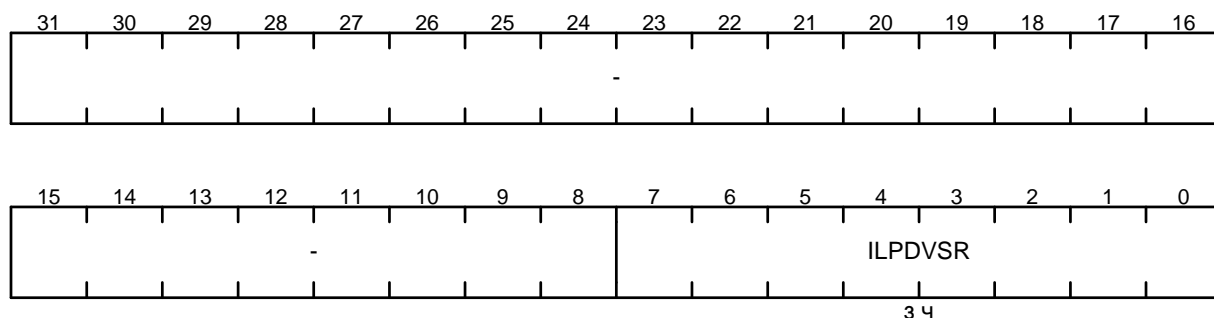


Поле	Биты	Описание
RI	8	Флаг состояния линии UART <sub>x</sub> _RI
		0   Линия неактивна
		1   Линия активна
TXFE/ RXFE	7/ 4	Флаг пустоты буфера передатчика/приемника. Установка флага зависит от состояния бита FEN регистра LCRH
		0   Буфер не пуст
		1   Буфер пуст
Примечание – Бит TXFE/RXFE не дает никакой информации о наличии данных в сдвиговом передающем регистре.		
RXFF/ TXFF	6/ 5	Флаг заполнения буфера приемника/передатчика. Установка флага зависит от состояния бита FEN регистра LCRH (т. е. включен режим FIFO или нет)
		0   Буфер не заполнен
		1   Если режим FIFO запрещен, бит устанавливается, когда буферный регистр приемника/передатчика занят. Если режим FIFO разрешен бит устанавливается, если заполнен буфер приемника/ передатчика
BUSY	3	Бит занятости блока UART
		0   Блок не занят
		1   Блок передает данные на линию. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Также бит устанавливается при наличии данных в буфере передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
DCD	2	Флаг состояния линии UART <sub>x</sub> _DCD
		0   Линия неактивна
		1   Линия активна
DSR	1	Флаг состояния линии UART <sub>x</sub> _DSR
		0   Линия неактивна
		1   Линия активна
CTS	0	Флаг состояния линии UART <sub>x</sub> _CTS
		0   Линия неактивна
		1   Линия активна
–	31-9	Зарезервировано

## ILPR – регистр делителя частоты для пониженного энергопотребления IrDA

Смещение: + 20h

Сброс: 0h

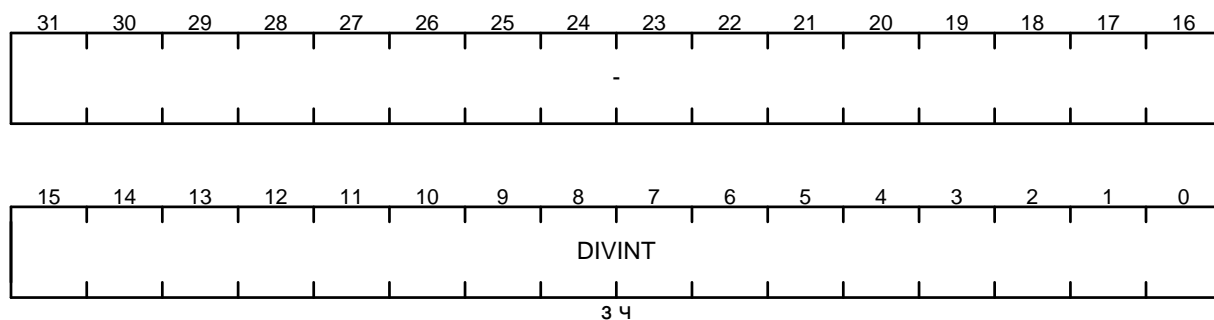


Поле	Биты	Описание
ILPDVSR	7-0	Значение делителя частоты для формирования тактового сигнала IrDA при работе в low-power режиме. 0h – недопустимое значение.  Примечание – При записи значения 0h тактовый сигнал не будет генерироваться.
–	31-8	Зарезервировано

## IBRD – регистр целой части делителя скорости обмена данными

Смещение: + 24h

Сброс: 0h

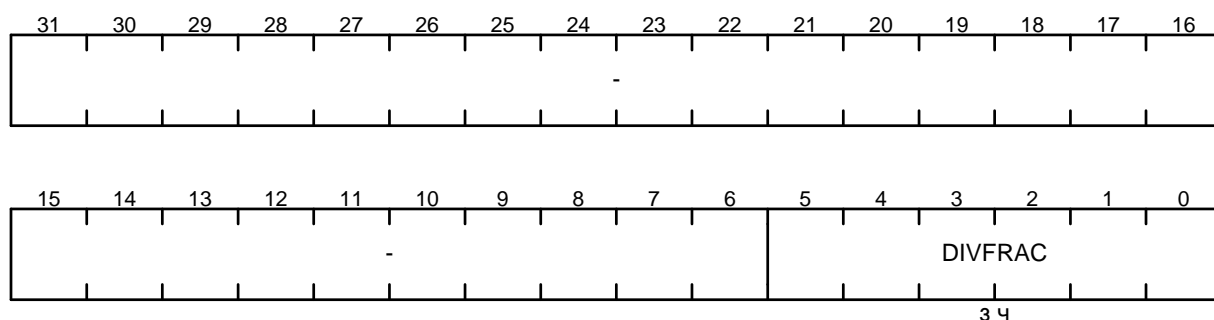


Поле	Биты	Описание
DIVINT	15-0	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. Минимальное значение 0001h
–	31-16	Зарезервировано

### FBRD – регистр целой дробной части делителя скорости обмена данными

Смещение: + 28h

Сброс: 0h

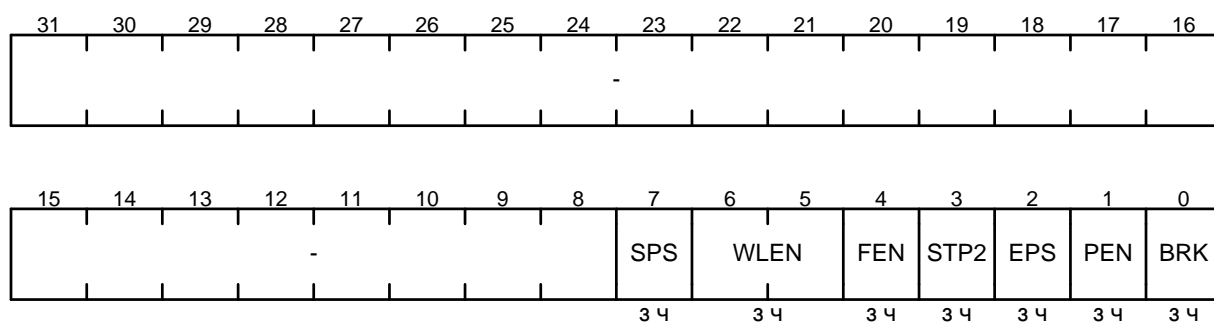


Поле	Биты	Описание
DIVFRAC	5-0	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. При DIVINT = FFFFh, значение DIVFRAC может быть только 00h.  Примечание – Невыполнение этого условия приведет к прерыванию приема/передачи
–	31-6	Зарезервировано

### LCRH – регистр управления линией

Смещение: + 2Ch

Сброс: 0h



Поле	Биты	Описание	
SPS	7	Бит разрешения передачи бита четности с фиксированным значением	
		0	Запрещено
		1	На месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. При EPS = 0 на месте бита четности передается единица. При EPS = 1 на месте бита четности передается ноль
WLEN	6-5	Поле количества передаваемых/принимаемых информационных бит	
		00	5 бит
		01	6 бит
		10	7 бит
	11	8 бит	

Поле	Биты	Описание	
FEN	4	Бит включения режима FIFO буфера приемника и передатчика	
		0	Выключен
		1	Включен
STP2	3	Бит выбора режима передачи стопового бита	
		0	Один стоповый бит
		1	Два стоповых бита (следует помнить, что приемник не проверяет наличие дополнительного стопового бита в кадре)
EPS	2	Бит паритета. Определяет четность числа, до которого дополняется количество единиц в информационной части кадра	
		0	Нечетное число
		1	Четное число
PEN	1	Бит включения проверки четности	
		0	Выключена. Кадр не содержит бита четности
		1	Включена. Бит четности передается в кадре и проверяется
BRK	0	Флаг разрыва линии	
		0	Нормальная работа
		1	По завершении передачи текущего символа на выходе передатчика 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
–	31-8	Зарезервировано	

Примечание – Дополнительная информация о бите паритета кадра в таблице А.21.1.

Таблица А.21.1 – Зависимость бита паритета в кадре от состояния битов регистра LCRH

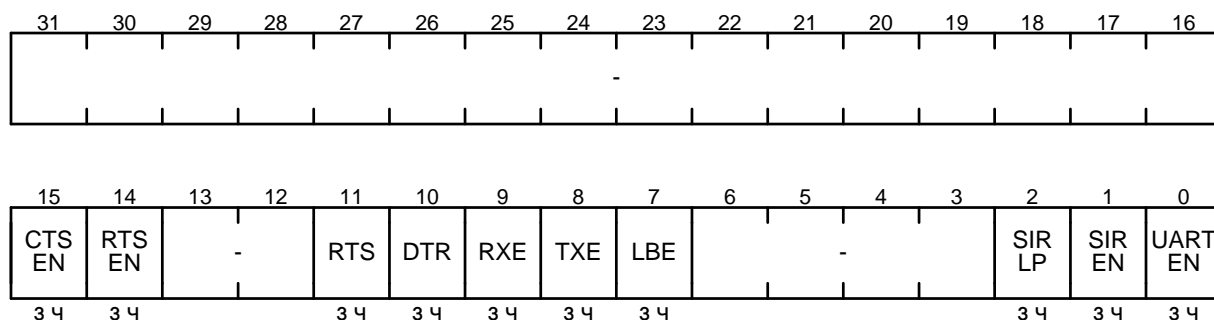
Биты регистра LCRH			Наличие и состояние бита паритета
SPS	EPS	PEN	
Не важно	Не важно	0	Не передается, не проверяется
0	0	1	Проверка нечетности слова данных
0	1	1	Проверка четности слова данных
1	0	1	Бит четности постоянно равен единице
1	1	1	Бит четности постоянно равен нулю



## CR – регистр управления

Смещение: + 30h

Сброс: 300h

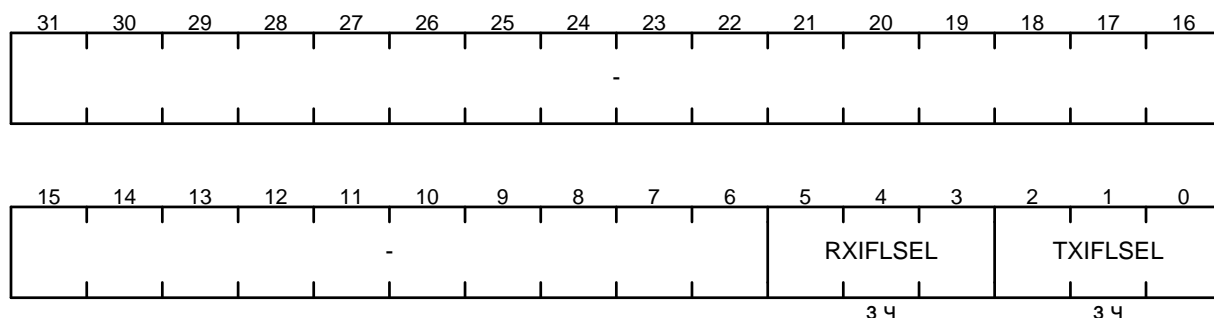


Поле	Биты	Описание	
CTSEN	15	Бит разрешения аппаратного управления потоком данных по линии CTS	
		0	Запрещено
		1	Разрешено. Данные передаются в линию только при активном значении сигнала UARTx_CTS
RTSEN	14	Бит разрешения аппаратного управления потоком данных по линии UARTx_RTS	
		0	Запрещено
		1	Разрешено. Запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере приемника
RTS	11	Бит программного управления состоянием линии модема UARTx_RTS	
		0	Высокий уровень
		1	Низкий уровень
DTR	10	Бит программного управления состоянием линии модема UARTx_DTR	
		0	Высокий уровень
		1	Низкий уровень
RXE	9	Бит разрешения приема	
		0	Запрещено
		1	Разрешено
TXE	8	Бит разрешения передачи	
		0	Запрещено
		1	Разрешено
SIRLP	2	Бит включения низкопотребляющего режима ИК передатчика	
		0	Запрещено
		1	Разрешено
SIREN	1	Бит разрешения работы ИК передатчика	
		0	Запрещено
		1	Разрешено
UARTE N	0	Бит разрешения работы приемопередатчика	
		0	Запрещено
		1	Разрешено
–	31-16, 13-12, 7-3	Зарезервировано	

## IFLS – регистр порога прерывания по заполнению буфера в режиме FIFO

Смещение: + 34h

Сброс: 12h

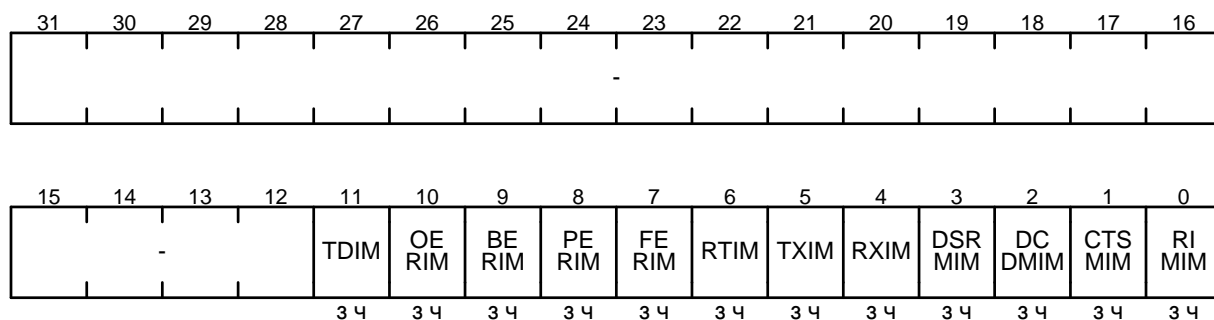


Поле	Биты	Описание	
RXIFLSEL/ TXIFLSEL	5-3/ 2-0	Порог заполнения/опустошения буфера приемника/передатчика, по достижении которого будет генерироваться прерывание	
		000	Заполнение/опустошение на 1/8
		001	Заполнение/опустошение на 1/4
		010	Заполнение/опустошение на 1/2 (по умолчанию)
		011	Заполнение/опустошение на 3/4
		100	Заполнение/опустошение на 7/8
		Комбинации 101, 110 и 111 зарезервированы	
–	31-6	Зарезервировано	

## IMSC – регистр маски прерываний

Смещение: + 38h

Сброс: 0h



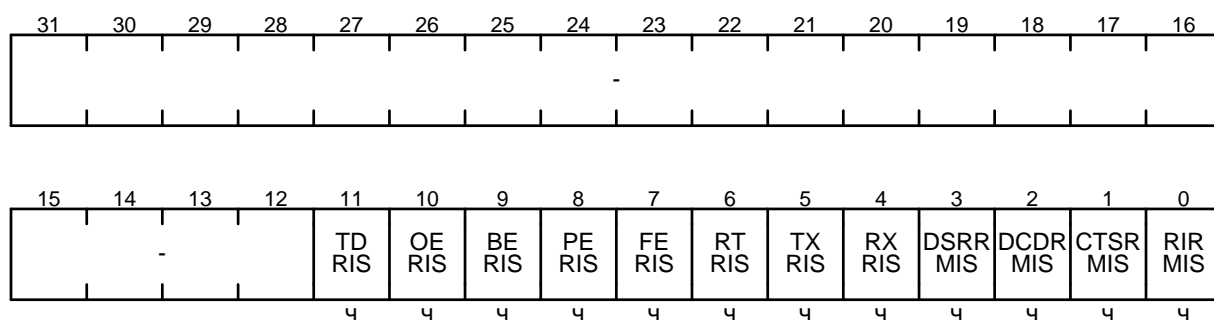
Поле	Биты	Описание
TDIM	11	Окончание передачи в линии
OEIM	10	Переполнение буфера приемника
BEIM	9	Разрыв линии
PEIM	8	Ошибка контроля четности
FEIM	7	Ошибка в структуре кадра
RTIM	6	Таймаут приема данных
TXIM	5	Порог опустошения буфер передатчика
RXIM	4	Порог переполнения буфера приемника
DSRMIM	3	Флаг изменение состояния линии UART <sub>x</sub> _DSR
DCDMIM	2	Флаг изменение состояния линии UART <sub>x</sub> _DCD
CTSMIM	1	Флаг изменение состояния линии UART <sub>x</sub> _CTS
RIMIM	0	Флаг изменение состояния линии UART <sub>x</sub> _RI
–	31-12	Зарезервировано

Примечание – Маска прерываний формируется установкой бит

## RIS – регистр состояния прерываний

Смещение: + 3Ch

Сброс: 0h

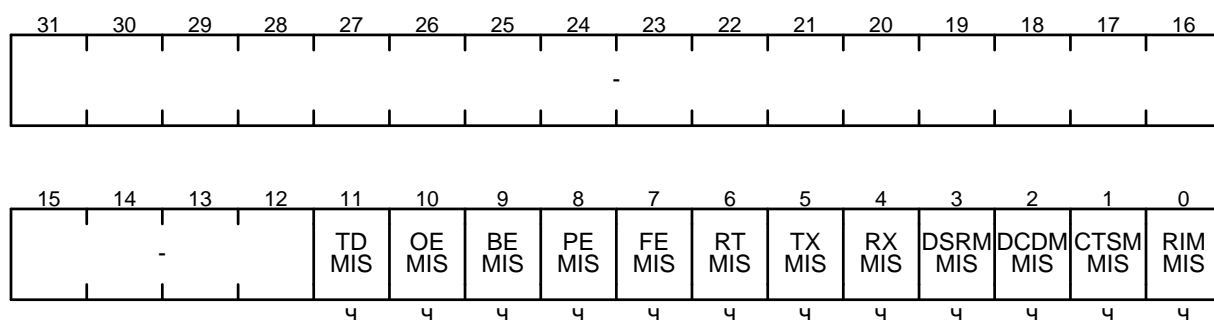


Поле	Биты	Описание
TDRIS	11	Флаг окончания передачи в линии
OERIS	10	Флаг переполнения буфера приемника
BERIS	9	Флаг разрыва линии
PERIS	8	Флаг ошибки контроля четности
FERIS	7	Флаг ошибки в структуре кадра
RTRIS	6	Флаг таймаута приема данных
TXRIS	5	Флаг порога опустошения буфер передатчика
RXRIS	4	Флаг порога переполнения буфера приемника
DSRRMIS	3	Флаг изменение состояния линии UARTx_DSR
DCDRMIS	2	Флаг изменение состояния линии UARTx_DCD
CTSRMIS	1	Флаг изменение состояния линии UARTx_CTS
RIRMIS	0	Флаг изменение состояния линии UARTx_RI
–	31-12	Зарезервировано

## MIS – регистр состояния прерываний с маскированием

Смещение: + 40h

Сброс: 0h



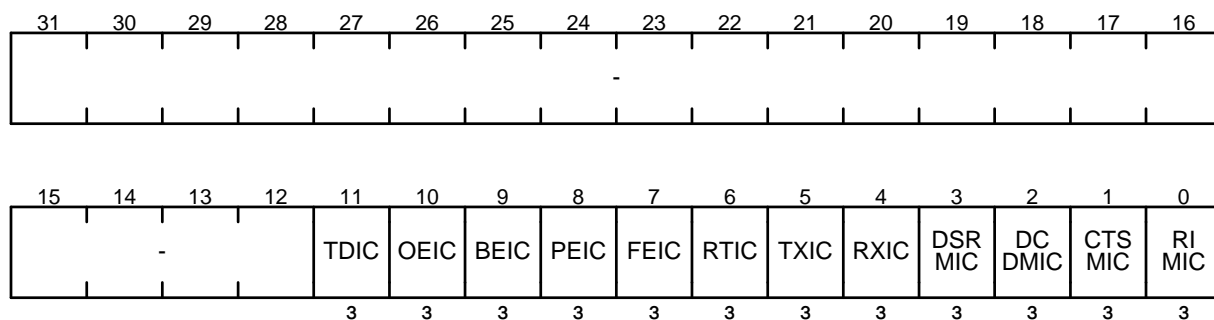
Поле	Биты	Описание
TDMIS	11	Флаг окончания передачи в линии
OEMIS	10	Флаг переполнения буфера приемника
BEMIS	9	Флаг разрыва линии
PEMIS	8	Флаг ошибки контроля четности
FEMIS	7	Флаг ошибки в структуре кадра
RTMIS	6	Флаг таймаута приема данных
TXMIS	5	Флаг порога опустошения буфер передатчика
RXMIS	4	Флаг порога переполнения буфера приемника
DSRMMIS	3	Флаг изменение состояния линии UARTx_DSR
DCDMMIS	2	Флаг изменение состояния линии UARTx_DCD
CTSMIS	1	Флаг изменение состояния линии UARTx_CTS
RIMMIS	0	Флаг изменение состояния линии UARTx_RI
–	31-12	Зарезервировано

Примечание – Устанавливаются флаги, которые закрыты маской регистра IMSC.

## ICR – регистр сброса прерываний

Смещение: + 44h

Сброс: 0h



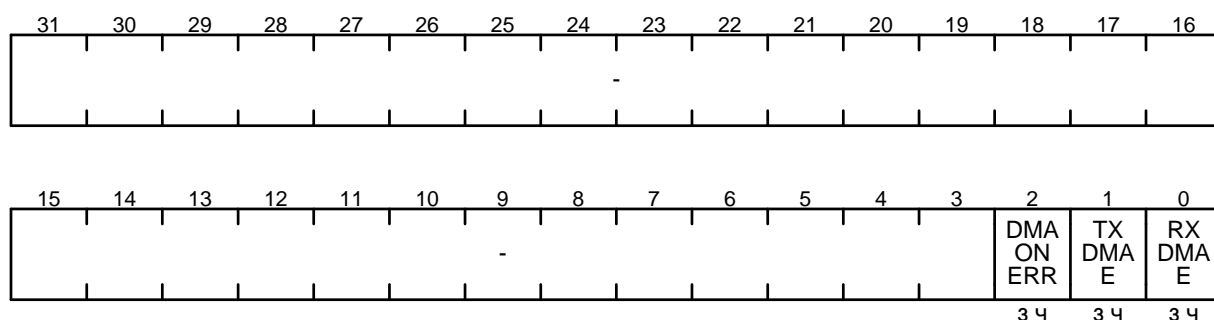
Поле	.ы	Описание
TDIC	11	Бит сброса флага окончания передачи в линии
OEIC	10	Бит сброса флага переполнения буфера приемника
BEIC	9	Бит сброса флага разрыва линии
PEIC	8	Бит сброса флага ошибки контроля четности
FEIC	7	Бит сброса флага ошибки в структуре кадра
RTIC	6	Бит сброса флага таймаута приема данных
TXIC	5	Бит сброса флага порога опустошения буфер передатчика
RXIC	4	Бит сброса флага порога переполнения буфера приемника
DSRMIC	3	Бит сброса флага изменения состояния линии UARTx_DSR
DCDMIC	2	Бит сброса флага изменения состояния линии UARTx_DCD
CTSMIC	1	Бит сброса флага изменения состояния линии UARTx_CTS
RIMIC	0	Бит сброса флага изменения состояния линии UARTx_RI
–	31-12	Зарезервировано

Примечание – Запись единиц в биты регистра сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов

## DMACR – регистр управления прямым доступом к памяти

Смещение: + 48h

Сброс: 0h



Поле	Биты	Описание
DMAONERR	2	Блокирование запросов UARTRXDMASREQ и UARTRXDMAABREQ от приемника в случае прерывания по ошибке
		0   Выключено
		1   Включено
TXDMAE/ RXDMAE	1/ 0	Бит разрешения формирования запросов блока DMA для обслуживания буфера передатчика/приемника
		0   Запрещено
		1   Разрешено
–	31-3	Зарезервировано

## A.22 Регистры контроллера интерфейса CAN 2.0b

**Базовый адрес:** 4002\_0000h

**Смещение:**

+ 100h (LIST)	Регистры свободного списка
+ 140h (MSPND)	Регистры ждущих прерываний
+ 180h (MSID)	Регистры индексов сообщений
+ 200h (Node_0)	Регистры узла 0
+ 300h (Node_1)	Регистры узла 1
+ 1000h + 20h*m (Msg_m)	Регистры объекта сообщений m

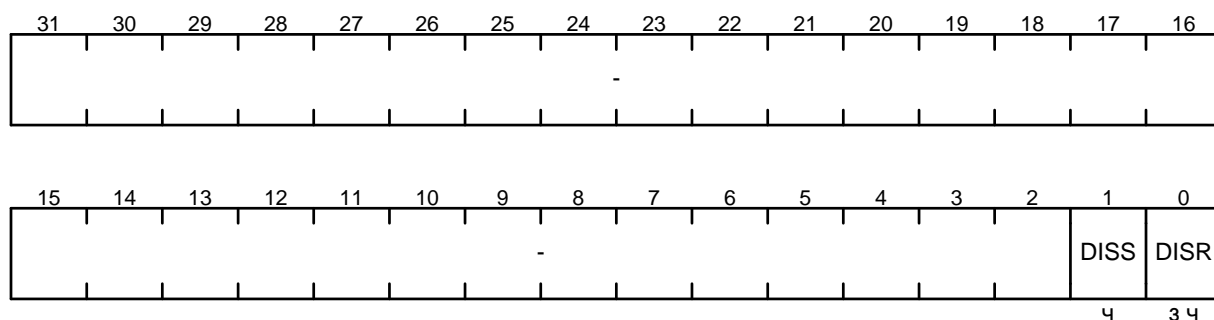
**Мнемоника:** Node\_n;  
Msg\_m

**Примечание:** In – номер списка от 0 до 7;  
n – номер узла 0 или 1;  
m – номер объекта сообщения от 0 до 255;  
mo – номер объекта сообщения в шестнадцатеричном формате от 0h до FFh (что соответствует диапазону от 0 до 255)

### CLC – регистр управления частотой

Смещение: + 00h

Сброс: 3h



Поле	Биты	Описание
DISS	1	Бит состояния контроллера CAN
		0   Включен
		1   Выключен
DISR	0	Бит выключения контроллера CAN
		0   Нет действий
		1   Запись единицы запускает механизм выключения
–	31-2	Зарезервировано

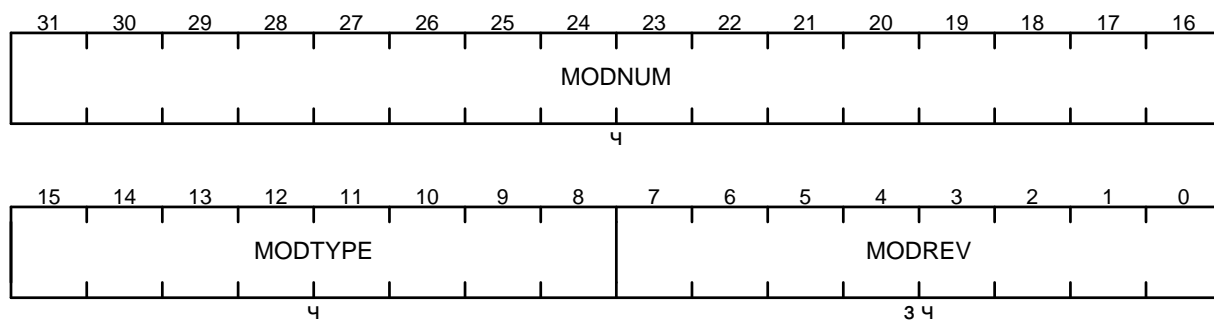
**Примечание** – Когда контроллер CAN находится в выключенном состоянии, только регистр CLC доступен для записи и чтения, доступ к остальным регистрам не возможен.



## ID – регистр идентификации

Смещение: + 08h

Сброс: 2B\_C051h

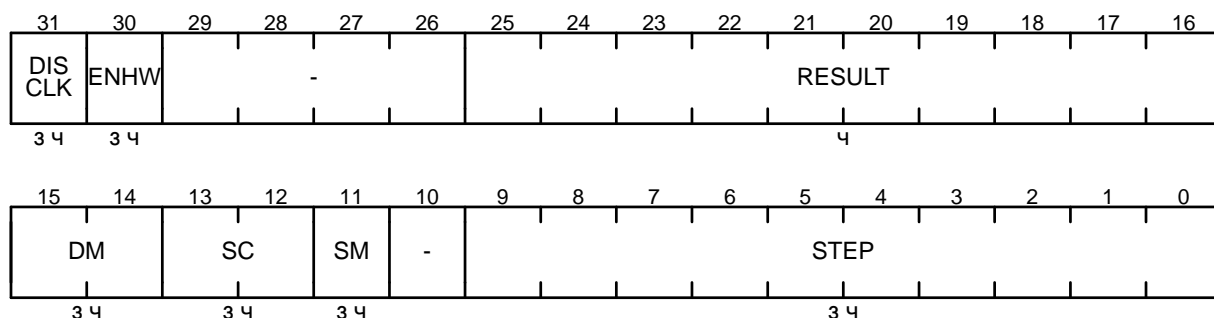


Поле	Биты	Описание
MODNUM	31-16	Идентификационный номер контроллера CAN
MODTYPE	15-8	Разрядность контроллера CAN
MODREV	7-0	Число модификаций контроллера CAN

## FDR – регистр делителя

Смещение: + 0Ch

Сброс: 0h

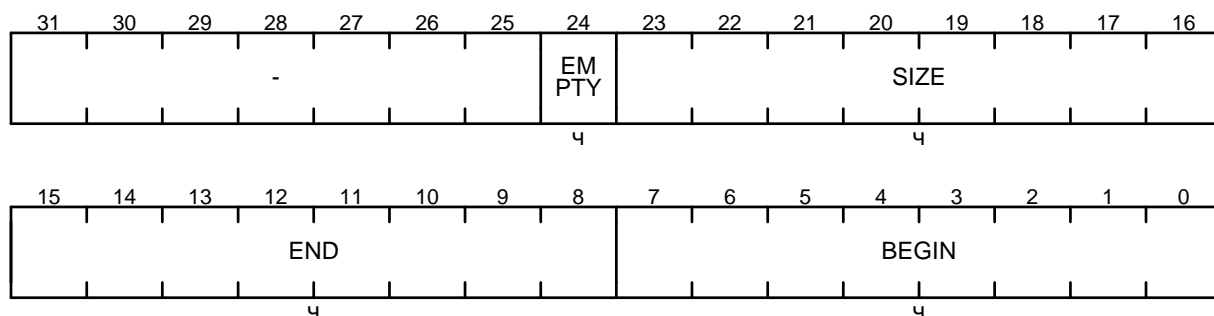


Поле	Биты	Описание	
DISCLK	31	Бит запрета внутреннего тактирования	
		0	Генерирование сигнала FCAN разрешено
		1	Генерирование сигнала FCAN запрещено
ENHW	30	Бит контроля синхронизации. Это бит аппаратно удерживается в сброшенном состоянии и не может быть установлен	
RESULT	25-16	Счетчик делителя частоты	
DM	15-14	Поле задания режима делителя частоты	
		00b	Счетчик выключен
		01b	Нормальный режим работы. Синхросигнал FOUT формируется. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP.
		10b	Режим дробного деления. Синхросигнал FOUT формируется. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP.
		11b	Синхросигнал FOUT не генерируется. Сигнал сброса внешнего делителя в состоянии логической единицы. Поле RESULT не меняется
STEP	9-0	Шаг делителя. Поле хранит значение, которое загружается в RESULT при переполнении счетчика делителя	
–	29-26, 13-10	Зарезервировано	

## LIST – массив регистров свободного списка

Смещение: LIST + 4\*ln

Сброс: 7F\_7F00h (для списка 0), 100\_0000h (для списков 1 – 7)

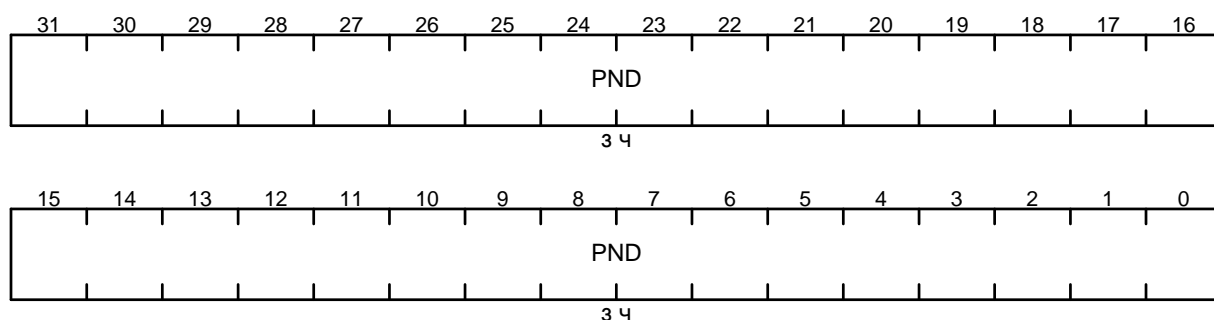


Поле	Биты	Описание
EMPTU	24	Индикатор пустого списка
		0   В списке есть как минимум один элемент
		1   Список пуст
SIZE	23-16	Размер списка. Количество элементов (объектов сообщений) в списке. Значение поля SIZE всегда на единицу меньше числа элементов. Если список пуст, SIZE = 00h
END	15-8	Номер объекта сообщения, находящегося последним в списке. Поле может принимать значения от 00h до Fh, согласно количеству объектов сообщений (256)
BEGIN	7-0	Номер объекта сообщения, находящегося первым в списке. Поле может принимать значения от 00h до Fh, согласно количеству объектов сообщений
–	31-25	Зарезервировано

## MSPND – массив регистров ждущих прерываний

Смещение: MSPND + 4\*In

Сброс: 0h

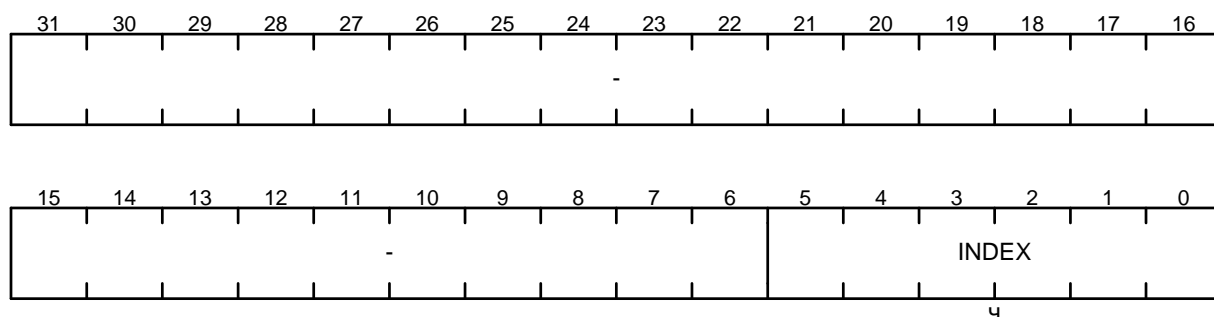


Поле	Биты	Описание
PND	31-0	Поле ждущих битов сообщений. Каждому объекту сообщения выделяется один бит. Биты устанавливаются только аппаратно. Установленные биты сбрасываются аппаратно по окончании обслуживания запроса прерывания или могут быть сброшены в любой момент программно

## MSID – массив регистров индекса сообщения

Смещение: MSID + 4\*In

Сброс: 20h

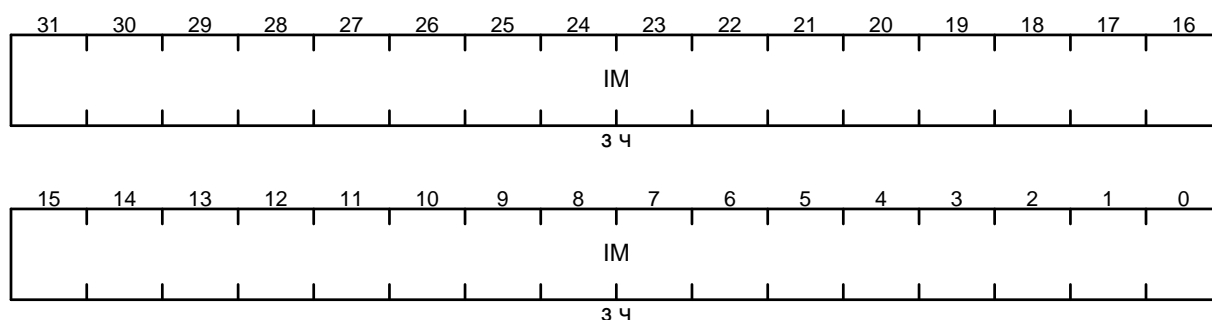


Поле	Биты	Описание
INDEX	5-0	Поле номера ждущего бита. Если в регистре MSPND есть установленные биты, которые не маскируются соответствующими битами регистра MSIMASK, то поле INDEX будет указывать на самый младший из них. Если в регистре MSPND нет установленных битов или они замаскированы, то в поле INDEX будет находиться значение 20h, указывающее на бит 31 регистра MSPND
–	31-6	Зарезервировано

## MSIMASK – регистр маски индекса сообщения

Смещение: + 1C0h

Сброс: 0h

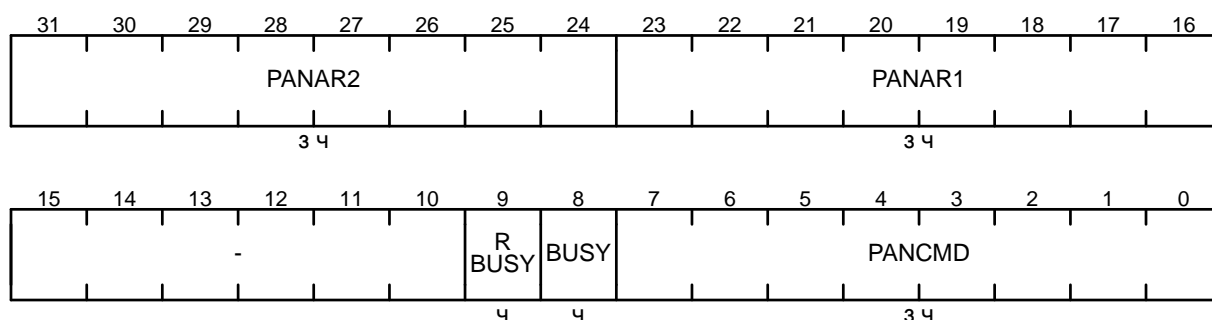


Поле	Биты	Описание
IM	31-0	Маска для ждущих битов сообщений. Учитывается состояние только тех бит регистра MSPND, для которых в поле IM установлены соответствующие биты

## PANCTR – регистр панели команд

Смещение: + 1C4h

Сброс: 301h



Поле	Биты	Описание
PANAR2	31-24	Панель аргумента 2 (см. таблицу А.22.1)
PANAR1	23-16	Панель аргумента 1 (см. таблицу А.22.1)
R BUSY	9	Флаг занятости панелей аргументов
		0   Нет действий 1   Выполняется команда списка, результат выполнения которой будет записан в поле PANAR1 и поле PANAR2
BUSY	8	Флаг занятости панелей аргументов
		0   Панели готовы для записи 1   Панели заняты – ожидают записи по окончании выполнения команды
PANCMD	7-0	Поле команды (см. таблицу А.22.1). После выполнения команды в это поле записывается 00h
–	15-10	Зарезервировано

Таблица А.22.1 – Коды команд работы со списками

Поле PANCMD	Поле PANAR2	Поле PANAR1	Описание команды
00h	–	–	Нет операции. Никаких действий не выполняется
01h	Результат: бит 7 – ошибка, бит 6 – не определен	–	Инициализация списков. Запуск инициализации для очистки битовых полей CTRL и LIST всех объектов сообщений. Регистры LIST0 – List8 устанавливаются в свои значения после сброса. Это приводит к переносу всех объектов сообщений в список №0 (список нераспределенных объектов сообщений). Инициализация списков требует, чтобы биты INIT и CCE регистра NCR были установлены для обоих узлов. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Инициализация завершена успешно; - 1 – Инициализация не завершена, поскольку не все биты INIT и CCE были установлены. Команда инициализации списков автоматически запускается при каждом сбросе контроллера CAN, за исключением случая, когда все регистры объектов сообщений уже сброшены
02h	Аргумент: номер списка	Аргумент: номер объекта сообщения	Статическое занесение объекта сообщения в список. Объект сообщения переносится из текущего списка в список, указанный полем PANAR2 и добавляется в его конец. Эта команда также используется для дераспределения объекта сообщения, т. е. переноса его в список № 0 (если поле PANAR2 равно 00h)
03h	Аргумент: номер списка Результат: бит 7 – ошибка, бит 6 – не определен	Результат: номер объекта сообщения	Динамическое занесение объекта сообщения в список. Первый объект сообщения списка №0 переносится в список, указанный полем PANAR2, и добавляется в его конец. Номер объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст
04h	Аргумент: номер объекта сообщения	Аргумент: текущий номер объекта сообщения	Перемещение по списку вверх. Перенос объекта сообщения с номером PANAR1 на одну позицию выше, чем расположен объект сообщения с номером PANAR2

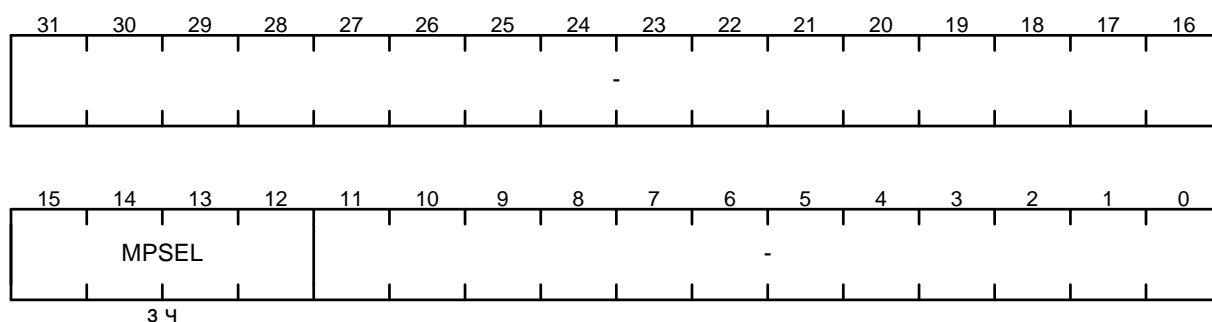
Окончание таблицы А.22.1

Поле PANCMD	Поле PANAR2	Поле PANAR1	Описание команды
05h	Аргумент: номер объекта сообщения  Результат: бит 7 – ошибка, бит 6 – не определен	Результат: номер добавленного объекта сообщения	Динамическая вставка в список. Первый объект сообщений списка №0 вставляется на одну позицию выше, чем расположен объект сообщения с номером PANAR2. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст
06h	Аргумент: номер объекта сообщения	Аргумент: текущий номер объекта сообщения	Перемещение по списку вниз. Перенос объекта сообщения с номером PANAR1 на одну позицию ниже, чем расположен объект сообщения с номером PANAR2
07h	Аргумент: номер объекта сообщения  Результат: бит 7– ошибка, бит 6 – не определен	Результат: номер добавленного объекта сообщения	Динамическая вставка в список. Первый объект сообщения списка №0 вставляется на одну позицию ниже, чем расположен объект сообщения с номером PANAR2. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст
08h – FFh	–	–	Зарезервировано

**MCR – регистр управления**

Смещение: + 1C8h

Сброс: 0h

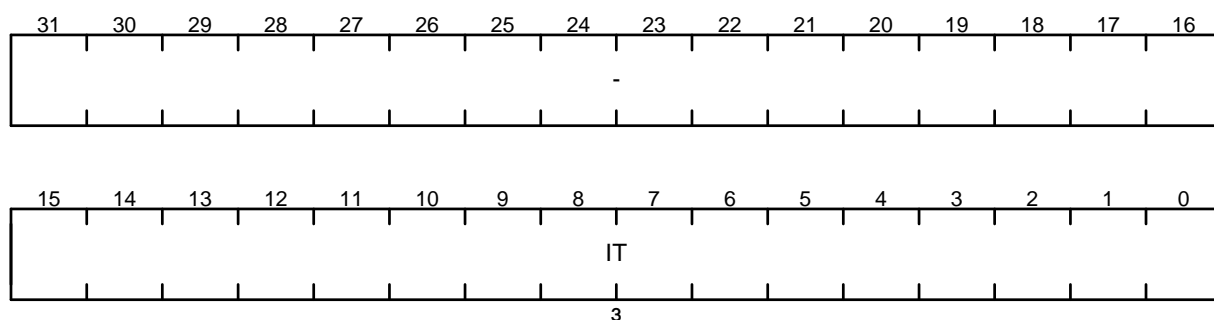


Поле	Биты	Описание
MPSEL	15-12	Поле задания позиции ждущего бита сообщения после приема/передачи сообщения
–	31-16, 11-0	Зарезервировано

## MITR – регистр прерываний

Смещение: + 1CCh

Сброс: 0h

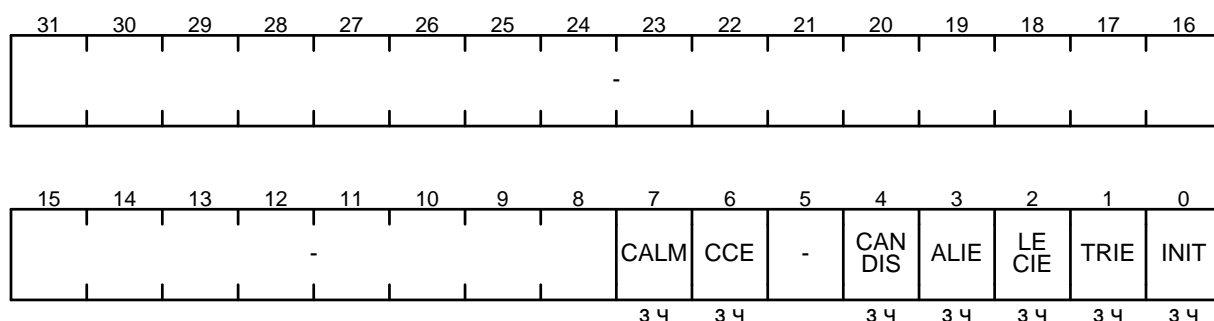


Поле	Биты	Описание
IT	15-0	Поле генератора прерываний. Каждый бит поля связан с одной из линий прерываний. Номера битов от 0 до 15 соответствуют номерам линий прерываний. Для того, чтобы сгенерировать одно или несколько прерываний, следует установить соответствующие биты. Установленные биты сбрасываются аппаратно
–	31-16	Зарезервировано

## NCR – регистр управления узла

Смещение: Node\_n + 00h

Сброс: 1h



Поле	Биты	Описание	
CALM	7	Бит включения режима анализа узла	
		0	Режим выключен
		1	Установка бита включает режим анализа узла. В этом режиме сообщения могут только приниматься, бит подтверждения не посылается после успешного приема сообщения, флаг активной ошибки посылается рецессивным вместо доминантного. На линии отправки сообщений поддерживается высокий уровень сигнала
		Бит может быть установлен только, если установлен бит INIT	
CCE	6	Бит разрешения изменения конфигурации узла. Управляет доступом к регистрам NBTR, NPCR и NECNT	
		0	Только чтение
		1	Полный доступ

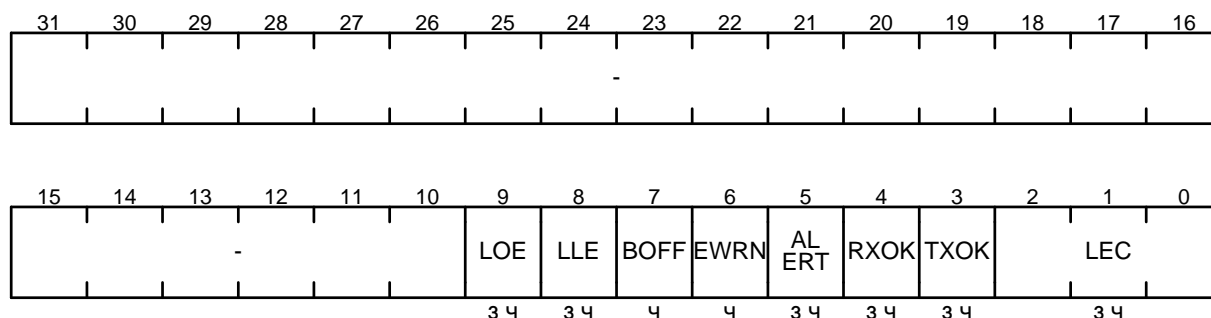


Поле	Биты	Описание	
CANDIS	4	Бит выключения узла	
		0	Сброс бита включает узел
		1	Установка бита выключает узел. Сначала узел переходит в состояние «простоя» или «отключен от шины», далее аппаратно устанавливается бит INIT, и, если разрешено, генерируется прерывание ALERT
ALIE	3	Бит разрешения прерывания ALERT от узла	
		0	Запрещено
		1	Разрешено
LEIE	2	Бит разрешения прерывания от узла при обнаружении кода последней ошибки	
		0	Запрещено
		1	Разрешено
TRIE	1	Бит разрешения прерывания от узла по окончании передачи/приема	
		0	Запрещено
		1	Разрешено
INIT	0	Инициализация узла	
		0	Сброс бита разрешает участие узла в работе CAN шины. Узел ожидает последовательность из 11 рецессивных бит на шине и включается в трафик. Если на момент сброса бита INIT узел находился в состоянии «отключен от шины», начинается процесс выхода из этого состояния в следующем порядке – получение 128 последовательностей бит (каждая из 11 рецессивных бит), выход из состояния «отключен от шины», включение в трафик
		1	Установка бита INIT прекращает участие узла в трафике. Все текущие передачи останавливаются, линии передач переходят в рецессивное состояние. Если на момент установки бита INIT узел находился в состоянии «отключен от шины», процесс выхода из этого состояния продолжается до его завершения. Далее узел остается неактивным до тех пор, пока установлен бит INIT
–	31-8, 5	Зарезервировано	

## NSR – регистр состояния узла

Смещение: Node\_n + 04h

Сброс: 0h



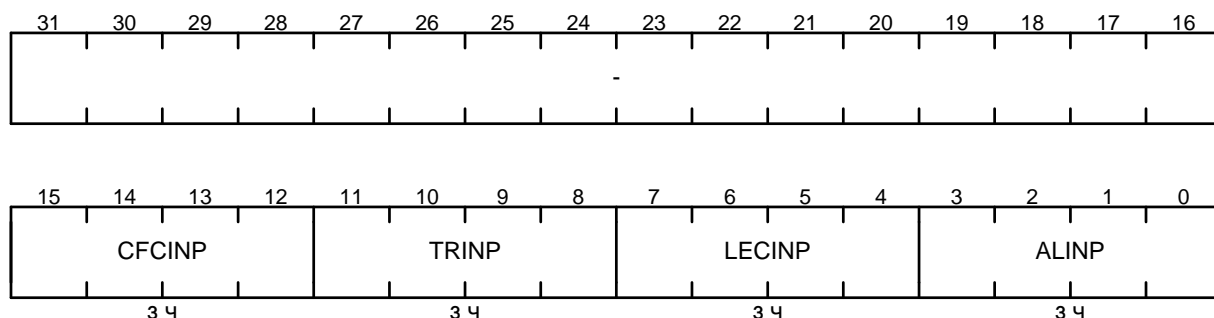
Поле	Биты	Описание
LOE	9	Флаг ошибки номера списка
		0   Ошибок не обнаружено
		1   Обнаружена ошибка при фильтрации принимаемого сообщения. В регистре MOSTAT объекта сообщения обнаружен неверный номер списка
		Бит должен сбрасываться программно записью нуля
LLE	8	Флаг ошибки списка
		0   Ошибок не обнаружено
		1   Обнаружена ошибка при фильтрации принимаемого сообщения. Количество элементов списка, принадлежащего узлу, отличается от указанного в поле SIZE соответствующего регистра списка
		Бит должен сбрасываться программно записью нуля
BOFF	7	Флаг состояния «отключен от шины»
		0   Узел не находится в состоянии «отключен от шины»
		1   Узел находится в состоянии «отключен от шины»
EWRN	6	Флаг критического количества ошибок
		0   Лимит ошибок еще не достигнут
		1   По крайней мере, один из счетчиков ошибок (REC, TEC) достиг лимита ошибок, заданного полем EWRNLVL регистра NECNT узла
ALERT	5	Флаг предупреждения ALERT
		0   Нет событий
		1   Произошло одно или несколько не взаимоисключающих событий: - модификация бита BOFF; - модификация/установка бита LOE; - установка бита LLE; - аппаратная установка бита INIT
		Бит должен сбрасываться программно записью нуля
RXOK	4	Флаг успешного приема сообщения
		0   Полученных сообщений нет
		1   Сообщение получено
		Бит должен сбрасываться программно записью нуля
TXOK	3	Флаг успешной передачи сообщения
		0   Переданных сообщений нет
		1   Сообщение передано без ошибок с получением подтверждения
		Бит должен сбрасываться программно записью нуля

Поле	Биты	Описание	
LEC	2-0	Поле кода последней из обнаруженных ошибок работы узла	
		000	Ошибок нет
		001	Ошибка стаффинга (заполнения, STUFF ERROR). Может быть обнаружена во время передачи шестого бита из последовательности шести одинаковых бит в поле сообщения, которое должно быть кодировано методом разрядного заполнения (заключается в том, что после передачи пяти битов одинаковой полярности, шестой бит должен иметь противоположную полярность и вставляться передатчиком в поток данных автоматически, приемник пропускает этот бит)
		010	Ошибка формы (FORM ERROR). Обнаруживается, если: - в битовом поле фиксированного формата содержится количество битов, отличающееся от установленного; - на месте рецессивного бита находятся доминантный или наоборот. Исключение – для приемника доминантный бит в течение последнего бита поля «конец кадра» не интерпретируется как ошибка формы
		011	Ошибка подтверждения (ACKNOWLEDGMENT ERROR). Обнаруживается передатчиком всякий раз, когда он не обнаруживает доминантный бит ACK в «области подтверждения»
		100	Разрядная ошибка или ошибка бита 1 (BIT 1 ERROR). Узел, который передает данные на шину, осуществляет мониторинг шины. Ошибка бита 1 имеет место, если при передаче рецессивного «1» бита (за исключением битов полей арбитража и подтверждения) на шине обнаруживается доминантный «0» бит
		101	Разрядная ошибка или ошибка бита 0 (BIT 0 ERROR). Ошибка возникает в случаях: - во время передачи сообщения (или бита подтверждения, флага активной ошибки, флага перезагрузки), узел передает доминантный бит «0», но на шине обнаруживается рецессивный «1»; - во время выхода из состояния «отключен от шины» при каждом обнаружении последовательности из 11 рецессивных битов. В этом случае, ЦП может использовать код 101 для отслеживания длительного простоя шины
		110	Ошибка циклического избыточного кода (CRC ERROR). Передатчик по установленному алгоритму вычисляет значение контрольной суммы (CRC) для передаваемых данных и вставляет ее в сообщение. Приемник, после получения данных, вычисляет CRC по тому же алгоритму, что и передатчик, и сравнивает вычисленное значение с принятым значением. В случае несовпадения фиксируется ошибка
		111	Код разрешения аппаратной записи в поле LEC
–	31-10	Зарезервировано	

## NIPR – регистр указателя прерываний узла

Смещение: Node\_n + 08h

Сброс: 0h



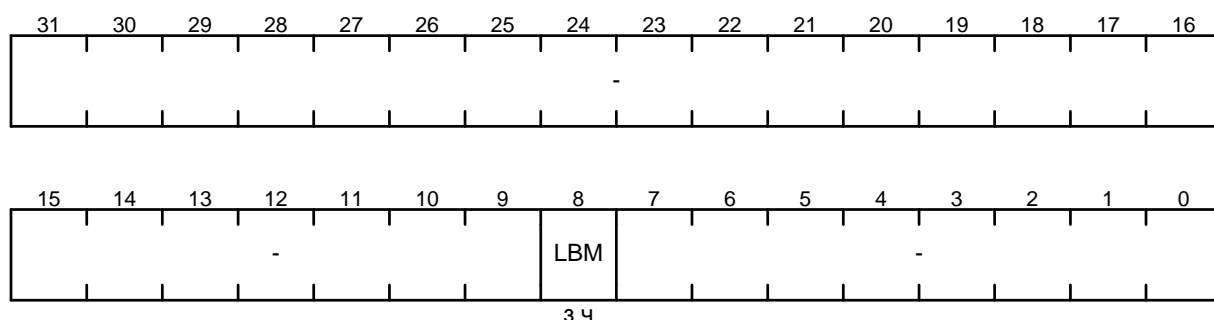
Поле	Биты	Описание
CFCINP	15-12	Указатель для прерывания при переполнении счетчика фреймов узла
TRINP	11-8	Указатель для прерывания по окончании передачи/приема сообщения
LECINP	7-4	Указатель для прерывания при записи кода последней ошибки
ALINP	3-0	Указатель для прерывания ALERT
–	31-16	Зарезервировано

Примечание – Каждый из указателей задает номер одной из 16 линий прерываний. Значение 0h соответствует нулевой линии, значение Fh – линии 15.

## NPCR – регистр управления портом узла

Смещение: Node\_n + 0Ch

Сброс: 0h

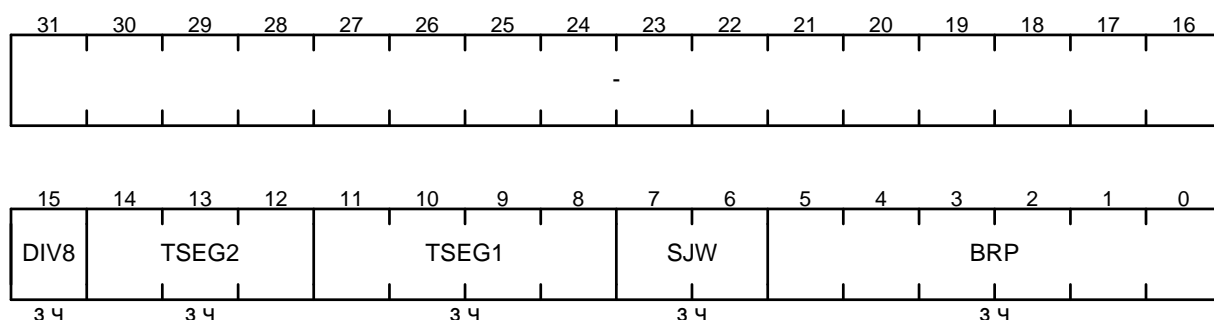


Поле	Биты	Описание	
LBM	8	Бит включения режима обратной петли (Loop-Back)	
		0	Выключен
		1	Включен. В этом режиме узел подсоединяется к внутренней виртуальной CAN шине. Если для обоих узлов включен режим обратной петли, то они объединяются виртуальной CAN шиной и могут взаимодействовать друг с другом. При этом на внешних выводах узлов, соединенных с внешней физической CAN шиной, поддерживается рецессивный уровень сигнала, т. е. узлы не активны
–	31-9, 7-0	Зарезервировано	

## NBTR – регистр синхронизации битов

Смещение: Node\_n + 10h

Сброс: 0h

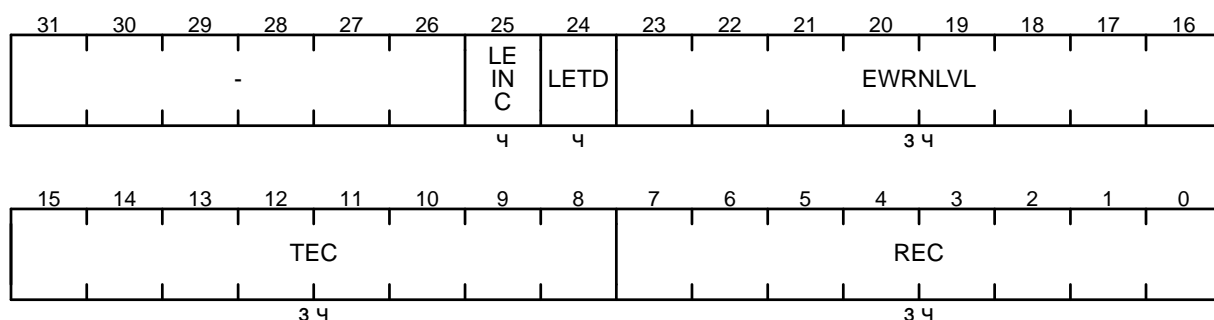


Поле	Биты	Описание	
DIV8	15	Делитель частоты на восемь	
		0	Выключен
		1	Включен
TSEG2	14-12	<p>Параметр 2.</p> <p>Временной промежуток от точки выборки до точки передачи, определяемый пользователем. Длительность сегмента равна <math>t_q \times (TSEG2 + 1)</math> и может быть уменьшена за счет ресинхронизации. Диапазон допустимых значений от 1h до 7h</p>	
TSEG1	11-8	<p>Параметр 1.</p> <p>Временной промежуток от сегмента синхронизации до точки выборки, определяемый пользователем и включающий в себя сегмент распространения. Длительность равна <math>t_q \times (TSEG1 + 1)</math> и может быть увеличена за счет ресинхронизации. Допустимые значения от 2h до 7h</p>	
SJW	7-6	<p>Ширина перехода ресинхронизации.</p> <p>Длительность равна <math>t_q \times (SJW + 1)</math></p>	
BRP	5-0	<p>Предделитель скорости передачи.</p> <p>Длительность одного кванта времени (в тактах частоты):</p> <ul style="list-style-type: none"> <li>– <math>(BRP + 1)</math>, если <math>DIV8 = 0</math>;</li> <li>– <math>8 \times (BRP + 1)</math>, если <math>DIV8 = 1</math>.</li> </ul>	
–	31-16	Зарезервировано	

## NECNT – регистр счетчика ошибок узла

Смещение: Node\_n + 14h

Сброс: 60\_0000h

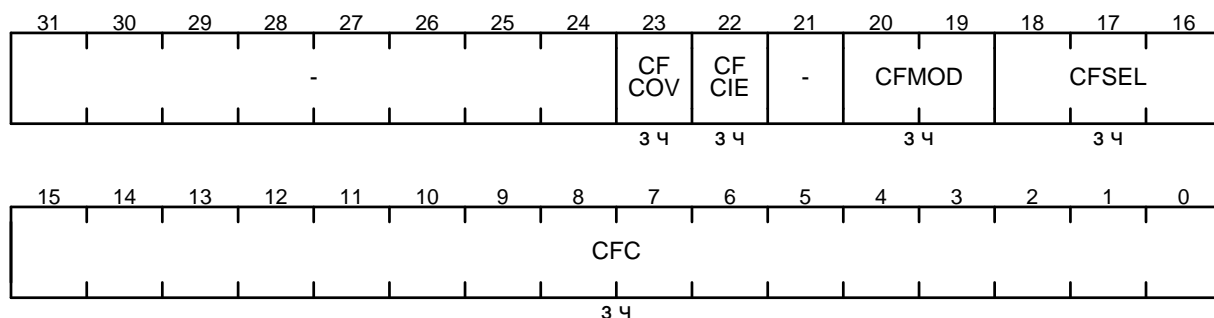


Поле	Биты	Описание	
LEINC	25	Индикатор инкрементирования при последней ошибке	
		0	Обнаруженная ошибка приводит к инкрементированию счетчика ошибок на единицу
		1	Обнаруженная ошибка приводит к инкрементированию счетчика ошибок на восемь
LETD	24	Флаг последней ошибки передачи	
		0	При приеме сообщения обнаружена ошибка, и произошло инкрементирование поля REC
		1	При передаче сообщения обнаружена ошибка, и произошло инкрементирование поля TEC
EWRNLVL	23-16	Поле задания лимита ошибок, по достижении которого выставляется флаг EWRN в регистре NSR (по умолчанию, количество ошибок – 96)	
TEC	15-8	Поле счетчика ошибок передачи сообщений	
REC	7-0	Поле счетчика ошибок приема сообщений	
–	31-26	Зарезервировано	

## NFCR – регистр счетчика сообщений узла

Смещение: Node\_n + 18h

Сброс: 0h

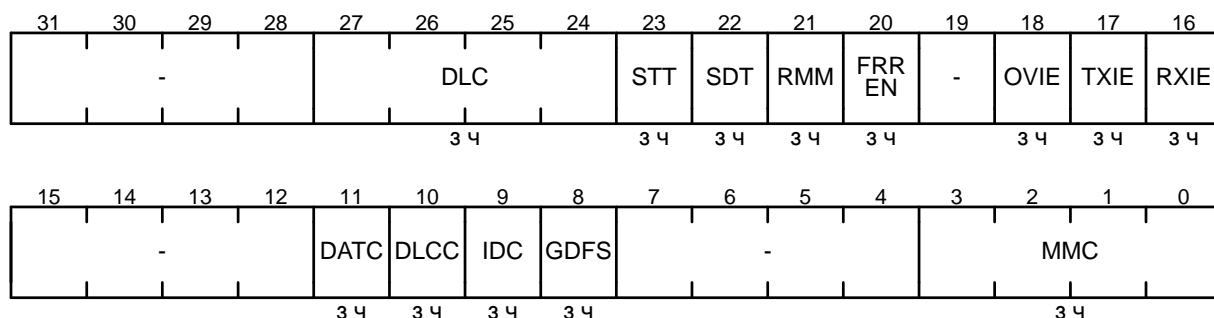


Поле	Биты	Описание
CFCOV	23	Флаг переполнения счетчика сообщений
		0   Счетчик не переполнен
		1   Счетчик переполнился. В режиме синхросчетчика этот флаг устанавливается при изменении поля CFC и, если установлен бит CFCIE, формируется прерывание
		Бит сбрасывается программно
CFCIE	22	Бит разрешения прерывания от счетчика сообщений
		0   Запрещено 1   Разрешено
CFMOD	20-19	Режим работы счетчика сообщений
		00   Счетчик сообщений инкрементируется после каждого успешного приема/передачи сообщения 01-11   Зарезервировано. Не использовать!
CFSEL	18-16	Поле задания условия инкрементирования счетчика сообщений
		**1   При получении сообщения, не имеющего объекта сообщения
		*1*   При получении сообщения, имеющего соответствующий объект сообщения
		1**   При успешной отправке сообщения
		000   Зарезервировано. Не использовать!
		Состояние бита, отмеченного символом *, неважно. Условия могут комбинироваться между собой (например, 110b или 101b).
CFC	15-0	Поле счетчика сообщений. Хранит значение счетчика сообщений при CFMOD = 00b
-	31-24, 21	Зарезервировано

## MOFCR – регистр управления функционированием объекта сообщения

Смещение: Msg\_mo + 00h

Сброс: 0h



Поле	Биты	Описание
DLC	27–24	Код длины данных. Показывает количество байт данных, находящихся в объекте сообщения. Диапазон – значение от 0 до 8. Если значение поля DLC больше 8, это автоматически указывает на 8 байт. Значение поля DLC полученного сообщения сохраняется таким, каким было получено
STT	23	Бит задания однократной пересылки данных
		0   Нет действий
SDT	22	Бит задания однократного участия объекта сообщения m в пересылке
		0   Нет действий
RMM	21	Бит включения удаленного мониторинга объекта передачи
		0   Выключен. Идентификатор, бит IDE и поле DLC объекта сообщения m остаются без изменений до получения корректного фрейма удаленного запроса
FRREN	20	Бит разрешения удаленного запроса. Определяет, будет ли устанавливаться бит TXRQ в объекте сообщения m или в другом объекте сообщения, на который указывает CUR
		0   Бит TXRQ объекта сообщения m устанавливается после получения корректного фрейма удаленного запроса
		1   Бит TXRQ другого объекта сообщения (на который указывает CUR) устанавливается после получения им корректного фрейма удаленного запроса



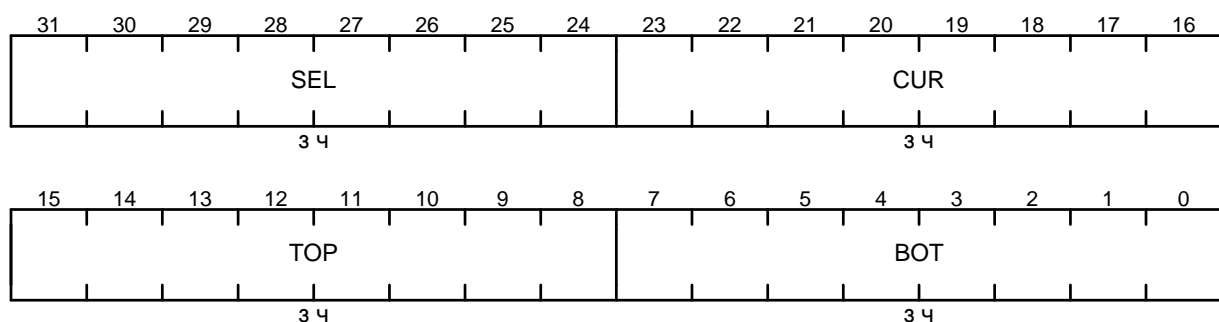
Поле	Биты	Описание
OVIE	18	Бит разрешения прерывания по заполнению FIFO объекта сообщения m. Прерывание генерируется, когда указатель CUR (указатель на текущий объект) достигает значения SEL регистра MOFGPR
		0   Запрещено
		1   Разрешено
Если объект сообщения m является объектом приема FIFO, то поле TXINP (регистр MOIPR) указывает на одну из 16 линий прерываний. Если объект сообщения m является объектом передачи FIFO, то поле RXINP (регистр MOIPR) указывает на одну из 16 линий прерываний. Для всех других режимов объекта сообщения состояние бита OVIE не важно		
TXIE	17	Бит разрешения прерывания по окончании передачи сообщения
		0   Запрещено
		1   Разрешено. Прерывание генерируется, если сообщение из объекта сообщения m было успешно передано. Поле TXINP (регистр MOIPR) указывает на одну из 16 линий прерываний
RXIE	16	Бит разрешения прерывания по окончании приема сообщения
		0   Запрещено
		1   Разрешено. Прерывание генерируется, если сообщение было успешно принято объектом сообщения m (напрямую или через шлюз). Поле RXINP (регистр MOIPR) указывает на одну из 16 линий прерываний
DATC	11	Индикатор копирования данных
		0   Данные не копируются
		1   Данные в регистрах MODATAN и MODATAL объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируются через шлюз в объект-приемник
		Бит DATC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует
DLCC	10	Индикатор копирования кода длины данных DLC
		0   Код не копируется
		1   Код длины данных объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируется через шлюз в объект-приемник
		Бит DLCC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует
IDC	9	Индикатор копирования идентификатора
		0   Идентификатор не копируется
		1   Идентификатор объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируется через шлюз в объект-приемник
		Бит IDC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует
GDFS	8	Индикатор отправки фрейма через шлюз
		0   Состояние бита TXRQ объекта-приемника без изменений
		1   Установлен бит TXRQ объекта-приемника после внутренней передачи из объекта-источника
		Бит GDFS используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует

Поле	Биты	Описание	
ММС	3-0	Задание режима объекта сообщения m	
		0000	Стандартный объект сообщения
		0001	Базовый объект приемной структуры FIFO
		0010	Базовый объект передающей структуры FIFO
		0011	Вспомогательный объект передающей структуры FIFO
		0100	Объект-источник шлюза
		Остальные комбинации зарезервированы	
–	31-28, 19, 15-12, 7-4	Зарезервировано	

### МОFGPR – регистр указателя FIFO/шлюза объекта сообщения

Смещение: Msg\_mo + 04h

Сброс: 0h

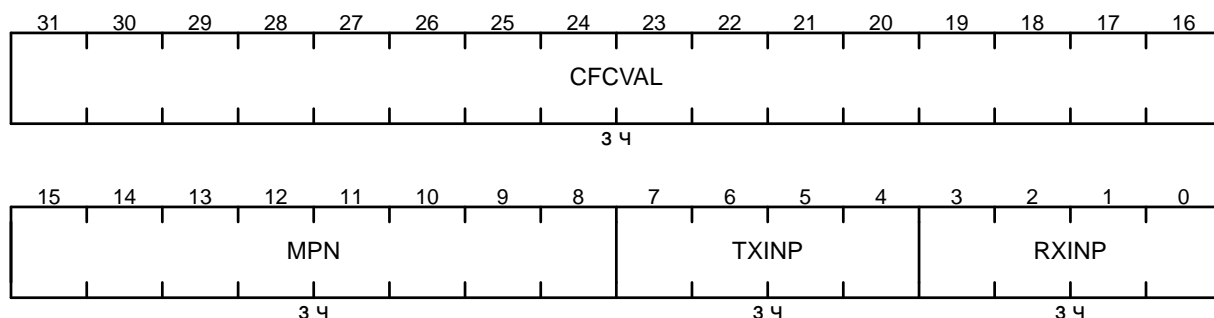


Поле	Биты	Описание
SEL	31-24	Указатель объекта сообщения. Второй (программный) указатель в дополнение к аппаратному указателю CUR при работе с FIFO. Поле SEL используется для общего мониторинга (генерирование прерываний FIFO)
CUR	23-16	Указатель на текущий объект в пределах FIFO или шлюза. После каждой операции FIFO или передачи через шлюз указатель CUR обновляется – в него заносится номер следующего объекта сообщения в списке (поле PNEXT регистра MOSTAT) – до тех пор, пока не будет достигнут верхний элемент FIFO (поле TOP), после чего CUR сбрасывается, и в него загружается номер нижнего элемента списка (из поля BOT)
TOP	15-8	Указатель верхнего элемента FIFO. В поле находится номер последнего элемента
BOT	7-0	Указатель нижнего элемента FIFO. В поле находится номер первого элемента

## МОIPR – регистр указателя прерываний объекта сообщения

Смещение: Msg\_mo + 08h

Сброс: 0h

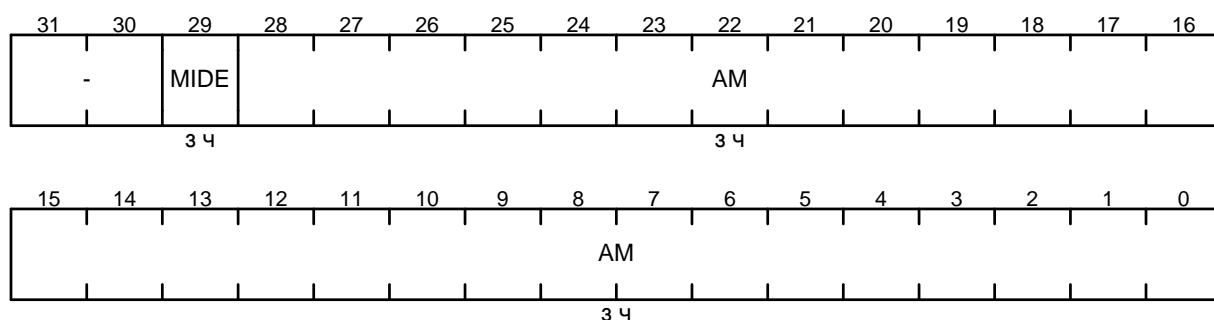


Поле	Биты	Описание
CFCVAL	31-16	Количество фреймов. Каждый раз после записи принятого сообщения в объект сообщения <i>m</i> или успешной передачи объекта сообщения <i>m</i> , значение счетчика фреймов CFC (регистр NFCR) копируется в CFCVAL
MPN	15-8	Номер ждущего бита сообщения. Указывает позицию бита, соответствующего объекту сообщения <i>m</i> в регистре MSPND
TXINP	7-4	Указатель линии прерываний для прерывания после передачи. Всего доступно 16 линий прерываний с номерами от 0 до 15. Значение 0000b, записанное в TXINP, выбирает нулевую линию прерываний, 0001b – первую, 0010b – вторую и т. д. Дополнительно бит TXINP используется для выбора позиции ждущего бита объекта сообщения <i>m</i>
RXINP	3-0	Указатель линии прерываний для прерывания после приема. Всего доступно 16 линий прерываний с номерами от 0 до 15. Значение 0000b, записанное в TXINP, выбирает нулевую линию прерываний, 0001b – первую, 0010b – вторую и т. д. Дополнительно бит RXINP используется для выбора позиции ждущего бита объекта сообщения <i>m</i>

## МОАМР – регистр маски объекта сообщения

Смещение: Msg\_mo + 0Ch

Сброс: 3FFFFFFh

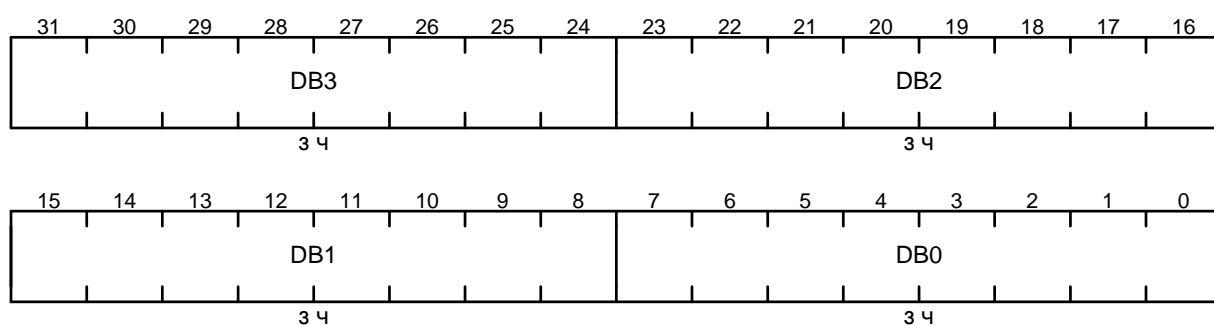


Поле	Биты	Описание
MIDE	29	Маска бита IDE сообщения
		0   Объект сообщения m может принимать как стандартные, так и расширенные фреймы
		1   Объект сообщения m может принимать только те фреймы, у которых состояние бита IDE совпадает с его битом IDE
AM	28-0	Маска идентификатора. При приеме расширенного сообщения используется вся маска. При приеме стандартного сообщения используются биты 28-18, при этом состояние битов 17-0 не важно
-	31, 30	Зарезервировано

## MODATAL – младший регистр данных объекта сообщения

Смещение: Msg\_mo + 10h

Сброс: 0h

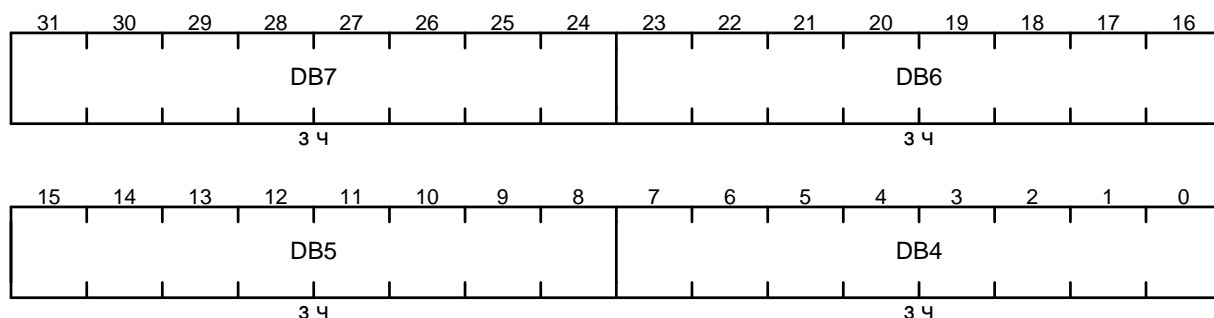


Поле	Биты	Описание
DB3	31-24	Третий байт данных
DB2	23-16	Второй байт данных
DB1	15-8	Первый байт данных
DB0	7-0	Нулевой байт данных

## MODATAN – старший регистр данных объекта сообщения

Смещение: Msg\_mo + 14h

Сброс: 0h

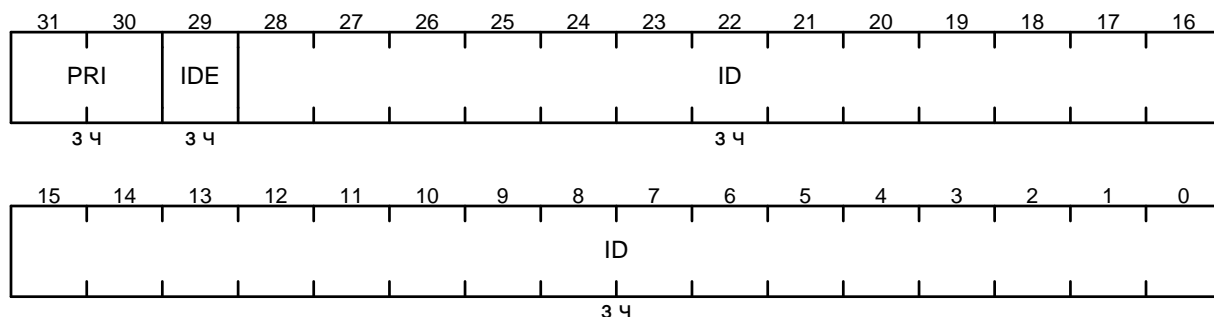


Поле	Биты	Описание
DB7	31-24	Седьмой байт данных
DB6	23-16	Шестой байт данных
DB5	15-8	Пятый байт данных
DB4	7-0	Четвертый байт данных

## MOAR – регистр арбитража объекта сообщения

Смещение: Msg\_mo + 18h

Сброс: 0h



Поле	Биты	Описание	
PRI	31-30	Класс приоритета. Поле определяет один из четырех классов (0, 1, 2 и 3) приоритета объекта сообщения m. Нулевой класс устанавливает наивысший приоритет. Объекты сообщений с нулевым классом всегда выигрывают арбитраж при передаче и приеме сообщений. Фильтрация сообщений на основе идентификатора (маскируемого) и позиции в списке организуются только для объектов сообщений с равным приоритетом. Кроме этого, поле PRI определяет метод фильтрации	
		00	Зарезервировано
		01	Фильтрация в зависимости от положения объекта сообщения в списке. Объект сообщения m получает приоритет на передачу сообщения только в случае, если нет других объектов сообщений с установленными битами MSGVAL, TXEN0 и TXEN1, стоящих выше по списку

Поле	Биты	Описание	
PRI	31-30	10	Фильтрация в зависимости от значения идентификатора. Объект сообщения <i>m</i> получает приоритет на передачу сообщения только в случае, если в списке нет других объектов сообщений с «Идентификатор + IDE + DIR» более высокого приоритета (согласно правилам арбитража в таблице А.22.2)
		11	Фильтрация в зависимости от положения объекта сообщения в списке (как при PRI = 01b)
IDE	29	Бит расширения идентификатора объекта сообщения <i>m</i>	
		0	Объект сообщения <i>m</i> оперирует с фреймами со стандартным 11-битным идентификатором
		1	Объект сообщения <i>m</i> оперирует с фреймами с расширенным 29-битным идентификатором
ID	28-0	Идентификатор объекта сообщения <i>m</i> . При оперировании с расширенными фреймами используются биты 28-0. При оперировании со стандартными фреймами используются биты 28-18, при этом состояние битов 17-0 не важно	

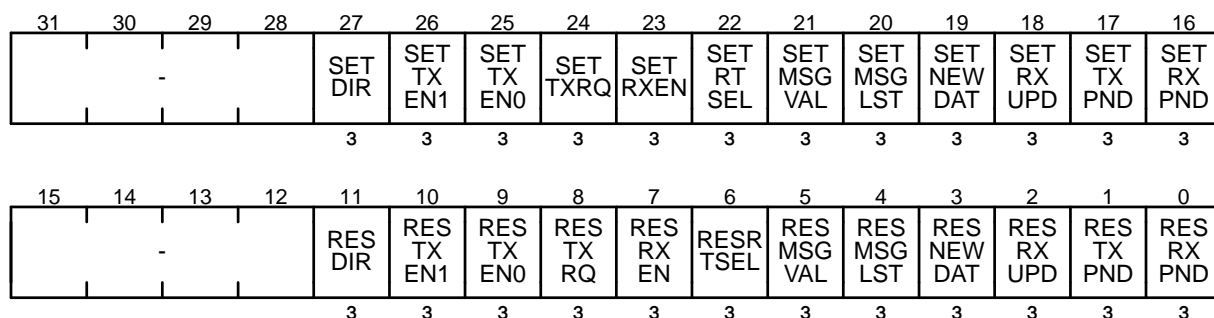
Таблица А.22.2 – Распределение приоритета между объектами сообщений согласно правилам арбитража

Установки для объектов сообщений 0 и 1, которые участвуют в арбитраже (приоритет объекта 0 выше приоритета объекта 1)	Пояснение
MOAR0[28:18] < MOAR1[28:18] (11-битный стандартный идентификатор объекта 0 меньше по числовому значению, чем 11-битный идентификатор объекта 1)	Стандартный фрейм с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом
MOAR0[28:18] = MOAR1[28:18]. В регистре MOAR0 бит IDE = 0. В регистре MOAR1 бит IDE = 1.	При равенстве значений стандартных идентификаторов, стандартный фрейм имеет приоритет перед расширенным
MOAR0[28:18] = MOAR1[28:18]. Биты IDE обоих объектов сброшены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0.	При равенстве значений идентификаторов стандартный фрейм данных имеет приоритет перед стандартным фреймом удаленного запроса
MOAR0[28:0] = MOAR1[28:0] Биты IDE обоих объектов установлены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0.	При равенстве значений идентификаторов расширенный фрейм данных имеет приоритет перед расширенным фреймом удаленного запроса
MOAR0[28:0] < MOAR1[28:0] Биты IDE обоих объектов установлены. (29-битный идентификатор объекта 0 меньше по числовому значению, чем 29-битный идентификатор объекта 1)	Расширенный фрейм с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом

## МОСТР – регистр управления объектом сообщения

Смещение: Msg\_mo + 1Ch

Сброс: 0h



Поле	Биты	Описание
RESRXPND, SETRXPND	0, 16	Сброс/установка бита RXPND
RESTXPND, SETTXPND	1, 17	Сброс/установка бита TXPND
RESRXUPD, SETRXUPD	2, 18	Сброс/установка бита RXUPD
RESNEWDAT, SETNEWDAT	3, 19	Сброс/установка бита NEWDAT
RESMSGVAL, SETMSGVAL	4, 20	Сброс/установка бита MSGVAL
RESMSGVAL, SETMSGVAL	5, 21	Сброс/установка бита MSGVAL
RESRTSEL, SETRTSEL	6, 22	Сброс/установка бита RTSEL
RESRXEN, SETRXEN	7, 23	Сброс/установка бита RXEN
RESTXRQ, SETTXRQ	8, 24	Сброс/установка бита TXRQ
RESTXEN0, SETTXEN0	9, 25	Сброс/установка бита TXEN0
RESTXEN1, SETTXEN1	10, 26	Сброс/установка бита TXEN1
RESDIR, SETDIR	11, 27	Сброс/установка бита DIR
–	15-12, 31-28	Зарезервировано. При чтении возвращаются нули. При записи следует писать 0h.

Примечание – Биты с префиксом SET и RES работают попарно. Комбинация состояний этих бит оказывает влияние на соответствующий бит регистра MOSTAT:

- SET\* = 1, RES\* = 0 устанавливает бит \*;
- SET\* = 0, RES\* = 1 сбрасывает бит \*;
- SET\* = RES\* = 0 или SET\* = RES\* = 1 не изменяет состояние бита \*.

## MOSTAT – регистр состояния объекта сообщения

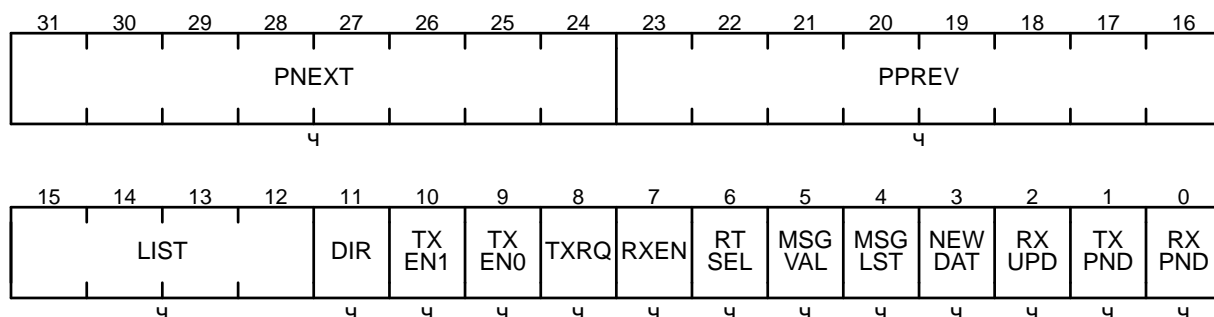
Смещение: Msg\_mo + 1Ch

Сброс:

- 0100\_0000h для объекта сообщения 0;

- [mo + 1][mo - 1][0000]h для объектов сообщений с номерами от 1 до 254, (соответственно mo от 01h до FEh);

- FFFE\_0000h для объекта сообщения 255;



Поле	Биты	Описание
PNEXT	31-24	Указатель на следующий элемент списка. В поле находится номер объекта сообщения, расположенного выше по списку относительно текущего
PPREV	23-16	Указатель на предыдущий элемент списка. В поле находится номер объекта сообщения, расположенного ниже по списку относительно текущего
LIST	15-12	Номер списка, которому принадлежит объект сообщения m. Поле обновляется аппаратно при распределении/перераспределении объекта сообщения
DIR	11	Бит распределения
		0   Объект приема сообщения данных. Объект принимает сообщение данных. При установленном бите TXRQ объект формирует сообщение удаленного запроса с идентификатором объекта сообщения m, а затем передает его. Полученное в ответ сообщение данных с соответствующим идентификатором сохраняется в объекте сообщения m
		1   Объект передачи сообщения данных. При установленном бите TXRQ объект формирует, а затем передает сообщение данных. Если объект сообщения m получает сообщение удаленного запроса с соответствующим идентификатором, то устанавливается флаг TXRQ его регистра MOSTAT, после чего в ответ передается сообщение данных, содержащихся в объекте сообщения m
TXEN1	10	Бит разрешения передачи фрейма
		0   Запрещено
		1   Передача фрейма разрешена. Объект сообщения m может участвовать в передаче только, если установлены оба бита – TXEN1 и TXEN0. Контроллер CAN использует бит TXEN1 для выбора активного объекта передачи сообщения из FIFO



Поле	Биты	Описание
TXEN0	9	Бит разрешения передачи фрейма
		0   Запрещено
		1   Передача фрейма разрешена. Объект сообщения m может участвовать в передаче, только если установлены оба бита TXEN0 и TXEN1. Контроллер CAN использует бит TXEN1 для выбора активного объекта передачи сообщения из FIFO. Можно программно очищать бит TXEN0 для запрета передачи сообщения, которое в настоящий момент формируется, или для запрета автоматической передачи в ответ на удаленный запрос
TXRQ	8	Бит инициации передачи
		0   Нет действий
		1   Установка бита инициирует передачу фрейма из объекта сообщения m. Инициация передачи фрейма возможна только в случае, если установлены биты TXRQ, TXEN0, TXEN1 и MSGVAL. Также бит TXRQ устанавливается аппаратно при получении фрейма удаленного запроса. Бит сбрасывается аппаратно при успешном завершении передачи, и если при этом не был повторно программно установлен бит NEWDAT
RXEN	7	Бит разрешения приема
		0   Запрещено
		1   Объект сообщения может принимать сообщения Состояние бита учитывается только при фильтрации принимаемых сообщений
RTSEL	6	Индикатор возможности приема/передачи
		0   Объект сообщения не может принимать/передавать сообщения
		1   Объект сообщения может принимать/передавать сообщения Прием фрейма. Бит RTSEL устанавливается аппаратно после того, как выбран объект сообщения m для сохранения только что принятого фрейма. Прежде, чем записать принятые данные в объект сообщения m, аппаратная часть проверяет состояние бита RTSEL. ЦПУ может сбрасывать этот бит, чтобы запретить запись принятого фрейма в объект сообщения m. Передача фрейма. Бит RTSEL устанавливается аппаратно после того, как выбран следующий объект сообщения m для передачи фрейма. Аппаратная часть перед началом передачи проверяет: установлен ли бит RTSEL и сброшен ли бит NEWDAT. Бит RTSEL должен оставаться установленным до окончания передачи. Проверка состояния бита RTSEL производится только при попытке изменения содержимого объекта сообщения m во избежание одновременного выполнения операций передачи фрейма и его изменения. Бит не участвует в фильтрации сообщений и не сбрасывается аппаратно
MSGVAL	5	Бит активности объекта сообщения m
		0   Не активен
		1   Активен Только те объекты сообщений, для которых установлен этот бит, могут использоваться для операций приема и передачи

Поле	Биты	Описание	
MSGLST	4	Бит потери сообщения	
		0	Ни одно сообщение не потеряно
		1	Принятое сообщение потеряно вследствие того, что контроллер CAN попытался установить бит NEWDAT по окончании приема сообщения при том, что флаг NEWDAT уже был установлен ранее после записи другого сообщения
NEWDAT	3	Индикатор новых данных	
		0	С момента сброса бита NEWDAT никаких изменений объекта сообщения m не обнаружено
		1	Объект сообщения был изменен. Бит устанавливается аппаратно после того, как принятое сообщение было сохранено в объекте сообщения m. Бит сбрасывается аппаратно после начала передачи объекта сообщения m. Бит NEWDAT следует устанавливать программно после того, как новые данные для передачи будут сохранены в объекте сообщения m для предотвращения автоматического сброса бита TRXQ в конце текущей передачи
RXUPD	2	Индикатор изменений	
		0	Нет текущих изменений
		1	Идентификатор сообщения, поле длины данных DLC и данные в объекте сообщения изменяются
TXPND	1	Индикатор окончания передачи	
		0	Переданных сообщений нет
		1	Сообщение объекта m было успешно передано
RXPND	0	Индикатор окончания приема	
		0	Принятых сообщений нет
		1	Сообщение было успешно принято объектом сообщения m (напрямую или через шлюз). Бит должен сбрасываться программно

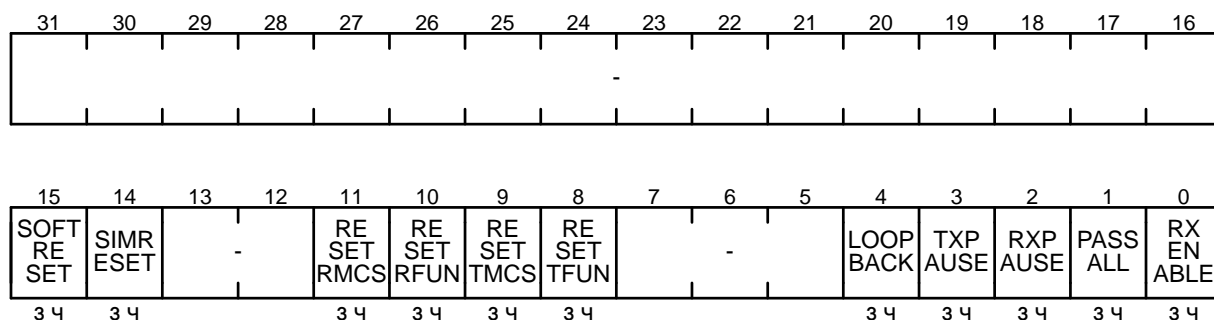
## A.23 Регистры контроллера интерфейса Ethernet 10/100

Базовый адрес: 4003\_0000h

### MAC1 – регистр 1 конфигурации MAC

Смещение: + 00h

Сброс: 8000h



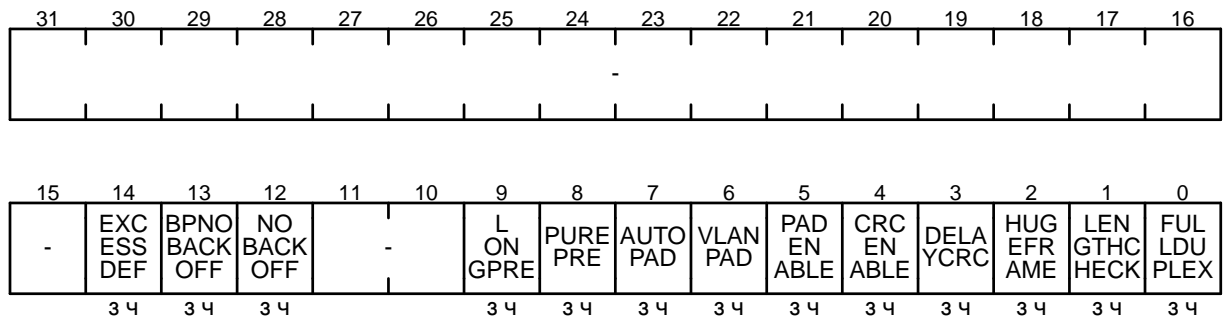
Поле	Биты	Описание
SOFTRESET	15	Бит сброса блока MAC контроллера Ethernet 10/100 кроме блока связи с процессором
		0   Нет действий
		1   Установка бита сбрасывает блок MAC
SIMRESET	14	Бит сброса генератора случайных чисел устройства передачи
		0   Нет действий
		1   Установка бита сбрасывает генератор
RESETRMCS	11	Бит сброса устройств MAC-уровня, отвечающих за управление фильтрацией адресов при приеме пакетов (контроль домена)
		0   Нет действий
		1   Установка бита сбрасывает устройства MAC-уровня
RESETRFUN	10	Бит сброса логики устройств приема пакетов
		0   Нет действий
		1   Установка бита сбрасывает логику устройств приема пакетов
RESETTMCS	9	Бит сброса устройств MAC-уровня, отвечающих за управление адресами при передаче информации (контроль домена)
		0   Нет действий
		1   Установка бита сбрасывает устройства MAC-уровня
RESETTFUN	8	Бит сброса логики устройств передачи пакетов
		0   Нет действий
		1   Установка бита сбрасывает логику устройств передачи пакетов
LOOPBACK	4	Бит активации приема передаваемых пакетов обратно через MACReceive-интерфейс. Очистка бита не вызывает прием передаваемых пакетов обратно
		0   Нет действий
		1   Установка бита вызывает прием передаваемых пакетов

Поле	Биты	Описание
TXPAUSE	3	Бит разрешения передачи пауз в Frame.
		0   Передача пауз в Frame заблокирована
		1   Разрешено
RXPAUSE	2	Бит разрешения приема паузы как части Frame
		0   Пауза игнорируется
		1   Разрешено
PASSALL	1	Бит управления PASS
		0   PASS подтверждается для текущего принимаемого Frame при успешной передаче Frame
		1   Блок MAC выдает PASS для текущего принимаемого Frame независимо от его типа (т. е. для всех Frame)
RXENABLE	0	Бит разрешения приема Frame
		0   Frame не принимаются
		1   Frame принимаются. Внутренняя MAC-синхронизация использует этот бит для приема потока и выхода SYNCHRONIZED RECEIVE ENABLE, используемого MAC для уточнения принимаемого фрейма
-	31-16, 13-12, 7-5	Зарезервировано

### MAC2 – регистр 2 конфигурации MAC

Смещение: + 04h.

Сброс: 0h



Поле	Биты	Описание
EXCESSDEF	14	0   Обработка пакета прерывается в случае избыточной задержки, которая превышает установленный лимит
		1   Блок MAC будет неопределенно долго осуществлять обработку пакета (в соответствии со стандартом)
BPNOBACKOFF	13	Бит разрешения повторной передачи при коллизии
		0
		1   В случае коллизии повторная передача будет повторена немедленно, без периода ожидания, с целью повышения вероятности успешной передачи пакета

Поле	Биты	Описание
NOBACKOFF	12	Бит задания параметра повторной передачи при коллизии
		0
		1 В случае коллизии период ожидания до повторной передачи определяется бинарно-экспоненциальным алгоритмом в соответствии со стандартом IEEE 802-3
LONGPRE	9	Бит выбора длины преамбулы принимаемых пакетов
		0 Блок MAC допускает прием пакетов с преамбулами любой длины, как в стандарте
		1 Блок MAC позволяет принимать только пакеты, преамбула в которых менее 12 байт
PUREPRE	8	Бит включения проверки преамбулы
		0 Проверка преамбулы не проводится
		1 Блок MAC верифицирует содержимое преамбулы на содержание ошибок. При обнаружении ошибки в преамбуле обработка пакета прекращается
AUTOPAD	7	Бит включения автоматического определения типа Frame. Бит игнорируется, если бит PADENABLE сброшен (см. таблицу A.23.1)
		0
		1 Блок MAC автоматически определяет тип Frame (теговый или не теговый), сравнивая 2 октета адреса последующего источника с 8100h (VLAN протокол ID) или PAD, соответственно
VLANPAD	6	Бит включения дополнения коротких Frame. Бит игнорируется, если бит PADENABLE сброшен (см. таблицу A.23.1)
		0
		1 Блок MAC добавляет PAD ко всем коротким Frame, доводя их длину до 64 байт, и присоединяет значение CRC
PADENABLE	5	Бит разрешения функционирования битов AUTOPAD и VLANPAD
		0 Бит следует сбросить, если все Frame имеют правильную длину
		1 Блок MAC добавляет PAD во все короткие Frame. Этот бит используется совместно с битами AUTO PAD ENABLE или VLAN PAD ENABLE (см. таблицу A.23.1)
CRCENABLE	4	Бит разрешения вставки CRC
		0 Бит следует сбросить, если CRC уже включен во Frame
		1 CRC автоматически добавляется к Frame (в PAD), независимо от того, требовалось это или нет
DELAYCRC	3	Бит разрешения добавления 4 байт CRC
		0 Бит следует сбросить, если добавление CRC отсутствует
		1 4 байта CRC поля служебной информации, которые расположены в начале Frame, добавляются при передаче и вырезаются при приеме

Поле	Биты	Описание
HUGEFRAME	2	Бит разрешения приема и передачи Frame произвольной длины
		0   Запрещено
		1   Разрешено
LENGTHCHECK	1	Бит разрешения компарации длины Frame
		0
		1   Длина передаваемого и принимаемого Frame компарируется с полем Length/TypeFrame. При успешной компарации – проверка выполнена. О несовпадении выдается сообщение в Transmit/Receive Statistics Vector
FULLDUPLEX	0	Бит выбора режима выполнения MAC-операций
		0   Half- Duplex
		1   Full-Duplex
–	31-15, 11-10	Зарезервировано

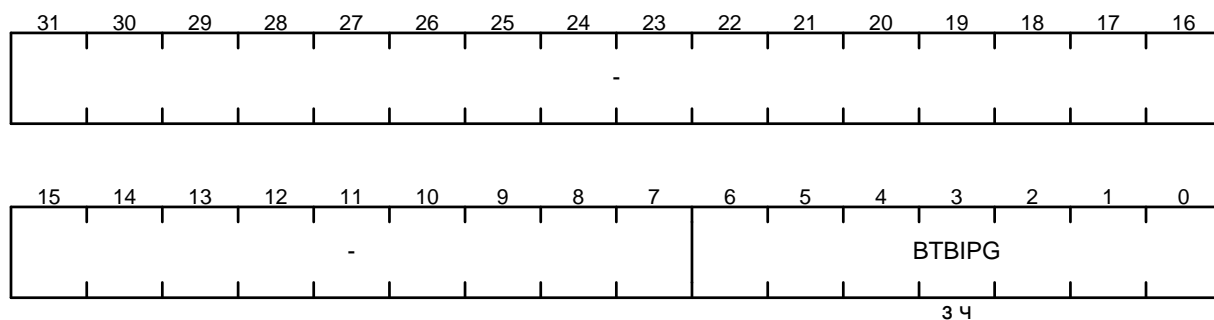
Таблица А.23.1 – PAD-операции

Тип	Биты регистра MAC2			Вид Frame
	AUTOPAD	VLANPAD	PADENABLE	
Все	X	X	0	Без PAD, проверка CRC
Все	0	0	1	PAD 60 бит, присоединено CRC
Все	X	1	1	PAD 64 бит, присоединено CRC
Все	1	0	1	Если безтеговый: PAD 60 бит, присоединено CRC. Если VLAN – теги: PAD 64 бит, присоединено CRC

## IPGT – регистр Back-to-Back Inter-Packet-Gap

Смещение: + 08h

Сброс: 0h

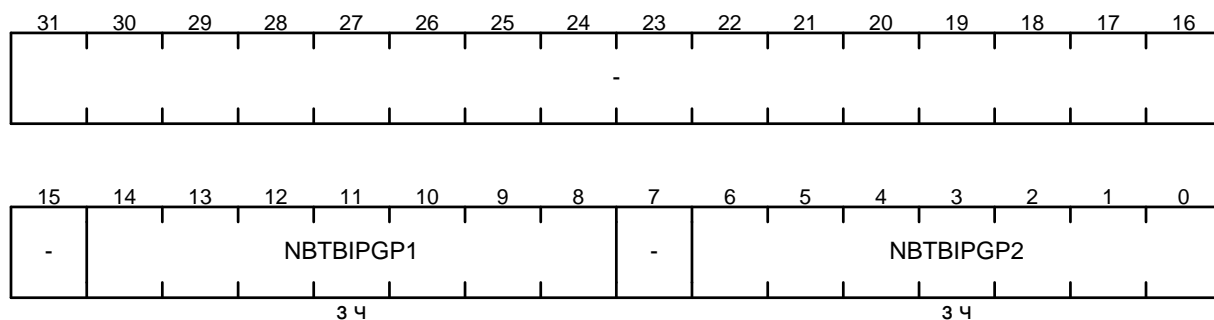


Поле	Биты	Описание
BTBIPG	6-0	Поле задания минимального времени от окончания передачи последнего полубайта пакета до начала передачи следующего пакета. В режиме Full-Duplex значение этого поля следует задавать равным периоду передачи полубайта минус три. Рекомендуемое значение – 15h, которое обеспечивает минимальный IPG равный 0,96 мкс для 100 Мбит/с и 9,6 мкс для 10 Мбит/с. В Half-Duplex режиме значение поля следует задавать равным периоду передачи полубайта минус шесть. Рекомендуемое значение – 12h, которое обеспечивает минимальный IPG равный 0,96 мкс для 100 Мб/с и 9,6 мкс для 10 Мб/с
–	31-7	Зарезервировано

## IPGR – регистр Non-Back-to-Back Inter-Packet-Gap

Смещение: + 0Ch

Сброс: 0h



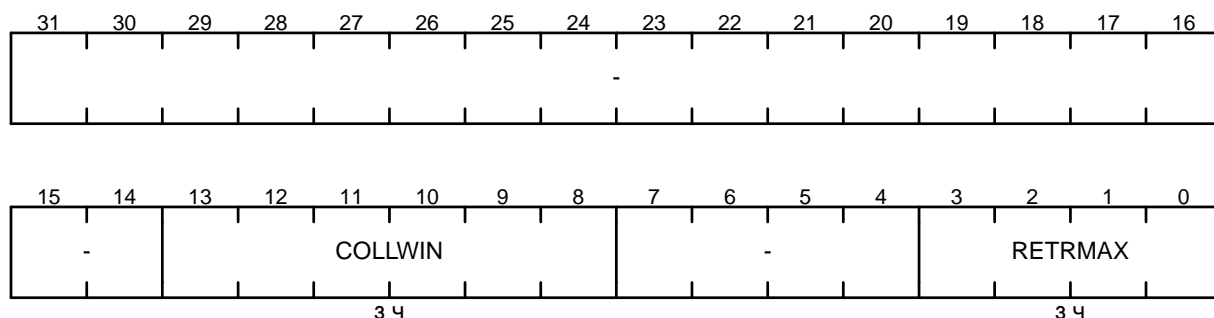
Поле	Биты	Описание
NBTVIPGP1	14-8	Поле представляет собой опцию carrierSense (определение несущей частоты), описанную в IEEE 802.3/4.2.3.2.1 «Carrier Deference». Если несущая частота идентифицирована во время синхронизации IPGR1, то блок MAC подстраивается под несущую частоту. Но, если несущая частота определена после IPGR1, то блок MAC в течении IPGR2 продолжает передачу, принудительно создавая коллизию, и таким образом, обеспечивая явный доступ к передающей среде. Поле может содержать значения 0h до NBTVIPGP2. Рекомендуемое значение – 0Ch
NBTVIPGP2	6-0	Поле представляет собой non-back-to-back Inter-Packet-Gap. Рекомендуемое значение – 12h (обеспечивает минимальное IPG равное 0,96 мкс для 100 Мб/с и 9,6 мкс для 10 Мб/с).
–	31-15, 7	Зарезервировано



### CLRT – регистр окна коллизий/повторов

Смещение: + 10h

Сброс: 370Fh

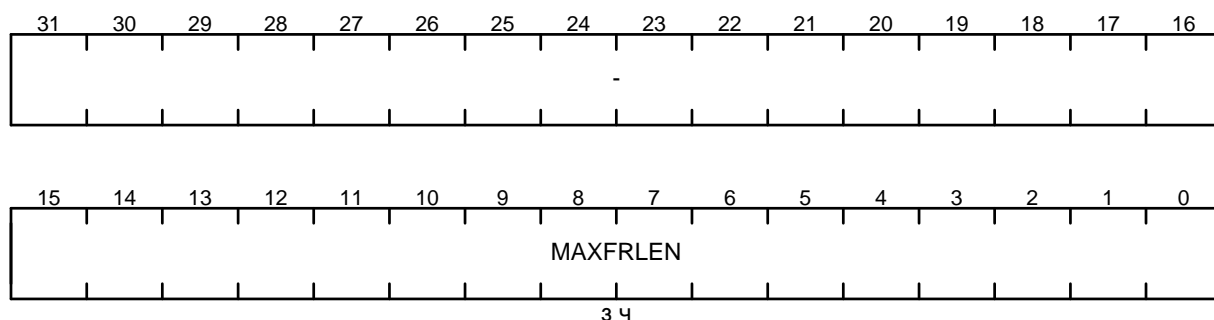


Поле	Биты	Описание
COLLWIN	13-8	Поле содержит время слота или окна коллизий, во время которого возможны коллизии в конфигурированных сетях. Окно коллизий начинается с начала преамбулы, включая SFD. Значение окна коллизий, по умолчанию 37h, что соответствует числу байт в Frame в конце окна
RETRMAX	3-0	Поле задания числа попыток повторной передачи после коллизии, прежде чем передача пакета будет прервана из-за превышения числа коллизий. Рекомендуемое значение Fh 9,6 мкс для 10 Мб/с).
–	31-14, 7-4	Зарезервировано

### MAXF – регистр верхней границы размера Frame

Смещение: + 14h

Сброс: 0000\_0600h

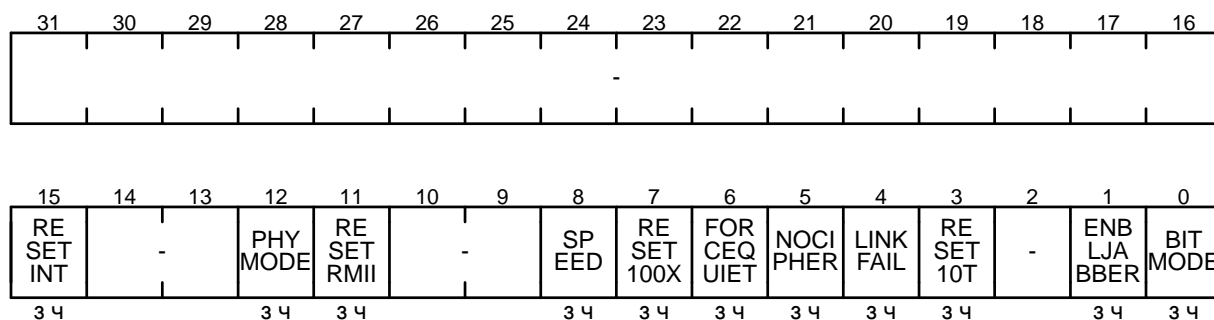


Поле	Биты	Описание
MAXFLEN	15-0	Поле содержит время слота или окна коллизий, во время которого возможны коллизии в конфигурированных сетях. Окно коллизий начинается с начала преамбулы, включая SFD. Значение окна коллизий, по умолчанию 37h, что соответствует числу байт в Frame в конце окна
–	31-16	Зарезервировано

## SUPP – регистр поддержки PHY-интерфейса

Смещение: + 18h

Сброс: 1000h



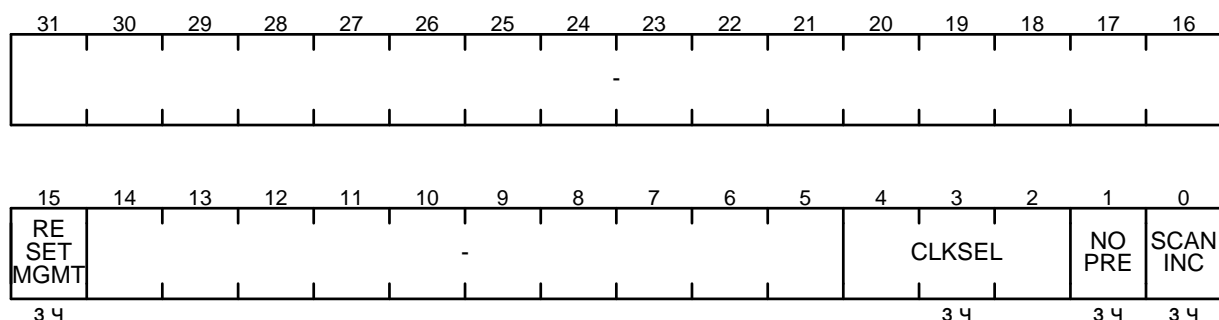
Поле	Биты	Описание
RESEINT	15	Бит сброса физического интерфейса. Если подключен один модуль физического интерфейса, то бит можно использовать вместо битов RESETRMII, RESET100X и RESET10T
		0   Нормальные операции с модулем физического интерфейса
		1   Установка сбрасывает выбранный модуль физического интерфейса (устройство связи с физической линией)
PHYMODE	12	Бит конфигурации последовательного МП с подключенными SМП-устройствами (используется при подключении SМПPHY)
		0   Активны функции SМПMAC. Если выбран SМПMAC, то операции приема/передачи пакетов выполняются в режиме Full Duplex на частоте 100 Мбит/с
		1
RESETRMII	11	Бит сброса логики упрощенного МП
		0   Нет действий
		1   Установка бита активирует сброс
SPEED	8	Бит задания рабочей скорости упрощенного МП
		0   10 Мбит/с
		1   100 Мбит/с
RESET100X	7	Бит сброса модуля, который содержит логику шифратора/дешифратора символов 4 бит/5 бит (4В/5В)
		0   Нет действий
		1   Установка бита активирует сброс
FORCEQUIET	6	Бит включения шифрации
		0   Выполняется нормальная операция (без шифрации)
		1   На выход передаются зашифрованные (4В/5В) данные
NOCIPHER	5	Бит выбора режима шифрования
		0   Передача с нормальным шифрованием
		1   Передача 5В символов без шифрования
LINKFAIL	4	Бит разрешения моделирования
		0   Выполняется нормальная операция
		1   330 мс LinkFail таймер отключается для разрешения короткого моделирования

Поле	Биты	Описание
RESET10T	3	Бит сброса модуля, который преобразует потоки полубайт МП в последовательный поток двоичных сигналов приемопередатчика режима 10T
		0   Нет действий
		1   Установка бита активирует сброс
ENBLJABER	1	Бит включения защиты от неправильных данных при передаче в режиме 10T ENDEC. Условием защиты является передача одного значения в линию более 50 мс и служит для устранения этого условия для передач другими станциями
		0   Выключено
		1   Включено
BITMODE	0	Блок активации режима 10BASE-T ENDEC
		0
		1   Блок MAC работает в режиме 10BASE-T ENDEC, при котором синхронизация происходит по фактическим данным битов, а не на основе тактового генератора полубайт
–	31-16, 14-13, 10-9, 2	Зарезервировано

## MCFG – регистр управления конфигурацией МП

Смещение: + 20h

Сброс: 0h

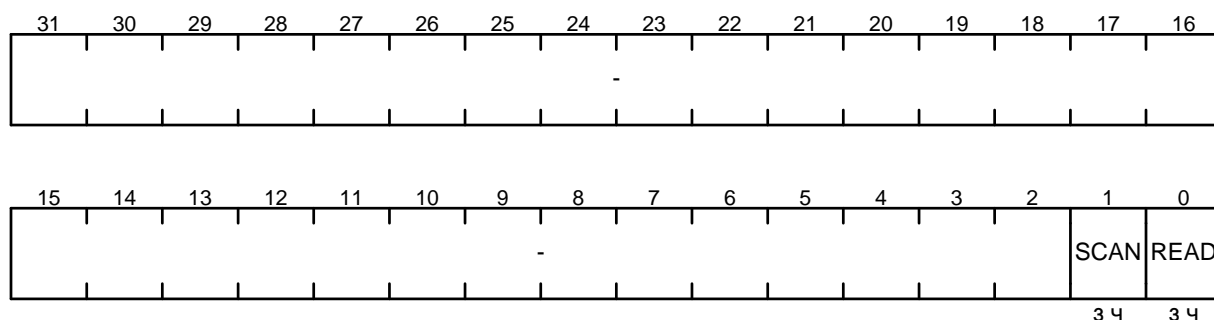


Поле	Биты	Описание
RESETMGMT	15	Бит сброса модуля управления МП
		0   Нет действий
		1   Установка бита активирует сброс
CLKSEL	4-2	Поле выбора делителя частоты. Делитель используется при делении частоты для формирования управляющей частоты (МДС) модуля МП, значение которой определено в стандарте IEEE 802.3и как не превышающее 2,5 МГц Примечание – Некоторые PHY поддерживают частоту до 12,5 МГц.
		000, 001   1/4
		010   1/6
		011   1/8
		100   1/10
		101   1/14
		110   1/20
		111   1/28
NOPRE	1	Бит управления преамбулой
		0   Поддерживаются нормальные циклы чтения/записи. Некоторые PHY поддерживают работу без преамбулы
		1   Модуль управления МП выполняет циклы чтения/записи без 32-битного поля преамбулы
SCANINC	0	Бит определяет режим чтения адресов PHY
		0   Непрерывное чтение одного PHY
		1   Модуль МП выполняет чтение всех адресов PHY, начиная с адреса 1 (поле возможных адресов, биты с 4 по 0)
–	31-16, 14-5	Зарезервировано

## MCMD – регистр команд МП

Смещение: + 24h

Сброс: 0h

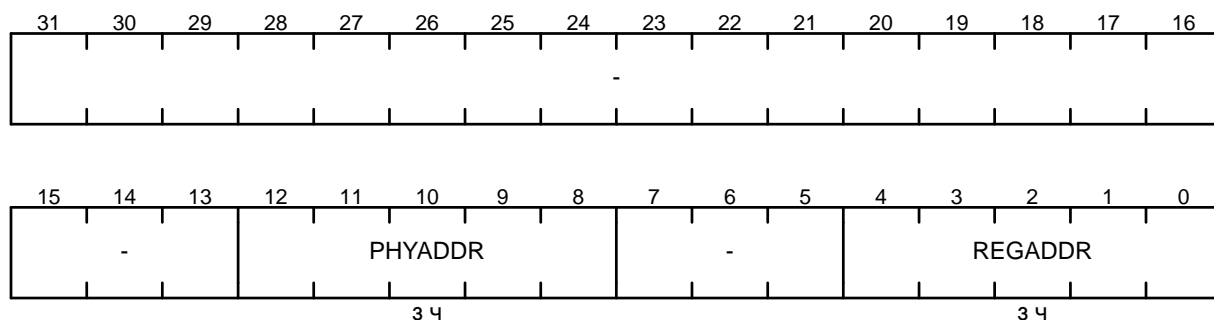


Поле	Биты	Описание		
SCAN	1	Непрерывное выполнение циклов чтения		
		<table border="1"> <tr> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>Модуль управления МП выполняет циклы чтения непрерывно. Это полезно, к примеру, для мониторинга сбоя LinkFail</td> </tr> </table>	0	
0				
1	Модуль управления МП выполняет циклы чтения непрерывно. Это полезно, к примеру, для мониторинга сбоя LinkFail			
READ	0	Однократное выполнение циклов чтения		
		<table border="1"> <tr> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>Модуль управления МП выполняет одиночные циклы чтения. Прочитанные данные возвращаются в регистр MRDD</td> </tr> </table>	0	
0				
1	Модуль управления МП выполняет одиночные циклы чтения. Прочитанные данные возвращаются в регистр MRDD			
–	31-2	Зарезервировано		

## MADR – регистр адреса МП

Смещение: + 28h

Сброс: 0h

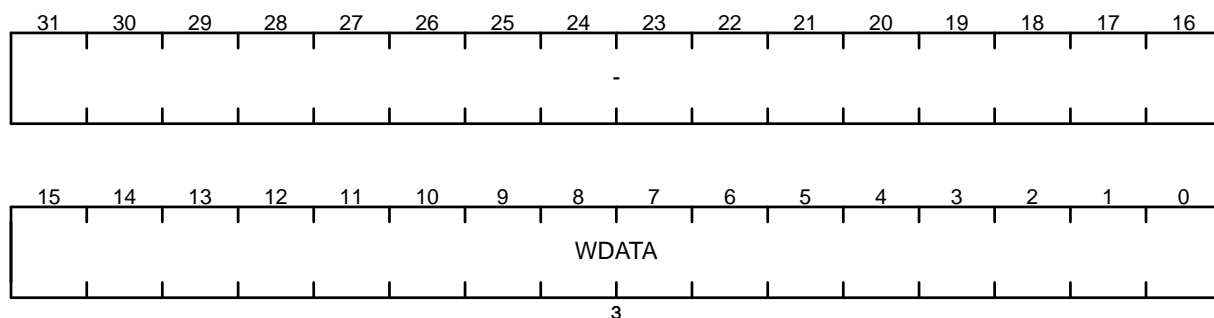


Поле	Биты	Описание
PHYADDR	12-8	Поле содержит 5-битный адрес PHY-устройства для циклов управления от МП. Можно адресовать до 31 PHY. Значение 00h является зарезервированным
REGADDR	4-0	Поле содержит 5-битный адрес регистра PHY-устройства для циклов управления от МП. Можно адресовать до 32 регистров
–	31-13, 7-5	Зарезервировано

### MWTD – регистр записываемых данных в МП

Смещение: + 2Ch

Сброс: 0h

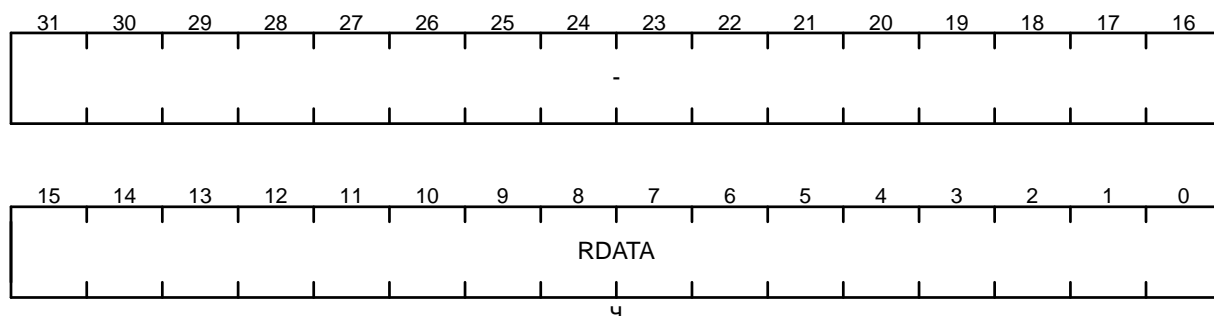


Поле	Биты	Описание
WDATA	15-0	Поле данных. При записи модуль управления МП использует данные этого регистра и предварительно сконфигурированные адрес РНУ и МП для записи в РНУ устройство
–	31-16	Зарезервировано

### MRDD – регистр считываемых данных из МП

Смещение: + 30h.

Сброс: 0h

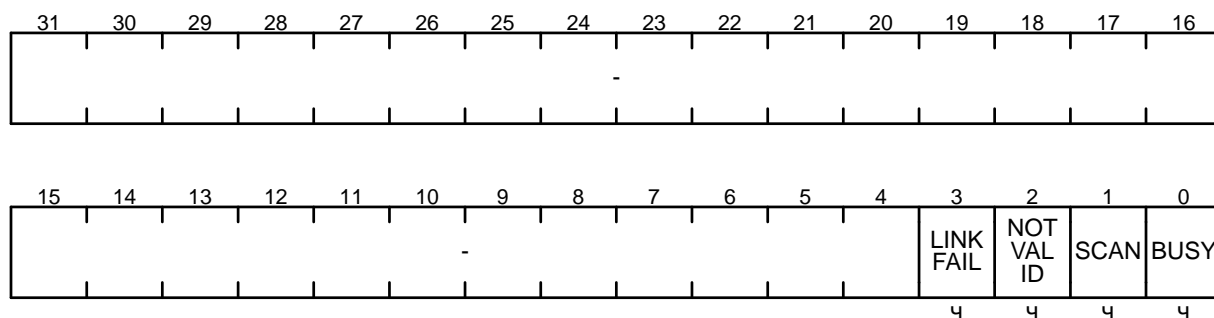


Поле	Биты	Описание
RDATA	15-0	После чтения модулем управления МП из РНУ-устройства прочитанные данные можно считать из этого регистра (по адресу 0070_000Ch)
–	31-16	Зарезервировано

## MIND – регистр флагов состояний МП

Смещение: + 34h

Сброс: 0h

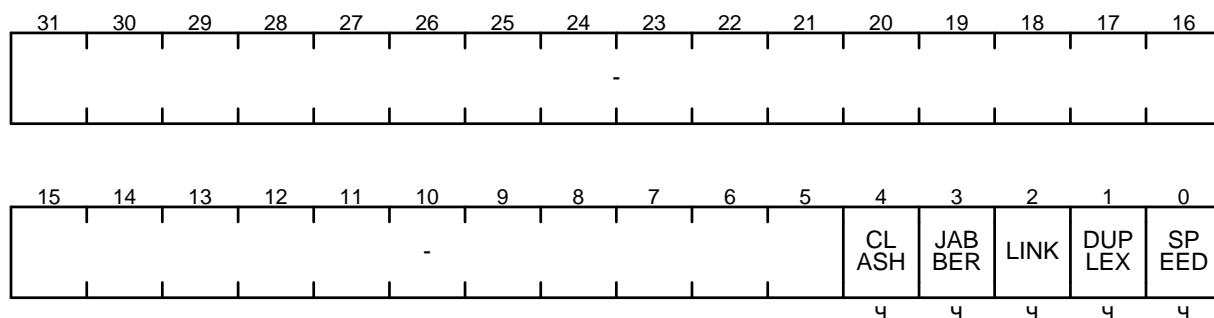


Поле	Биты	Описание
LINKFAIL	3	Флаг сбоя управления МП (управление РНУ-устройством)
		0   Нормальная работа 1   Сбой
NOTVALID	2	Флаг незавершенного цикла чтения МП (недоступности данных для чтения)
		0   1   Цикл чтения не завершен
SCAN	1	Флаг продолжающейся скан-операции (продолжаются циклы чтения управления МП)
		0   1   Скан-операция в процессе
BUSY	0	Флаг продолжающегося цикла чтения/записи управления МП
		0   1   Чтение/запись в процессе
–	31-4	Зарезервировано

## SMII – регистр состояния контроллера MII

Смещение: + 38h

Сброс: 0h



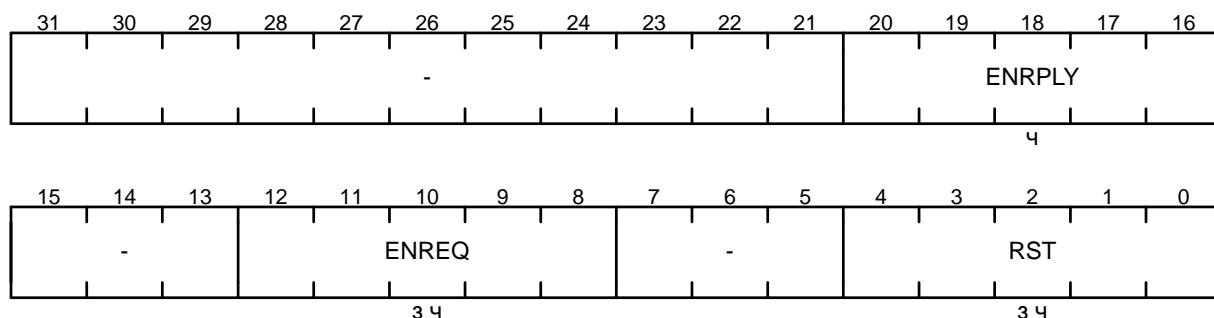
Поле	Биты	Описание
CLASH	4	Флаг выбранного режима
		0
JABBER	3	1 МАС-to-МАС, за исключением, если обнаружен PHY
		Флаг возникновения условия Jabber
LINK	2	0 Нет
		1 Условие обнаружено
DUPLEX	1	Индикатор состояния LINK
		0 Выключено
SPEED	0	1 Включено
		Индикатор режима
-	31-5	0 Half-Duplex
		1 Full-Duplex
-	-	Индикатор скорости
		0 10 Мбит/с
-	-	1 100 Мбит/с
		Зарезервировано



## FIFOCFG – регистр конфигурации буфера MIFIFO

Смещение: + 3Ch

Сброс: 1Fh

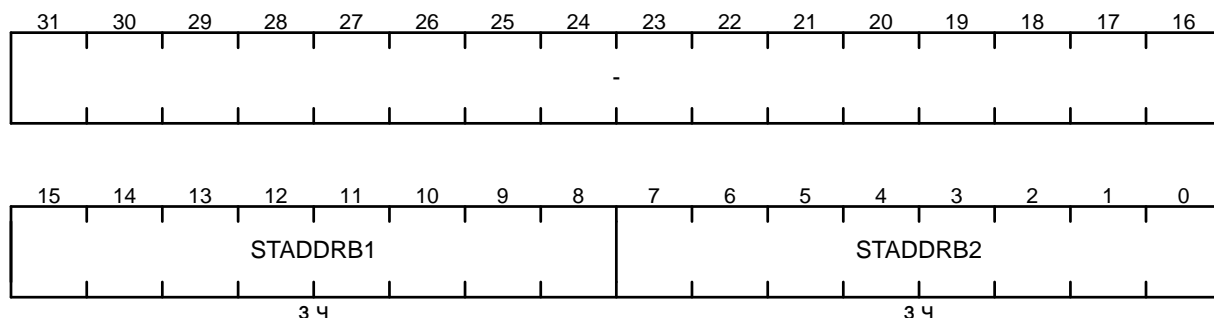


Поле	Биты	Описание
ENRPLY	20-16	Индикация работы модулей MIFIFO
		00h   Отключены
		1Fh   Включены
ENREQ	12-8	Разрешение запросов модуля MIFIFO
		00h   Запрещены
		1Fh   Разрешены
RST	4-0	Сброс модулей MIFIFO
		00h   Модули функционируют
		1Fh   Переключение в режим сброса модулей
-	31-21, 15-13, 7-5	Зарезервировано

## SA0 – регистр 0 адреса станции

Смещение: + 40h.

Сброс: 0h

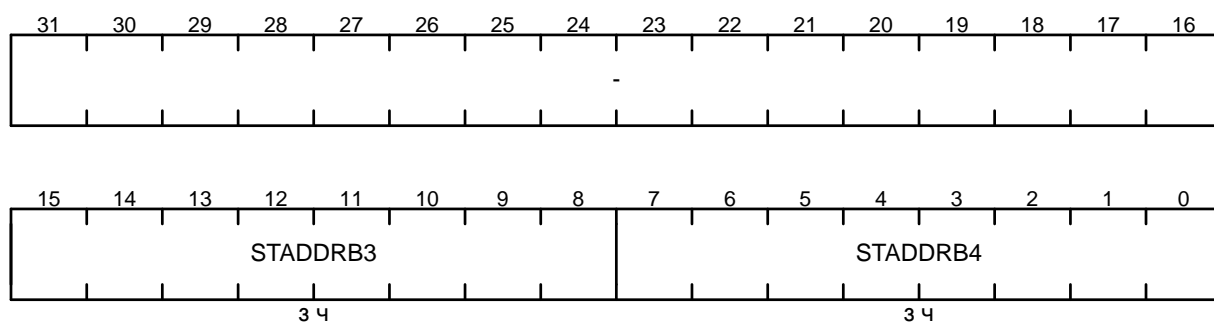


Поле	Биты	Описание
STADDRB1	15-8	Первый октет адреса станции
STADDRB2	7-0	Второй октет адреса станции
-	31-16	Зарезервировано

### SA1 – регистр 1 адреса станции

Смещение: + 44h

Сброс: 0h

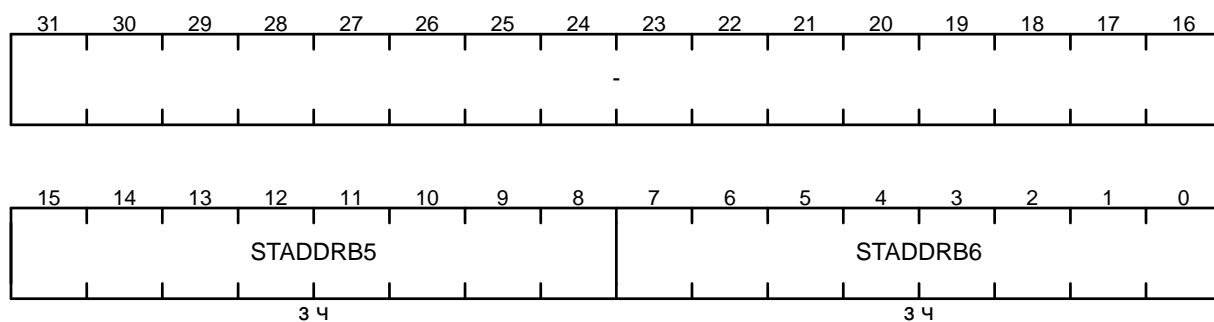


Поле	Биты	Описание
STADDRB3	15-8	Третий октет адреса станции
STADDRB4	7-0	Четвертый октет адреса станции
–	31-16	Зарезервировано

### SA2 – регистр 2 адреса станции

Смещение: + 48h.

Сброс: 0h

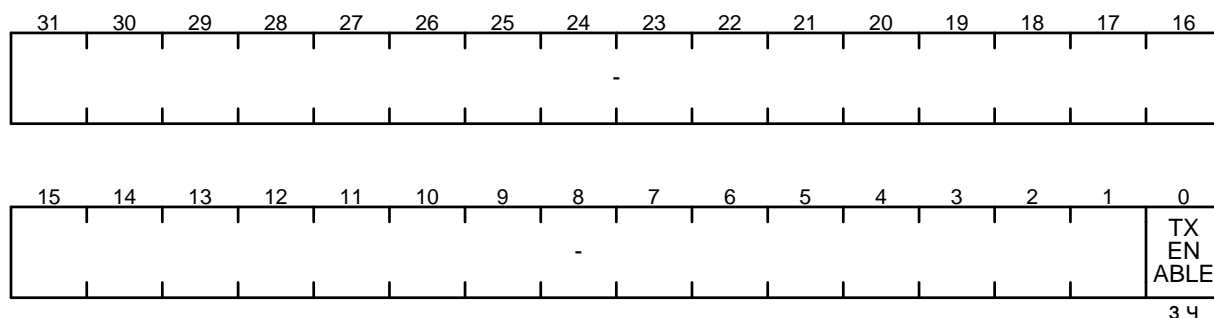


Поле	Биты	Описание
STADDRB5	15-8	Пятый октет адреса станции
STADDRB6	7-0	Шестой октет адреса станции
–	31-16	Зарезервировано

## DMATXCTRL – регистр управления передачей

Смещение: + 180h

Сброс: 0h

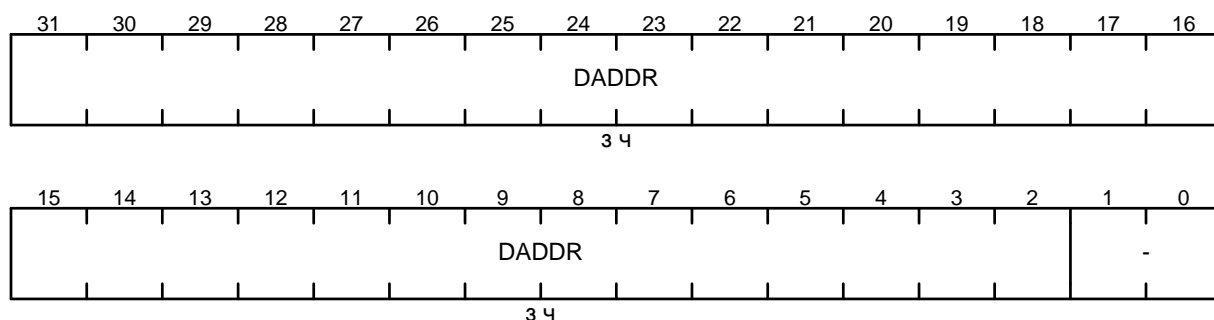


Поле	Биты	Описание
TXENABLE	0	Бит разрешения устройству ПДП доступа к пакету, предназначенному для передачи. Бит очищается встроенным ПДП-контроллером, если происходит неполная передача или возникает состояние ошибки в передающей линии
		0   Запрещено
		1   Разрешено
–	31-1	Зарезервировано

## DMATXDESCR – регистр указателя дескриптора передачи

Смещение: + 184h

Сброс: 0h

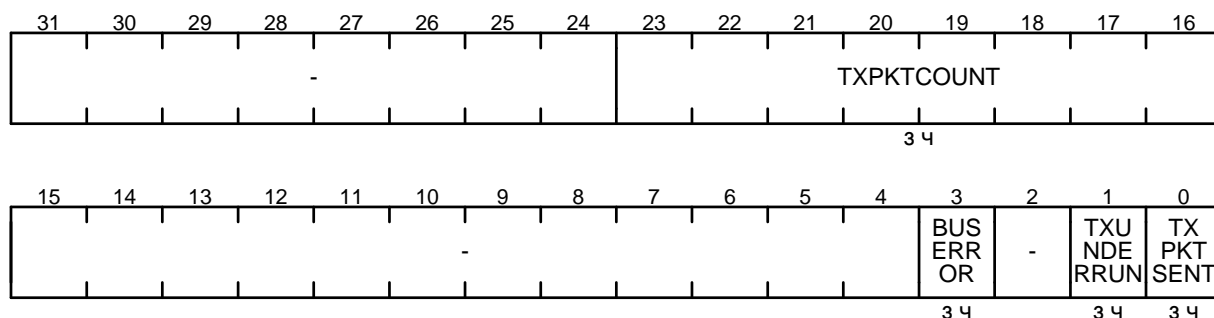


Поле	Биты	Описание
DADDR	31-2	Поле адреса регистра памяти, в котором расположены данные о первом передаваемом пакете. Встроенный ПДП контроллер читает этот регистр с целью получения адреса только в том случае, если процессором установлен бит TXENABLE в регистре DMATXCTRL
–	1-0	Зарезервировано

## DMATXSTAT – регистр статуса передачи

Смещение: + 188h

Сброс: 0h

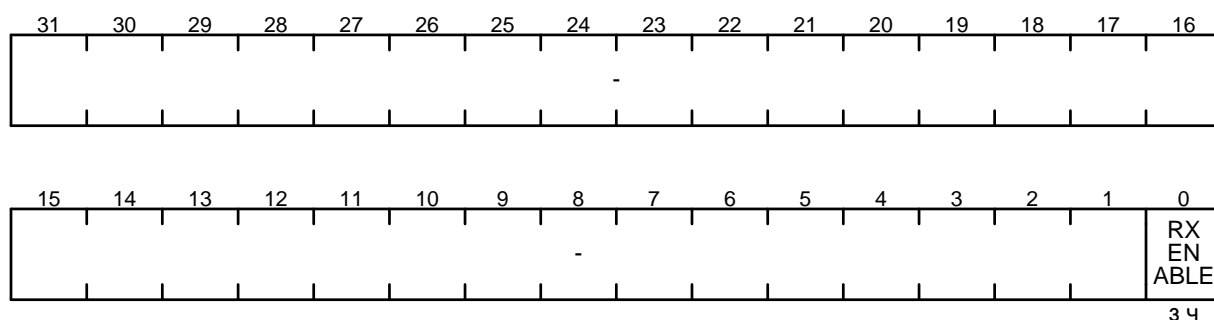


Поле	Биты	Описание
TXPKTCOUNT	23-16	8-битный счетчик пакетов передачи. Увеличивается на единицу каждый раз, когда встроенный ПДП контроллер успешно завершает передачу пакета, и уменьшается на единицу, если головной процессор записывает единицу в бит TXPKTSENT
BUSERROR	3	0   Нормальная работа
		1   Ошибка
TXUNDERRUN	1	0   Индикатор обработки данных
		1   ПДП контроллер читает и устанавливает флаг Empty в дескрипторе
TXPKTSENT	0	0   Флаг успешной передачи одного или более пакетов
		1   Бит сбрасывается всякий раз, когда счетчик TXPKTCOUNT обнуляется Установка бита приводит к уменьшению значения TXPKTCOUNT на единицу
–	31-24, 15-4, 2	Зарезервировано

## DMARXCTRL – регистр управления приемом

Смещение: + 18Ch

Сброс: 0h

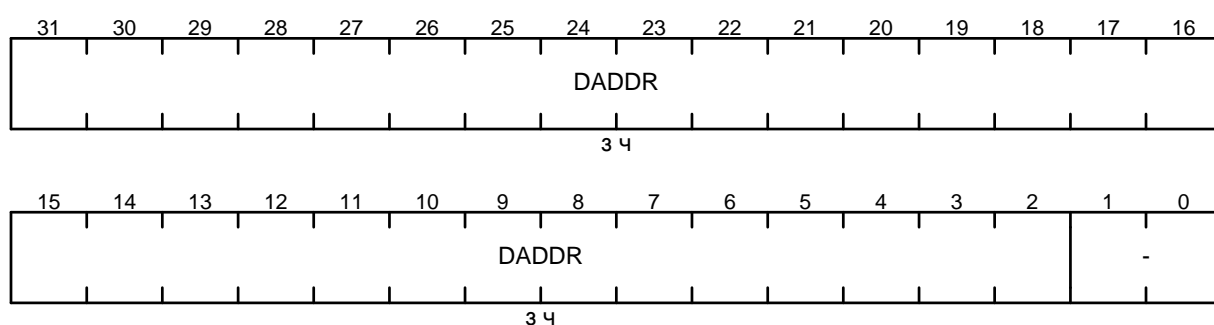


Поле	Биты	Описание	
RXENABLE	0	Бит разрешения устройству ПДП принимать пакеты. При установленном бите Бит очищается в случае обнаружения состояния RxOverflow или BusError	
		0	Запрещено
		1	Разрешено. Встроенный ПДП-контроллер готов начать принимать новый пакет, как только FIFO покажет, что новый пакет доступен (FRSOF подтвержден)
–	31-1	Зарезервировано	

## DMARXDESCR – регистр указателя дескриптора приема

Смещение: + 190h

Сброс: 0h

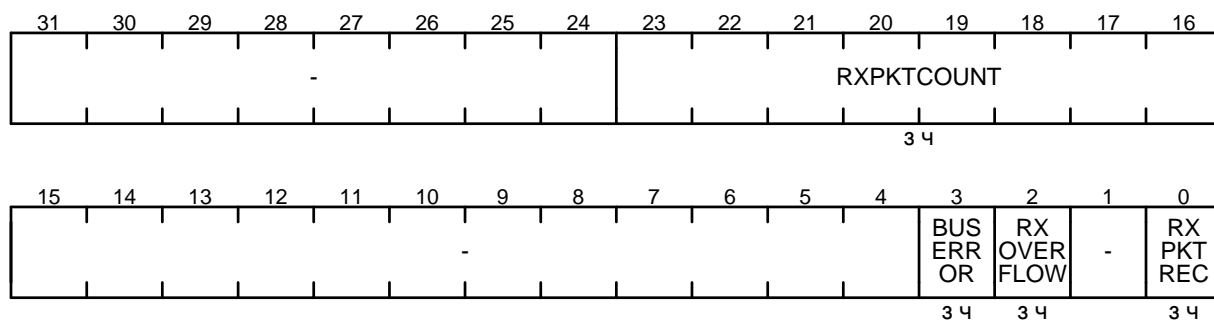


Поле	Биты	Описание
DADDR	31-2	Поле адреса регистра памяти, в который будут записаны данные о первом принятом пакете. Встроенный ПДП контроллер читает этот регистр с целью получения адреса только в том случае, если процессором установлен бит RXENABLE в регистре DMARXCTRL
–	1-0	Зарезервировано

## DMARXSTAT – регистр статуса приема

Смещение: + 194h

Сброс: 0h

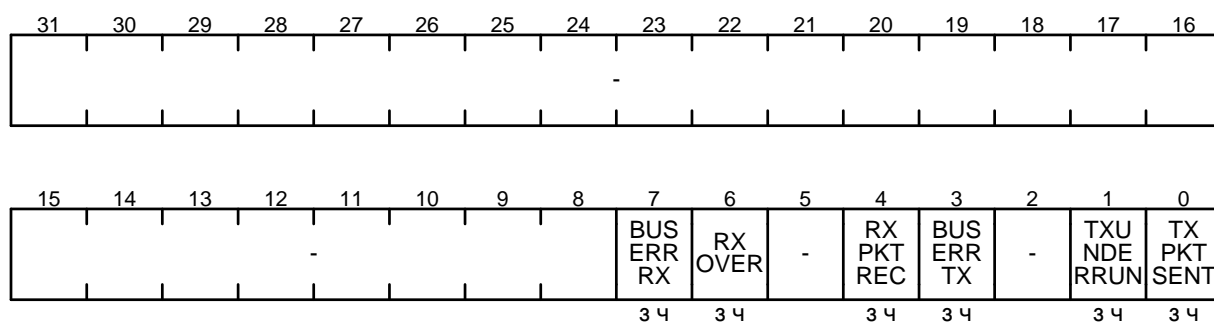


Поле	Биты	Описание
RXPKTCount	23-16	8-битный счетчик принятых пакетов. Увеличивается на единицу каждый раз, когда встроенный ПДП контроллер успешно завершает транзакцию пакета, и уменьшается на единицу, если головной процессор записывает единицу в бит RXPKTREC
BUSERROR	3	0   Нормальная работа
		1   Ошибка
RXOVERFLOW	2	0   Индикатор обработки данных
		1   ПДП контроллер прочитал «0» из флага Empty в дескрипторе приема во время обработки данных
RXPKTREC	0	0   Флаг успешного приема одного или более пакетов
		1   Бит сбрасывается всякий раз, когда счетчик RXPKTCount обнуляется
-	31-24, 15-4, 1	0   Установка бита приводит к уменьшению значения RXPKTCount на единицу
		1   Зарезервировано

## DMAINTMASK – регистр маски прерываний

Смещение: + 198h

Сброс: 0h



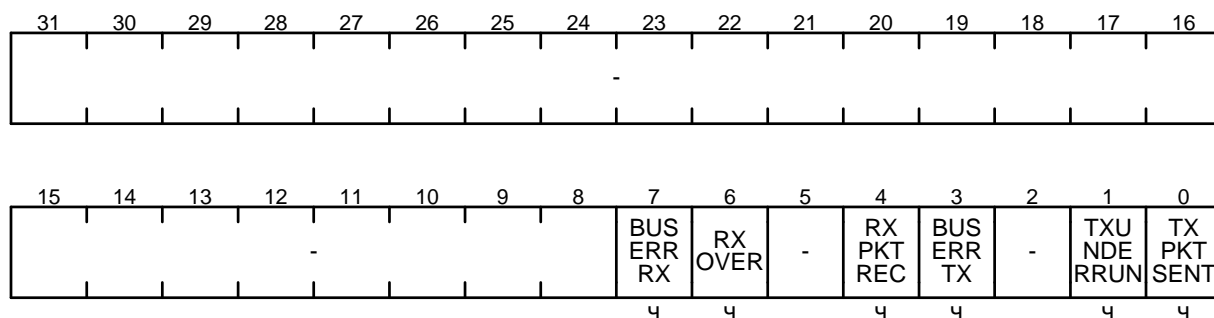
Поле	Биты	Описание
BUSERRRX	7	Бит разрешения прерывания при установке бита BUSERROR регистра DMARXSTAT
RXOVER	6	Бит разрешения прерывания при установке бита RXOVERFLOW регистра DMARXSTAT
RXPKTREC	4	Бит разрешения прерывания при установке бита RXPKTREC регистра DMARXSTAT
BUSERRTX	3	Бит разрешения прерывания при установке бита BUSERROR регистра DMATXSTAT
TXUNDERRUN	1	Бит разрешения прерывания при установке бита TXUNDERRUN регистра DMATXSTAT
TXPKTSENT	0	Бит разрешения прерывания при установке бита TXPKTSENT регистра DMATXSTAT
–	31-8, 5, 2	Зарезервировано

Примечание – Для разрешения прерывания следует установить соответствующий бит. Сброшенный бит запрещает установку соответствующего флага в регистре DMAINT (т. е. запрещает генерирование прерывания).

## DMAINT – регистр прерываний

Смещение: + 19Ch

Сброс: 0h



Поле	Биты	Описание
BUSERRRX	7	Флаг прерывания при установке бита BUSERROR регистра DMARXSTAT
RXOVER	6	Флаг прерывания при установке бита RXOVERFLOW регистра DMARXSTAT
RXPKTREC	4	Флаг прерывания при установке бита RXPKTREC регистра DMARXSTAT
BUSERRTX	3	Флаг прерывания при установке бита BUSERROR регистра DMATXSTAT
TXUNDERRUN	1	Флаг прерывания при установке бита TXUNDERRUN регистра DMATXSTAT
TXPKTSENT	0	Флаг прерывания при установке бита TXPKTSENT регистра DMATXSTAT
–	31-8, 5, 2	Зарезервировано

Примечание – Флаг прерывания устанавливается только в том случае, если оба бита – бит-источник прерывания и соответствующий бит регистра маски прерываний – установлены.



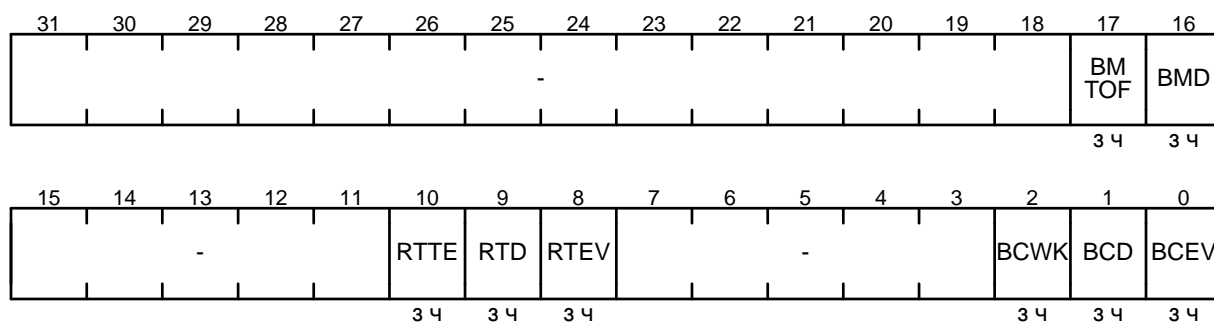
## A.24 Регистры контроллера МПИ (по ГОСТ Р 52070–2003)

Базовый адрес: 4007\_2000h      Регистры контроллера МПИ 0  
 4007\_3000h      Регистры контроллера МПИ 1

### IR – регистр прерываний МПИ

Смещение: + 00h

Сброс: 0h



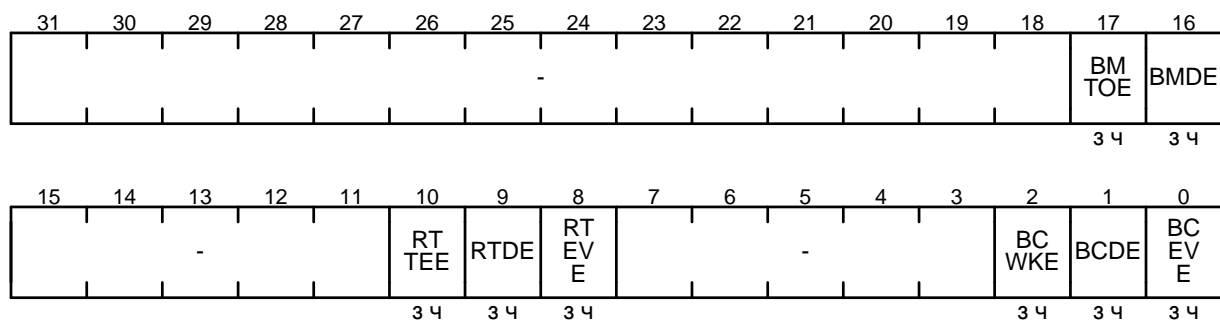
Поле	Биты	Описание
BM TOF	17	Переполнение таймера МПИ
BMD	16	Ошибка DMA МПИ
RTTE	10	Ошибка доступа к таблице УТ
RTD	9	Ошибка DMA УТ
RTEV	8	Прерывание по передаче УТ
BCWK	2	Прерывание по запуску таймера КИШ (не поддерживается)
BCD	1	Ошибка DMA КИШ
BCEV	0	Прерывание по передаче КИШ
–	31-18, 15-11, 7-3	Зарезервировано

Примечание – Биты регистра читаются как «1», если произошло прерывание, необходимо записывать «1» для сброса.

## IENR – регистр разрешения прерываний МПИ

Смещение: + 04h

Сброс: 0h

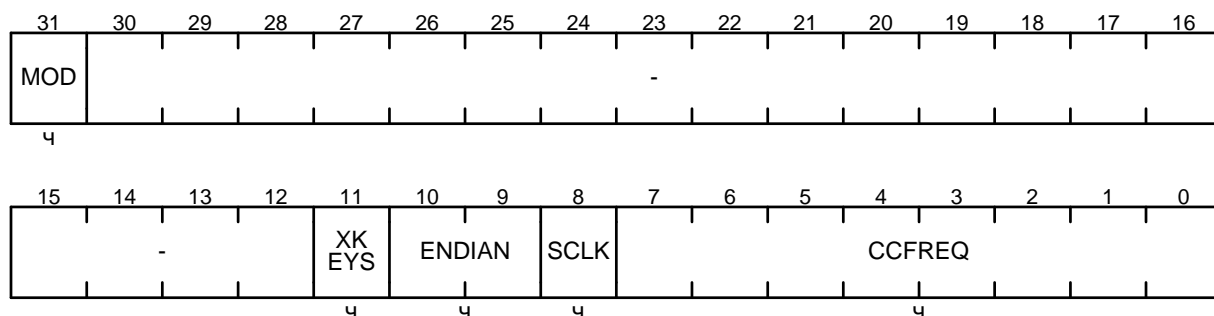


Поле	Биты	Описание
BMTOE	17	Разрешение прерывания по переполнению таймера МПИ
BMDE	16	Разрешение прерывания по ошибке DMA МПИ
RTTEE	10	Разрешение прерывания по ошибке доступа к таблице УТ
RTDE	9	Разрешение прерывания по ошибке DMA УТ
RTEVE	8	Разрешение прерывания по передаче УТ
BCWKE	2	Разрешение прерывания по запуску таймера КИ
BCDE	1	Разрешение прерывания по ошибке DMA КИ
BCEVE	0	Разрешение прерывания по передаче КИ
–	31-18, 15-11, 7-3	Зарезервировано

## HCON – регистр конфигурации МПИ

Смещение: + 10h

Сброс: 0h

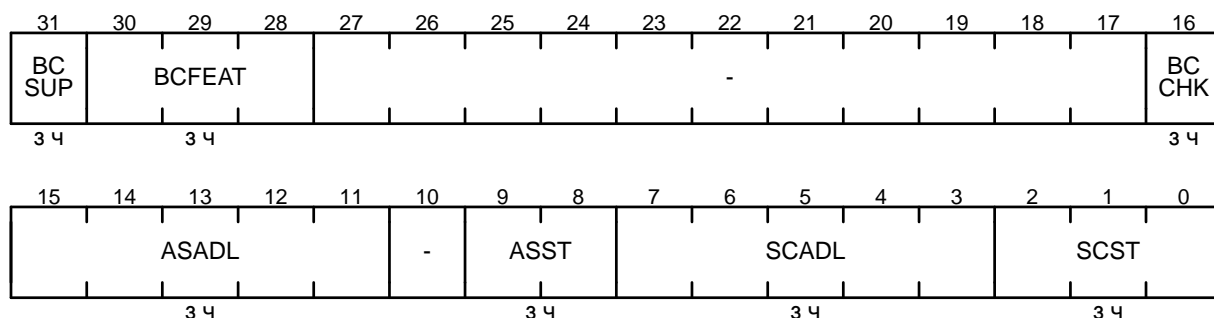


Поле	Биты	Описание
MOD	31	Модификация модуля. Бит зарезервирован
XKEYS	11	Бит установлен, если разрешено использование ключей в регистрах конфигурации МПИ и УТ ВМCON и RTCON
ENDIAN	10-9	Порядок передачи по АНВ
		00   Старший байт вперед
		01   Младший байт вперед
		10   Зарезервировано
	11   Зарезервировано	
SCLK	8	Частота модуля. Бит зарезервирован для будущих версий.
CCFREQ	7-0	Частота кодека. Зарезервировано для будущих версий модуля, указывает на изменение частоты кодека. Значение «0» показывает, что частота кодека составляет 20 МГц.
–	30-12	Зарезервировано

## BCSTCON – регистр статуса и конфигурации КШ

Смещение: + 40h

Сброс: F000\_0000h

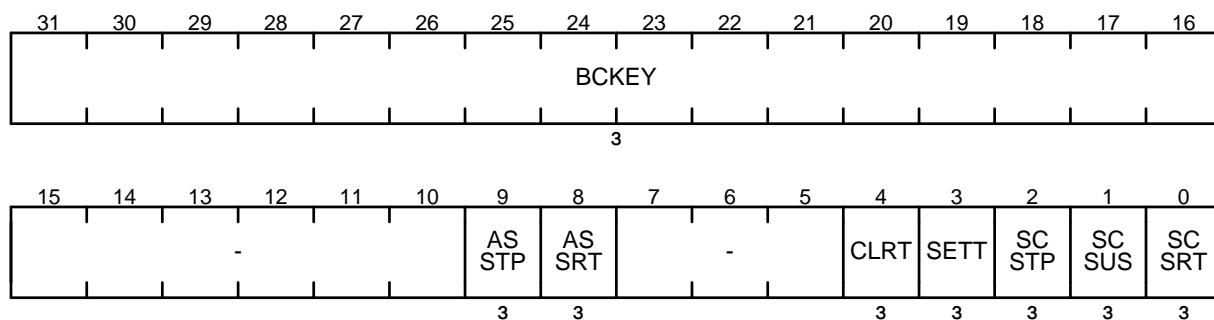


Поле	Биты	Описание	
BCSUP	31	Поддержка режима КШ – читается как «1», если режим поддерживается	
BCFEAT	30-28	Битовое поле показывает доступные опции КШ, «1» – если опция поддерживается	
		бит 30	Поддержка таймера последовательности передач КШ
		бит 29	Поддержка прерывания по запуску таймера последовательности передач КШ
		бит 28	Поддержка регистра КШ замены шины для УТ BCBSW и бита дескриптора STBUS в слове 1 дескриптора передачи
BCCHK	16	Проверка на групповые сообщения. Если бит установлен в «1», то разрешено ожидание и проверка всех групповых сообщений (неожидаемых)	
ASADL	15-11	Младшие биты адреса асинхронного списка передач. Биты 8-4 исполняемого в данный момент (если ASST = 01b) или адрес следующего дескриптора из асинхронного списка	
ASST	9-8	Состояние асинхронного списка передач	
		00	Последовательность передач остановлена
		01	Выполнение команды
		10	Ожидание таймслота
		11	Зарезервировано
SCADL	7-3	Младшие биты адреса основного списка передач. Биты 8-4 исполняемого в данный момент (если SCST = 001b) или адрес следующего дескриптора передачи	
SCST	2-0	Состояние основного списка передач	
		000	Последовательность передач остановлена
		001	Выполнение команды
		010	Ожидание таймслота
		011	Последовательность в состоянии ожидания
		100	Ожидание внешнего синхроимпульса
		101	Зарезервировано
		110	Зарезервировано
		111	Зарезервировано
–	21-17, 10	Зарезервировано	

## ВСАСТ – регистр действий КШ

Смещение: + 44h

Сброс: 0h

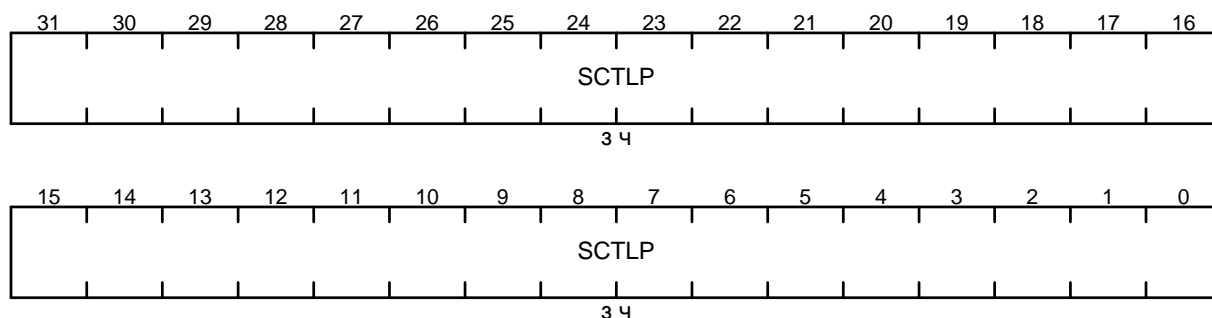


Поле	Биты	Описание
ВСКЕY	31-16	Ключ КШ. Необходимо записывать 1552h, иначе запись в регистр будет проигнорирована.
ASSTP	9	Запись «1» останавливает асинхронный список передач (после завершения текущей передачи).
ASSRT	8	Запись «1» запускает передачу из асинхронного списка.
CLRT	4	Запись «1» сбрасывает запомненный синхроимпульс
SETT	3	Запись «1» программно устанавливает синхроимпульс
SCSTP	2	Запись «1» останавливает последовательность передач (после завершения текущей передачи).
SCSUS	1	Запись «1» переводит последовательность передач в режим ожидания (после завершения текущей передачи).
SCSRT	0	Запись «1» запускает последовательность передач.
–	15-10, 7-5	Зарезервировано

### BCLNP – указатель следующей передачи списка передач КШ

Смещение: + 48h

Сброс: 0h

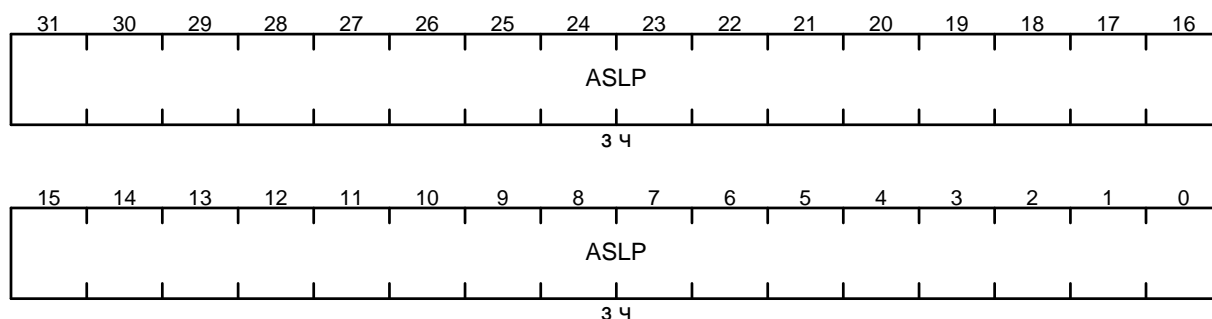


Поле	Биты	Описание
SCTLP	31-0	Запись: изменение адреса. Если последовательность передач запущена, то на указанный адрес будет выполнен переход после завершения текущей передачи Чтение: текущая передача (если SCST = 001b) или следующая передача из основного списка передач

### BCALNP – указатель следующей передачи асинхронного списка передач КШ

Смещение: + 4Ch

Сброс: 0h

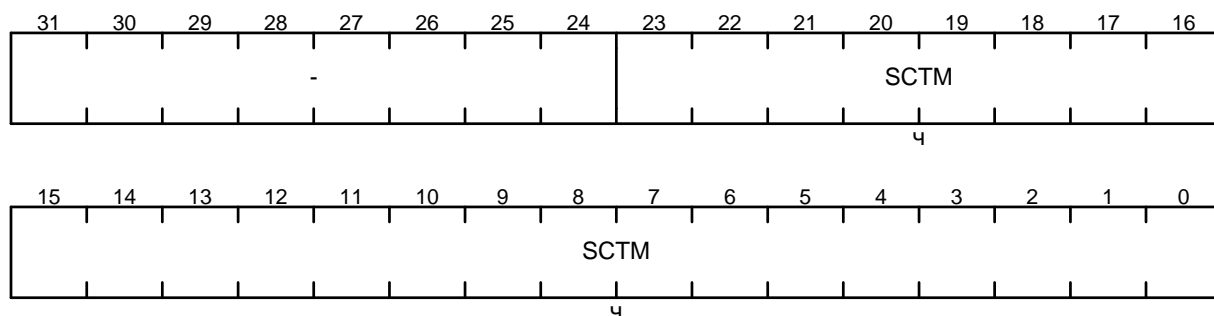


Поле	Биты	Описание
ASLP	31-0	Запись: изменение адреса. Если последовательность передач запущена, то на указанный адрес будет выполнен переход после завершения текущей передачи Чтение: текущая передача (если ASST = 01b) или следующая передача из асинхронного списка передач

## ВСТИМ – регистр таймера КШ

Смещение: + 50h

Сброс: 0h



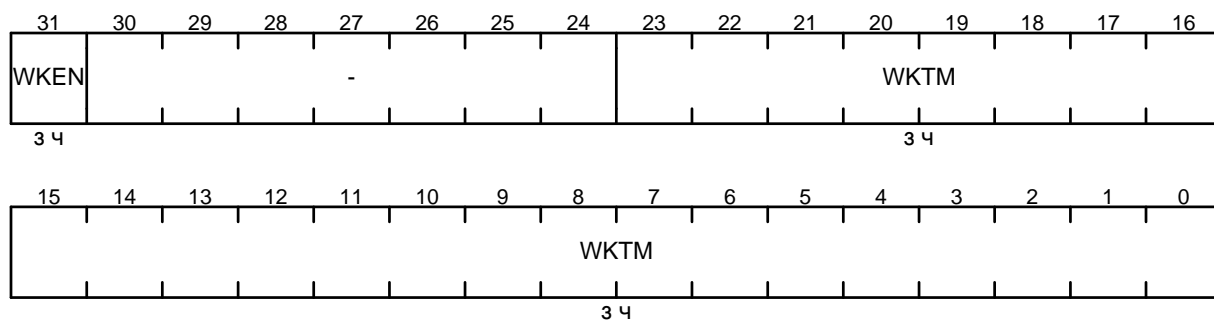
Поле	Биты	Описание
SCTM	23-0	Только для чтения. Время с начала выполнения передач (в микросекундах). Принимает нулевое значение при остановке последовательности передачи или при возникновении внешнего синхроимпульса
–	31-24	Зарезервировано

Примечание – Этот регистр является опциональным, см. BCSTCON.

## ВСТИМWK – Регистр запуска таймера

Смещение: + 54h

Сброс: 0h



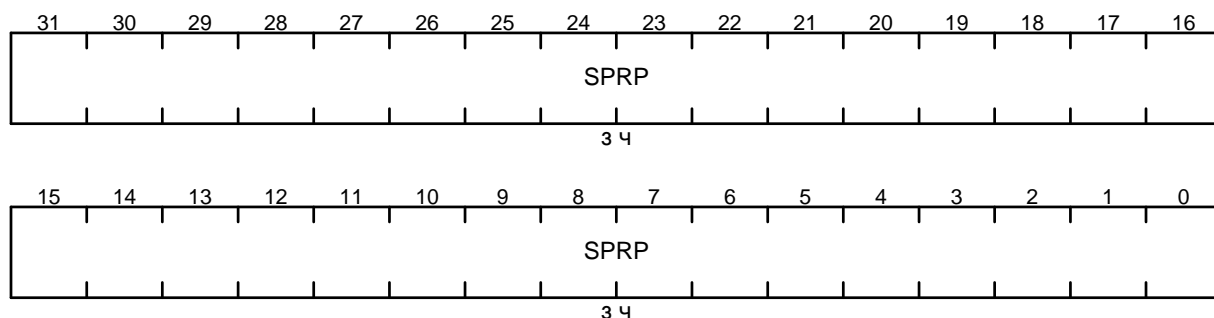
Поле	Биты	Описание
WKEN	31	Разрешение запуска таймера. Если установлен этот бит, то будет сгенерировано прерывание при равенстве значений WKTМ = SCTМ
WKTМ	23-0	Время запуска
–	30-24	Зарезервировано

Примечание – Этот регистр является опциональным, см. BCSTCON.

### BCTRP – регистр позиции прерывания по передаче КШ

Смещение: + 58h

Сброс: 0h

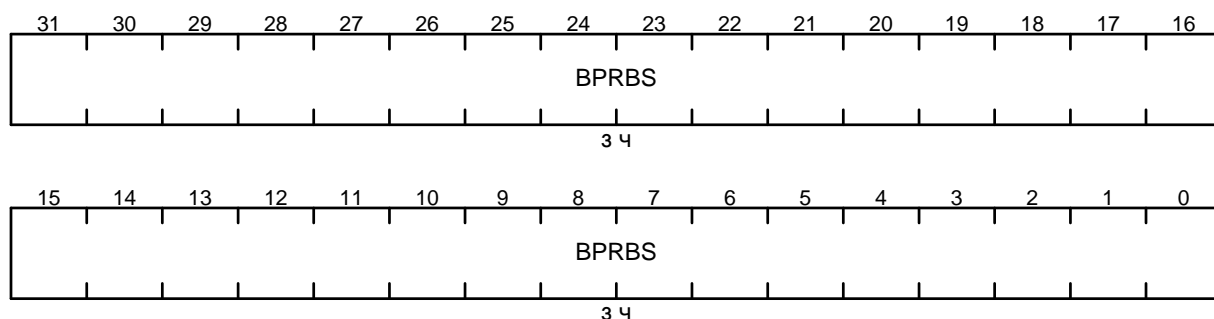


Поле	Биты	Описание
SPRP	31-0	Текущий указатель записи в кольцо прерываний. Биты 1-0 всегда равны «0» (выравнивание 4 байта). Выравнивание кольца прерываний составляет 64 байта, поэтому пользователь может менять только биты 31-6

### BCBSW – регистр КШ замены шины для УТ

Смещение: + 5Ch

Сброс: 0h



Поле	Биты	Описание
BPRBS	31-0	Между битом маски из данного регистра, соответствующего адресу выбранного УТ (УТ- приемнику в формате передачи от УТ к УТ), и значением бита выбора шины производится операция логического исключающего «ИЛИ». Этот регистр обновляется аппаратно, если был использован бит STBUS (бит 19 Слова 0 дескриптора передачи). Подробнее об использовании данного регистра – в подразделе 27.2 пункт «Выбор шины».

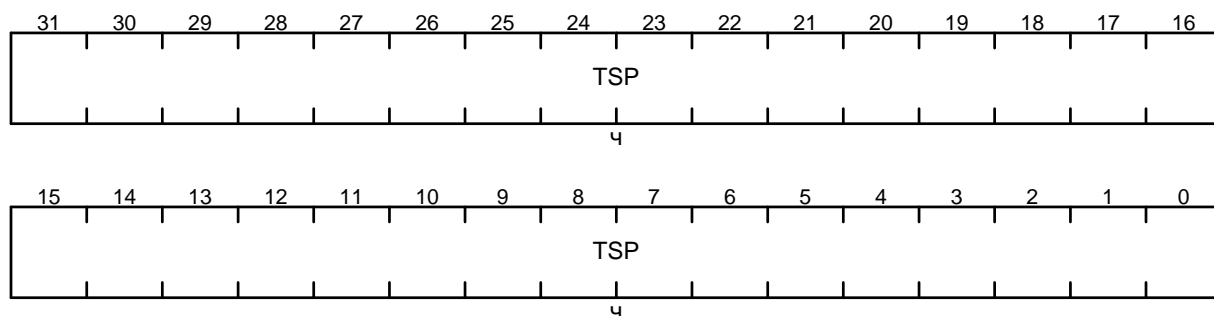
Примечание – Этот регистр является опциональным, см. бит 28 регистра BCSTCON.



### BCSTSP – указатель таймслота текущей передачи списка передач КШ

Смещение: + 68h

Сброс: 0h

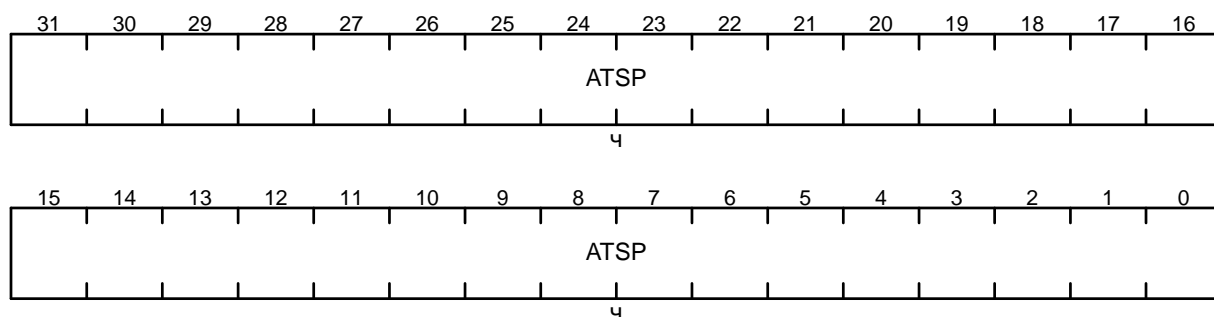


Поле	Биты	Описание
TSP	31-0	Только для чтения. Указывает на дескриптор передачи, соответствующий текущему таймслоту (содержит актуальное значение только во время выполнения списка передачи). Биты 3-0 всегда содержат нулевое значение (выравнивание 16 байт)

### BCATSP – указатель таймслота текущей передачи асинхронного списка передач КШ

Смещение: + 6Ch

Сброс: 0h

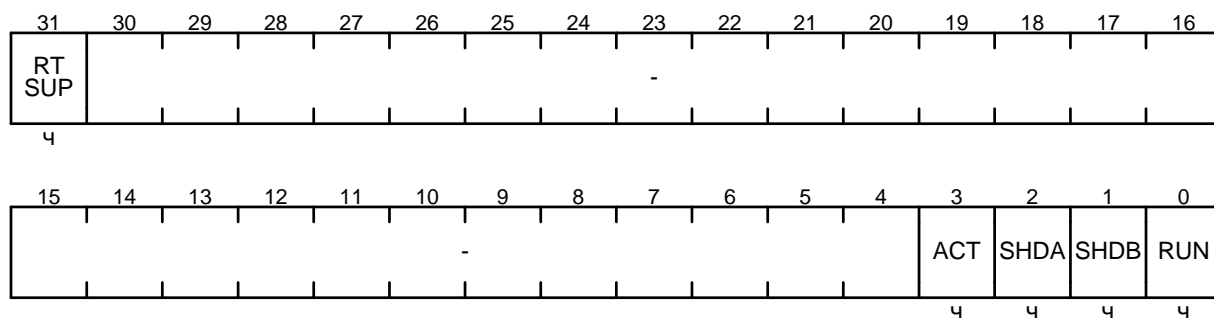


Поле	Биты	Описание
ATSP	31-0	Только для чтения. Указывает на дескриптор передачи, соответствующий текущему таймслоту (содержит актуальное значение только во время выполнения асинхронного списка передачи). Биты 3-0 всегда содержат нулевое значение (выравнивание 16 байт)

## RTSTAT – регистр статуса УТ

Смещение: + 80h

Сброс: 8000\_0000h

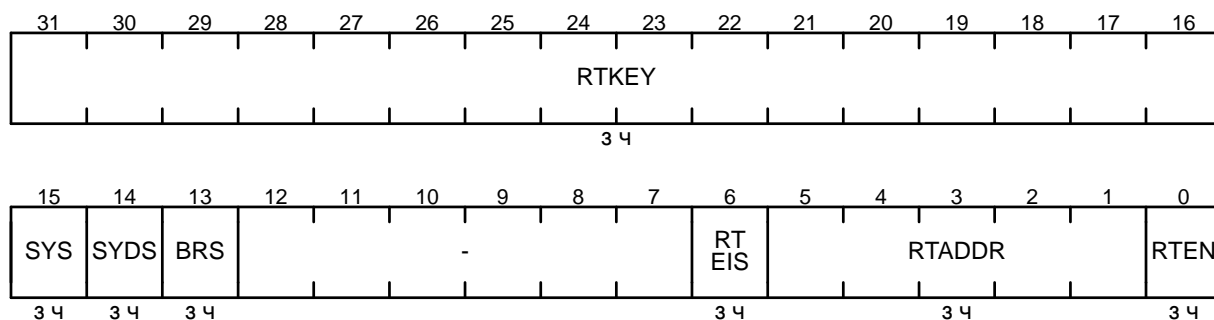


Поле	Биты	Описание
RTSUP	31	Читается как «1», если поддерживается режим работы в качестве УТ
ACT	3	Читается как «1», если УТ в данный момент выполняет передачу
SHDA	2	Читается как «1», если шина «А» была отключена контроллером шины (по шине «В» отправлена КУ «Блокировать передатчик»)
SHDB	1	Читается как «1», если шина «В» была отключена контроллером шины (по шине «А» отправлена КУ «Блокировать передатчик»)
RUN	0	Читается как «1», если УТ считывает команды
–	30-4	Зарезервировано

## RTCON – регистр конфигурации УТ

Смещение: + 84h

Сброс: 0000\_E03Eh

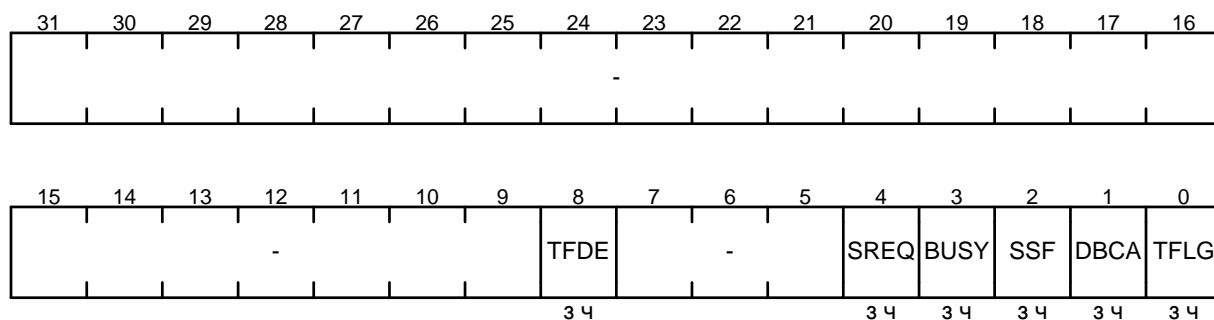


Поле	Биты	Описание
RTKEY	31-16	Ключ УТ. Необходимо записывать 1553h при смене адреса УТ, иначе поле RTADDR будет недоступно для записи. При чтении данное поле содержит нулевые значения
SYS	15	Разрешение сигнала синхронизации. Необходимо установить «1» для формирования сигнала синхронизации при приеме КУ «Синхронизация» (без слова данных)
SYDS	14	Разрешение сигнала синхронизации (со словом данных). Необходимо установить «1» для формирования сигнала синхронизации при приеме КУ «Синхронизация со словом данных»
BRS	13	Необходимо установить «1» для формирования сигнала сброса УТ при приеме КУ «Установить УТ в исходное состояние»
RTEIS	6	Читается как «1» если текущий адрес был задан с помощью внешних выводов. После установки адреса программно, бит сбрасывается
RTADDR	5-1	Адрес данного УТ (значения от 0 до 30)
RTEN	0	Разрешение работы УТ. Необходимо установить «1» для разрешения работы УТ
–	12-7	Зарезервировано

## RTBST – регистр статуса шины УТ

Смещение: + 88h

Сброс: 0h

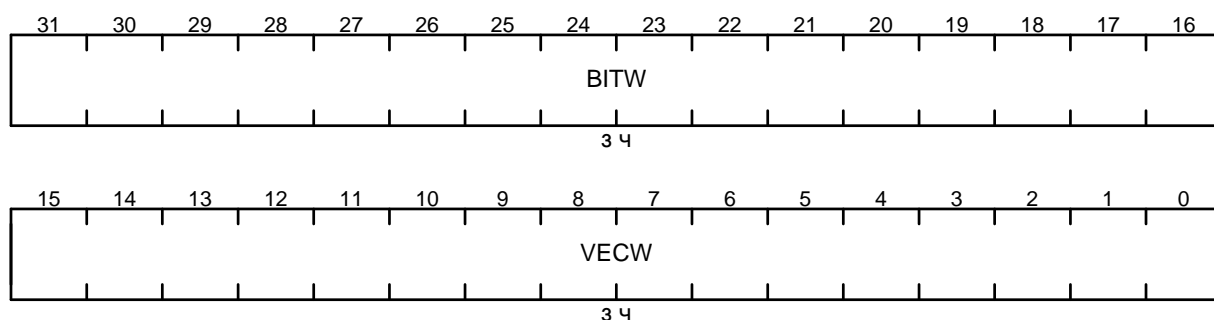


Поле	Биты	Описание
TFDE	8	Автоматическая установка флага TFLG при ошибках DMA и ошибках таблицы поадресов
SREQ	4	Запрос на обслуживание
BUSY	3	Абонент занят. Если установлен бит BUSY, то УТ посылает только ответное слово, передача считается неудачной
SSF	2	Флаг подсистемы. Используется для предоставления данных о состоянии подсистем, к которым подключен УТ. Данный бит сигнализирует о наличии неисправности или сбоя
DBCA	1	Бит подтверждения принятия управления шиной. Этот бит посылается только в ответ на прием КУ «Принять управление шиной»
TFLG	0	Флаг УТ, указывает на состояние неисправности УТ. КШ может замаскировать данный бит, используя КУ «Маскировать флаг УТ» (если разрешено использование данной КУ в регистре RTMOD)
–	31-9, 7-5	Зарезервировано

## RTSW – статусный регистр слов УТ

Смещение: + 8Ch

Сброс: 0h

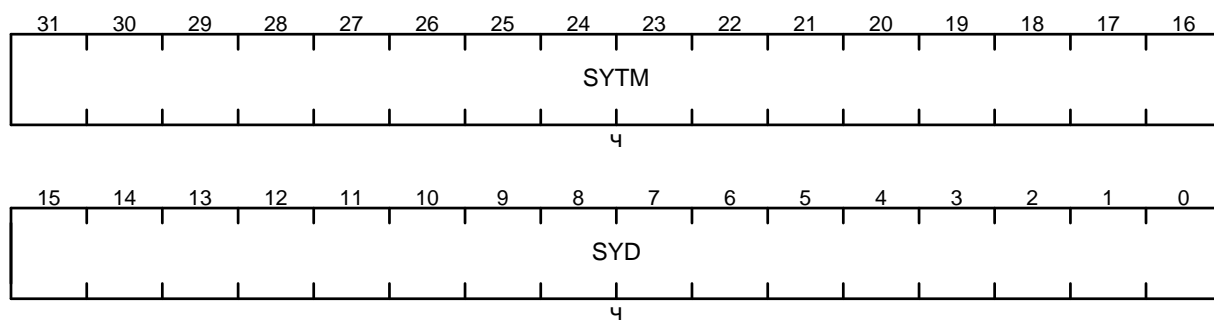


Поле	Биты	Описание
BITW	31-16	Слово ВІТ (Built-in-Test), передается в ответ на КУ «Передать слово ВІТ» (эта команда может использоваться после выполнения команды «Начать самоконтроль УТ»), если разрешено использование данной КУ в регистре RTMOD
VECW	15-0	Векторное слово, передается в ответ на КУ «Передать векторное слово», если разрешено использование данной КУ в регистре RTMOD

## RTSYNC – регистр синхронизации УТ

Смещение: + 90h

Сброс: 0h

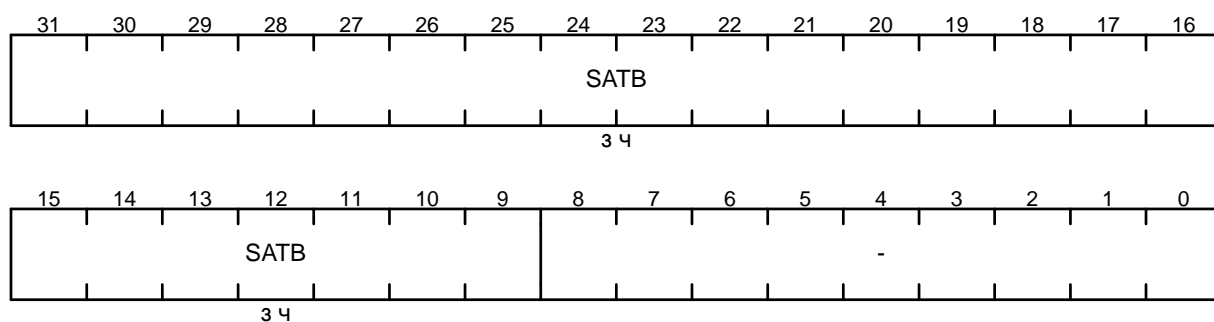


Поле	Биты	Описание
SYTM	31-16	Значение таймера УТ при выполнении последней КУ «Синхронизация» или «Синхронизация со словом данных», если разрешено использование данной КУ в регистре RTMOD
SYD	15-0	Слово данных, принятое при выполнении КУ «Синхронизация со словом данных», если разрешено использование данной КУ в регистре RTMOD

## RTSADDR – регистр основного адреса таблицы подадресов УТ

Смещение: + 94h

Сброс: 0h

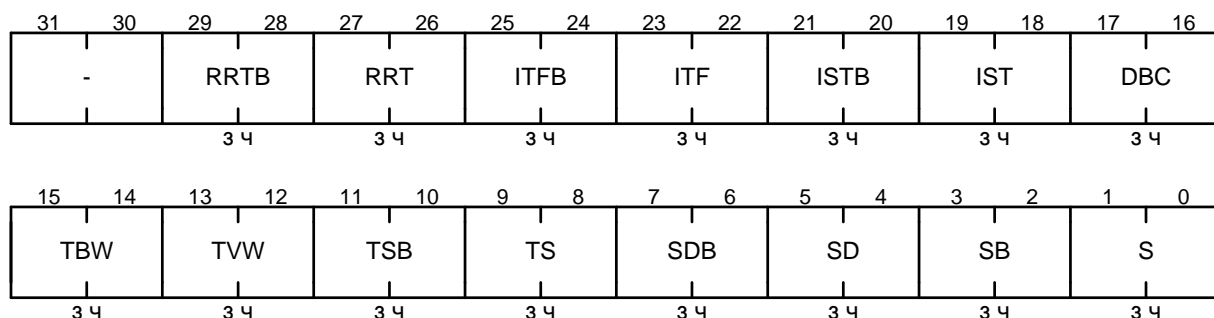


Поле	Биты	Описание
SATB	31-9	Начальный адрес таблицы подадресов, биты 31-9 адреса
–	12-7	Зарезервировано

## RTMOD – регистр УТ управления кодами команд

Смещение: + 98h

Сброс: 0000\_0555h



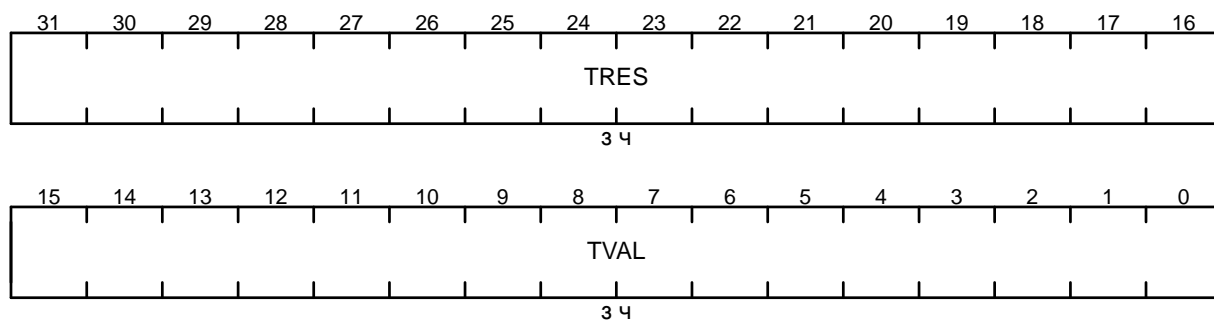
Поле	Биты	Описание
RRTB	29-28	Установить УТ в исходное состояние, групповая команда
RRT	27-26	Установить УТ в исходное состояние
ITFB	25-24	Маскировать флаг УТ/Отмена маскирования флага УТ, групповая команда
ITF	23-22	Маскировать флаг УТ/Отмена маскирования флага УТ
ISTB	21-20	Начать самоконтроль УТ, групповая команда
IST	19-18	Начать самоконтроль УТ
DBC	17-16	Принять управление шиной
TBW	15-14	Передать слово ВIT
TVW	13-12	Передать векторное слово
TSB	11-10	Блокировать/Разблокировать выбранный передатчик, групповая команда
TS	9-8	Блокировать/Разблокировать выбранный передатчик
SDB	7-6	Синхронизация со словом данных, групповая команда
SD	5-4	Синхронизация со словом данных
SB	3-2	Синхронизация, групповая команда
S	1-0	Синхронизация
–	31-30	Зарезервировано

Примечание – Для управления выбранной командой необходимо записать в соответствующее поле: «00» – для запрещения команды;  
 «01» – для разрешения команды;  
 «10» – для разрешения команды и разрешения логирования;  
 «11» – для разрешения команды, разрешения логирования и прерывания.

### RTTIM – регистр таймера УТ

Смещение: + A4h

Сброс: 0h

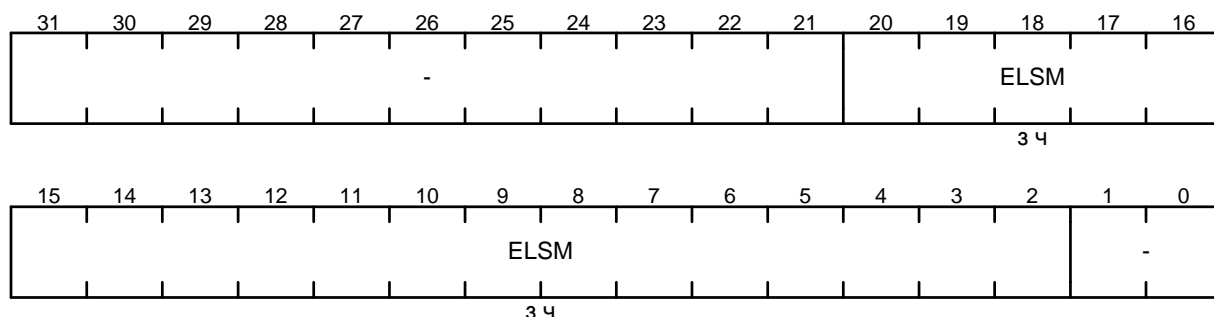


Поле	Биты	Описание
TRES	31-16	Период таймера УТ в микросекундах минус 1
TVAL	15-0	Текущее значение таймера

### RTTELSM – регистр масок лога события УТ

Смещение: + ACh

Сброс: FFFF\_FFFCh



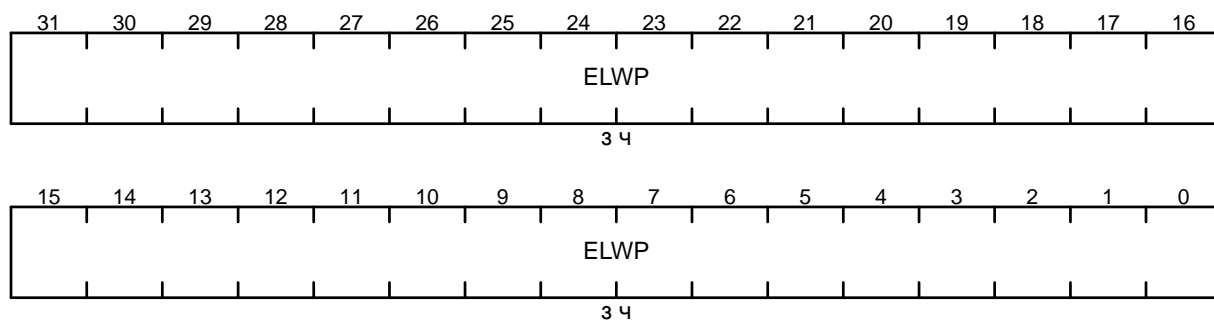
Поле	Биты	Описание
–	31-21	Биты необходимо устанавливать в «1»
ELSM	20-3	Маска, определяющая размер и выравнивание для кольцевого буфера лога событий УТ. Для выравнивания биты 31-21 должны содержать «1», биты 2-0 – «0»
–	2-0	Биты необходимо устанавливать в «0»



### RTELP – регистр позиции лога событий УТ

Смещение: + B0h

Сброс: 0h

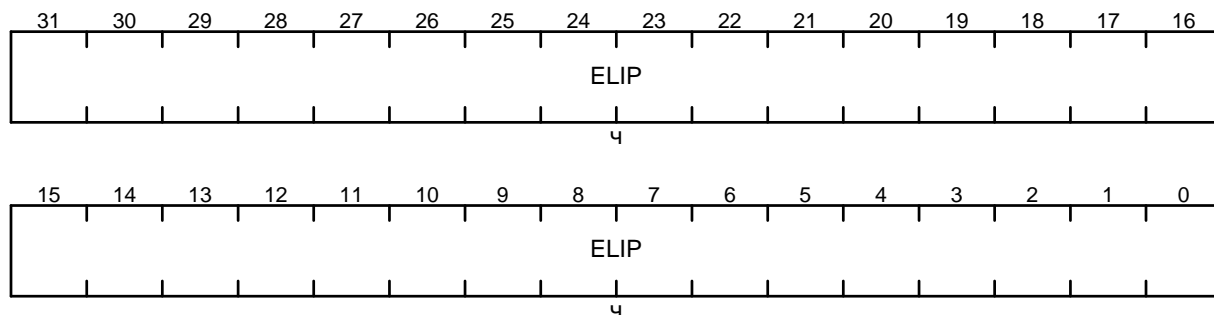


Поле	Биты	Описание
ELWP	31-0	Адрес первой неиспользуемой (наиболее ранней) записи в буфер лога событий, выравнивание 32 бита

### RTELP – регистр позиции прерываний в логе событий УТ

Смещение: + B4h

Сброс: 0h

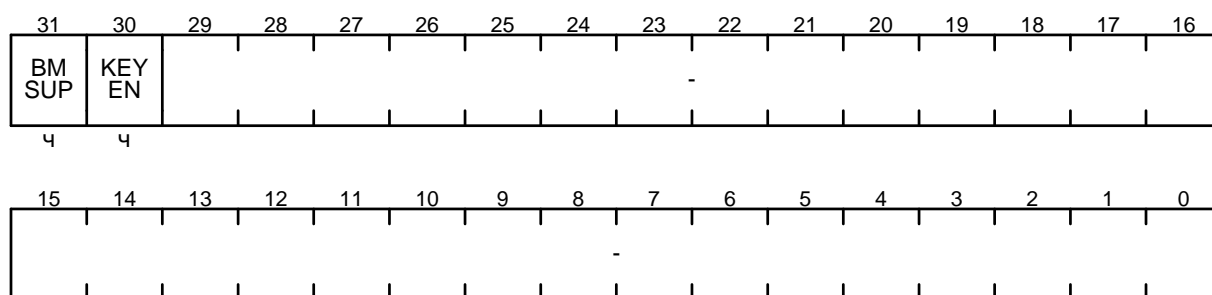


Поле	Биты	Описание
ELIP	31-0	Адрес для записи буфера лога событий, вызывающей прерывание, выравнивание 32 бита. Этот регистр записывается для первого прерывания, для следующих прерываний он записывается только после обработки первого

## BMSTAT – регистр статуса МШ

Смещение: + C0h

Сброс: 8000\_0000h

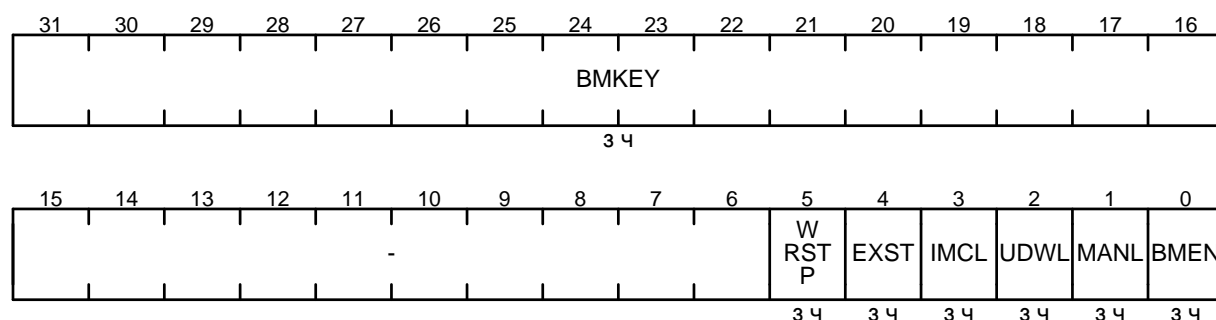


Поле	Биты	Описание
BMSUP	31	Читается как «1», если поддерживается режим работы МШ
KEYEN	30	Читается как «1», если МШ учитывает ключ ВМKEY при записи регистра конфигурации
-	29-0	Зарезервировано

## VMCON – регистр конфигурации МШ

Смещение: + C4h

Сброс: 0h

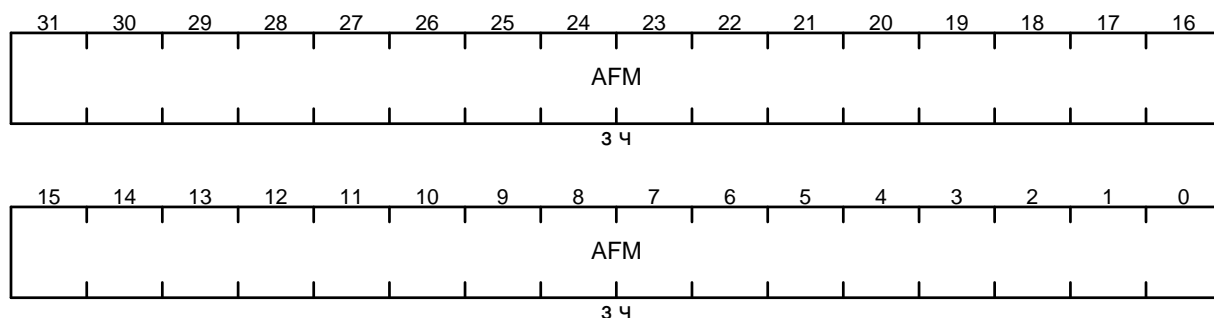


Поле	Биты	Описание
VMKEY	31-16	Ключ МШ. Если разрешены дополнительные ключи безопасности (поле KEYEN регистра VMSTAT), то в это поле необходимо записывать 1543h для разрешения записи, иначе запись будет игнорироваться. При чтении содержит нулевые значения
WRSTP	5	Если установлен в «1», то бит VMEN будет сброшен и МШ остановится после того, как позиция в буфере логирования вернется от конца буфера к его началу.
EXST	4	Если установлен в «1», то бит VMEN так же будет установлен в «1» и МШ начнет работу после приема внешнего синхроимпульса от КШ
IMCL	3	Необходимо установить в «1» для логирования недопустимых сообщений или зарезервированных КУ (см. таблицу 27.8 – Коды команд управления УТ)
UDWL	2	Необходимо установить в «1» для логирования слов данных, не относящихся к какой-либо команде
MANL	1	Необходимо установить в «1» для логирования ошибок декодирования (ошибка кода Манчестера, ошибка четности)
VMEN	0	Необходимо установить в «1» для разрешения логирования МШ
–	15-6	Зарезервировано

### BMADF – регистр фильтрации МШ по адресу УТ

Смещение: + C8h

Сброс: FFFF\_FFFFh

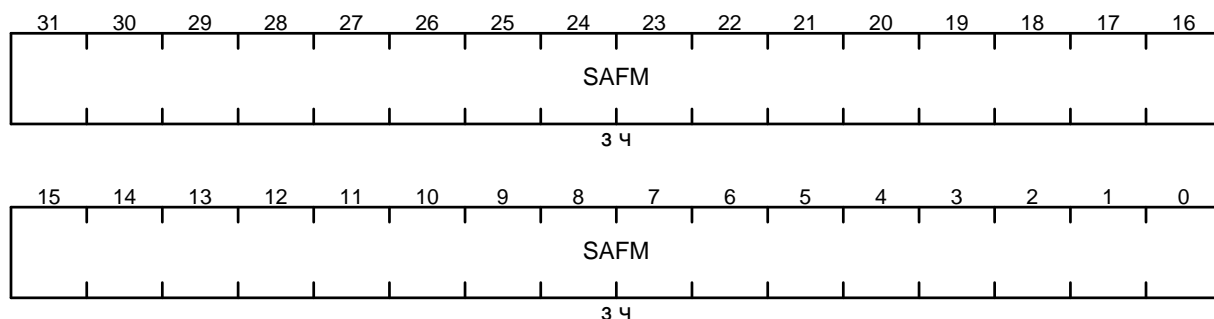


Поле	Биты	Описание
AFM	31-0	Для разрешения логирования событий для определенного адреса УТ, необходимо установить в «1» бит, соответствующий адресу. Бит 31 разрешает логирование групповых сообщений

### BMSADF – регистр фильтрации МШ по подадресу УТ

Смещение: + CCh

Сброс: FFFF\_FFFFh

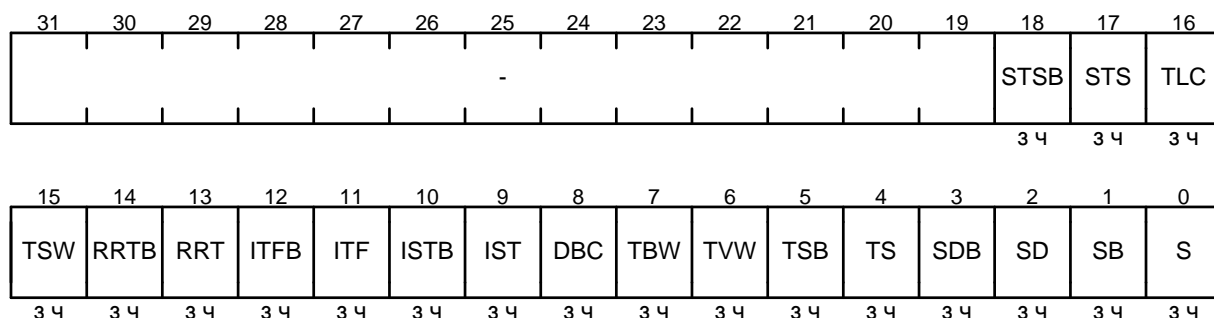


Поле	Биты	Описание
SAFM	31-0	Для разрешения логирования событий для определенного подадреса УТ, необходимо установить в «1» бит, соответствующий подадресу. Бит 31 разрешает логирование КУ с подадресом 31, бит 0 разрешает логирование КУ с подадресом 0

## BMMODF – регистр фильтрации МШ по коду управляющей команды

Смещение: + D0h

Сброс: FFFF\_FFFFh



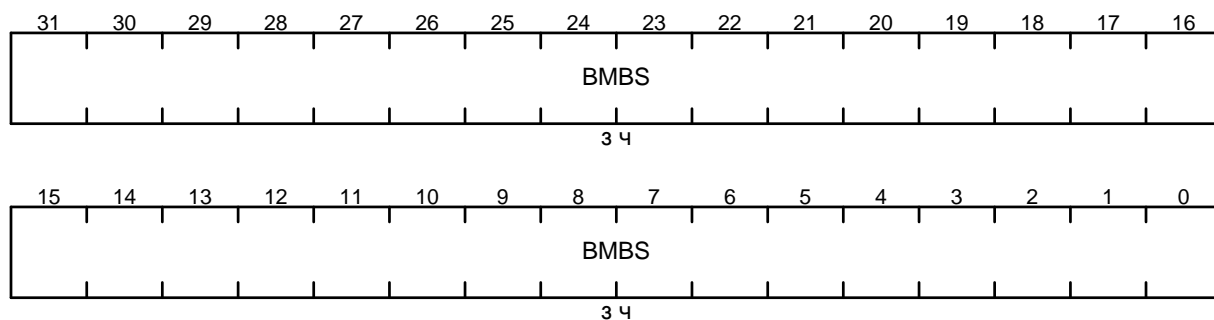
Поле	Биты	Описание
STSB	18	Блокировать/Разблокировать выбранный передатчик, групповая команда
STS	17	Блокировать/Разблокировать выбранный передатчик
TLC	16	Передать последнюю команду
TSW	15	Передать ответное слово
RRTB	14	Установить УТ в исходное состояние, групповая команда
RRT	13	Установить УТ в исходное состояние
ITFB	12	Маскировать флаг УТ/Отмена маскирования флага УТ, групповая команда
ITF	11	Маскировать флаг УТ/Отмена маскирования флага УТ
ISTB	10	Начать самоконтроль УТ, групповая команда
IST	9	Начать самоконтроль УТ
DBC	8	Принять управление шиной
TBW	7	Передать слово ВIT
TVW	6	Передать векторное слово
TSB	5	Блокировать/Разблокировать передатчик, групповая команда
TS	4	Блокировать/Разблокировать передатчик
SDB	3	Синхронизация со словом данных, групповая команда
SD	2	Синхронизация со словом данных
SB	1	Синхронизация, групповая команда
S	0	Синхронизация
–	31-19	Зарезервировано

Примечание – Для разрешения логирования определенной КУ необходимо установить в «1» соответствующий ей бит.

### **BMLBS – начальный адрес лога**

Смещение: + D4

Сброс: 0h

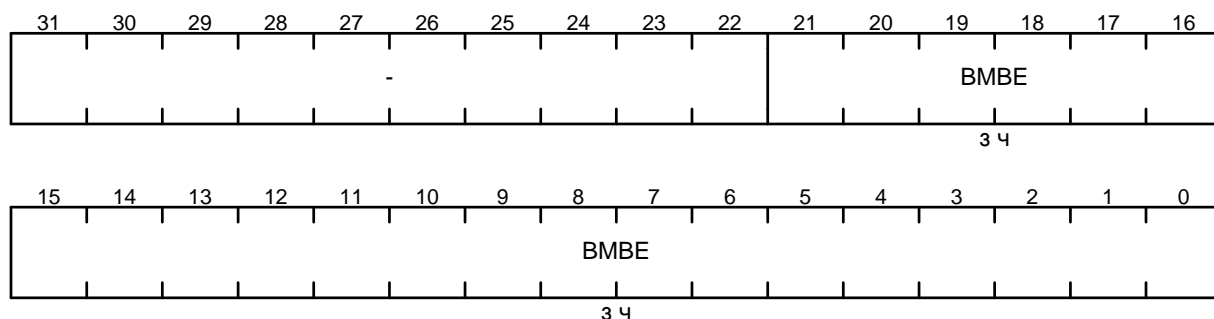


Поле	Биты	Описание
BMBS	31-0	Указатель на начальный адрес (младший) в буфере логирования МШ, выравнивание 8 байт (для обеспечения выравнивания, биты 2-0 всегда содержат нулевые значения)

### **BMLBE – конечный адрес лога**

Смещение: + D8

Сброс: 0000\_0007h

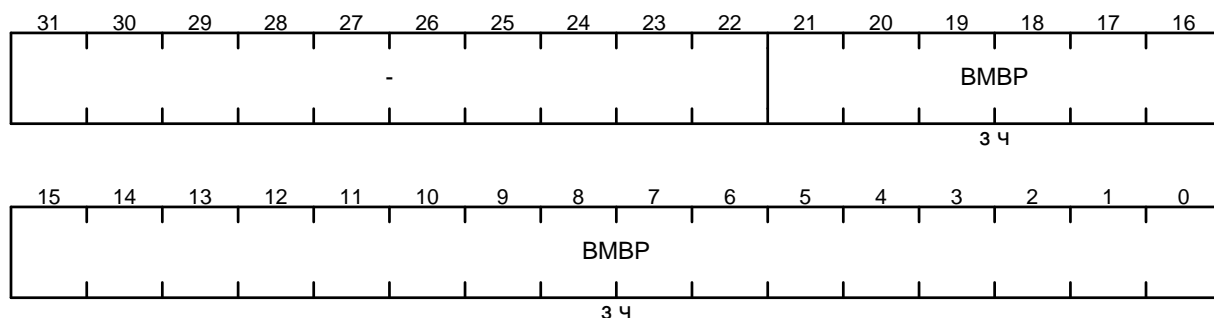


Поле	Биты	Описание
BMBE	21-0	Указатель на конечный адрес (старший) в буфере логирования МШ. Записываемыми являются только биты 21-3, то есть буфер ограничен 4 МБ. Для обеспечения выравнивания биты 2-0 всегда равны 1
–	31-22	Зарезервировано. При чтении содержит такое же значение, как и в регистре начального адреса

## ВМЛВР – указатель буфера лог МШ

Смещение: + DC

Сброс: 0h

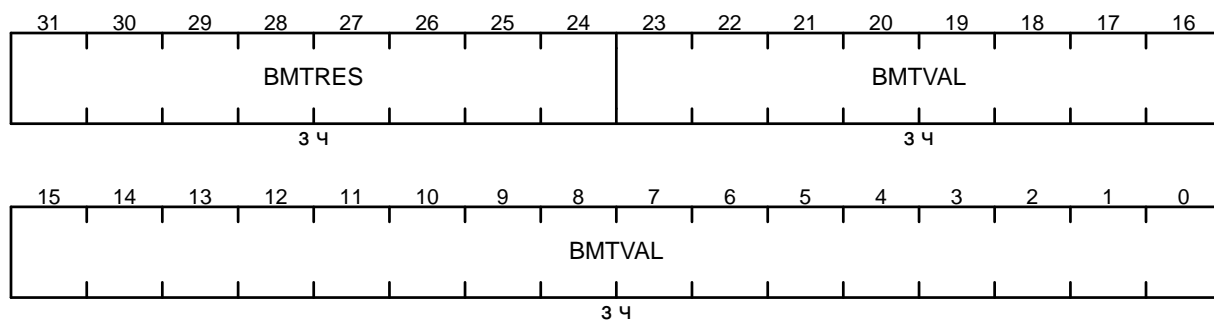


Поле	Биты	Описание
ВМВР	21-0	Указатель на позицию, которая будет записана следующей в лог МШ. Записываемыми являются только биты 21-3, то есть буфер ограничен 4 МБ. Для обеспечения выравнивания биты 2-0 всегда содержат нулевые значения
–	31-22	Зарезервировано. При чтении содержит такое же значение, как и в регистре начального адреса

## ВМТІМ – регистр таймера МШ

Смещение: + E0

Сброс: 0h



Поле	Биты	Описание
ВМТRES	31-24	Период таймера МШ в микросекундах минус 1
ВМТVAL	23-0	Текущее значение таймера

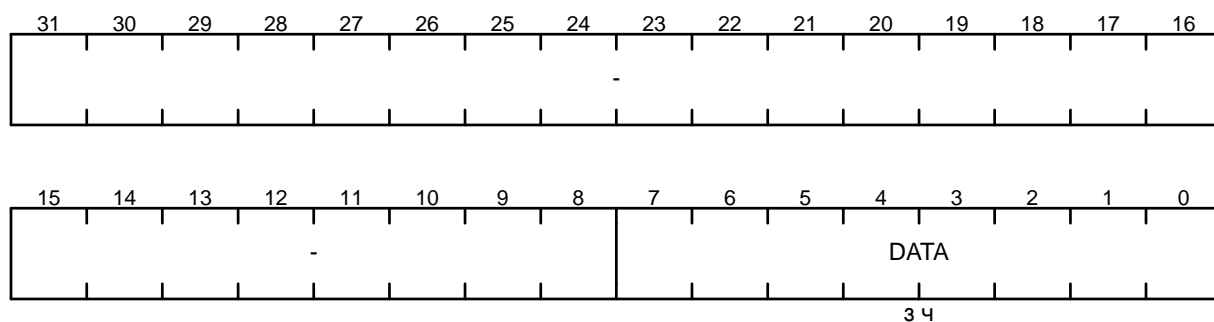
## A.25 Регистры контроллера интерфейса I2C

**Базовый адрес:** 4008\_B000h      Регистры блока I2C0  
 4008\_C000h      Регистры блока I2C1

### SDA – регистр данных

Смещение: + 00h

Сброс: 0xxh



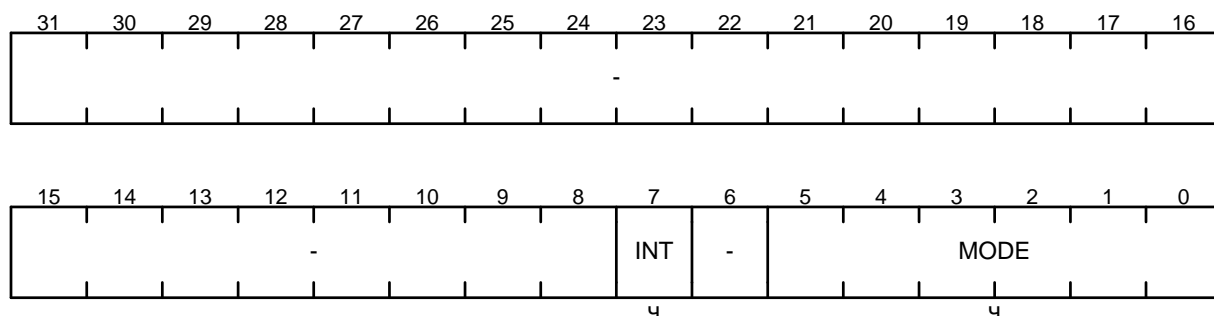
Поле	Биты	Описание
DATA	7-0	Поле данных
-	31-8	Зарезервировано



## ST – регистр состояния

Смещение: + 04h

Сброс: 0h

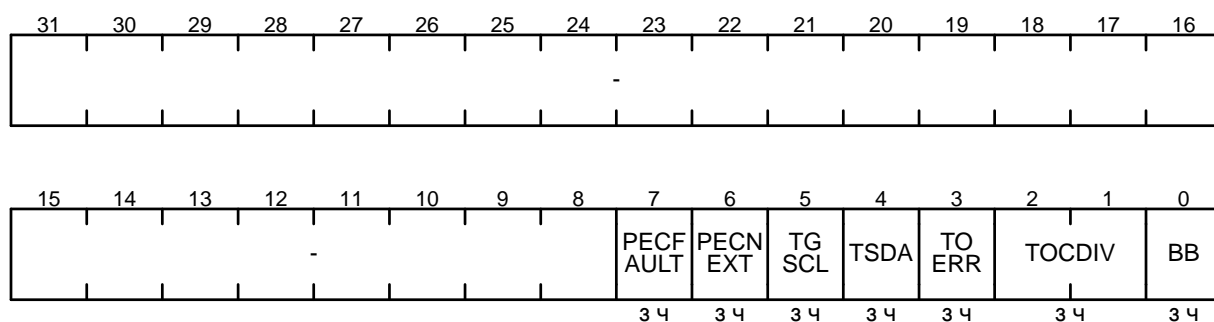


Поле	Биты	Описание
INT	7	<p>Флаг прерывания.</p> <p>Устанавливается после девятого такта сигнала SCL (когда SCL = 0) в любое запрограммированное время. Условия выставления флага INT:</p> <ul style="list-style-type: none"> <li>- во время приема/передачи, как в режиме мастера, так и в режиме ведомого;</li> <li>- при совпадении адреса (адреса ведомого, адреса отклика или адреса общего вызова) содержимое регистра SDA должно контролироваться программно для определения типа полученного адреса;</li> <li>- после успешного формирования стартового состояния или состояния повторного старта;</li> <li>- в случае неквитирования переданной информации;</li> <li>- при потере арбитража во время передачи последнего бита;</li> <li>- при обнаружении валидного состояния останова или состояния повторного старта;</li> <li>- при обнаружении ошибки на шине.</li> </ul> <p>Пока установлен флаг INT, на линии SCL удерживается низкий уровень сигнала.</p> <p>Флаг INT может быть сброшен установкой бита CLRST в регистре CTL0 или выключением модуля I2C (обнуление бита ENABLE в регистре CTL1).</p> <p>Условия выставления флага INT (не влияющие на уровень сигнала на линии SCL):</p> <ul style="list-style-type: none"> <li>- простой на линии SCL;</li> <li>- состояние останова в режиме ведомого (MODE = 1Ch);</li> <li>- потеря арбитража, вследствие чего ведомый переключился в безадресный режим (MODE = 03h или MODE = 23h);</li> <li>- неквитированная передача байта данных (MODE = 17h)</li> </ul>
MODE	5-0	<p>Код состояния.</p> <p>Возникновение того или иного состояния в течение функционирования модуля I2C сопровождается записью соответствующего кода в поле MODE</p>
–	31-8, 6	Зарезервировано

## CST – регистр управления и статуса

Смещение: + 08h

Сброс: 0h



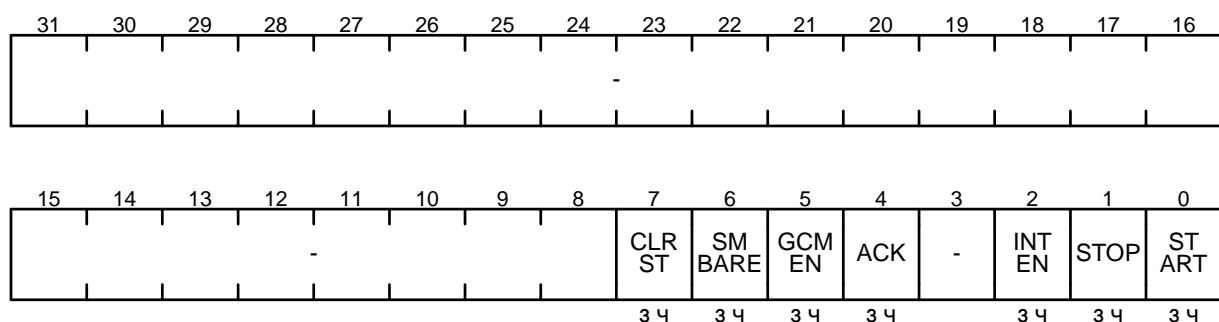
Поле	Биты	Описание
PECFAULT	7	Флаг ошибки. Устанавливается в случае, если после расчета контрольной суммы для пакета данных и сравнения ее с полученной суммой, значение во внутреннем регистре ошибок не нулевое
PECNEXT	6	Бит управления отправкой байта контрольной суммы. Установка бита указывает на то, что следующий передаваемый байт будет байтом CRC (байт контрольной суммы). Реакция на установку бита PECNEXT зависит от режима работы. В режиме мастера передатчика установка бита PECNEXT вызовет загрузку результата вычисления CRC в регистр SDA. После сброса флага INT начнется передача байта CRC. В режиме приемника установка этого бита будет указывать логике управления на то, что следующий байт, который будет принят, будет байтом CRC. В режиме ведомого приемника модуль I2C автоматически будет квитировать или не квитировать прием байта CRC, в зависимости от того, будет ли выявлена ошибка пакета данных или нет. В режиме мастера приемника по окончании приема байта CRC, будет отправлено значение бита ACK регистра CTL0
TG SCL	5	Бит переключения SCL. Бит позволяет переключать вывод I2C_SCL во время восстановления после ошибки. Когда на выводе I2C_SDA – низкий уровень сигнала, запись «1» в бит TG SCL переключит вывод SCL на один такт. Когда на выводе I2C_SDA высокий уровень сигнала, запись «1» в бит TG SCL игнорируется. Бит очищается аппаратно по окончании такта
TSDA	4	Бит тестирования SDA. Содержит текущее значение SDA. Этот бит можно использовать для отслеживания окончания процесса восстановления после ошибки, в течение которого ведомый постоянно поддерживает низкий уровень сигнала на выводе SDA
TOERR	3	Флаг ошибки простоя на шине. Если TOERR = 1b, это указывает на то, что на линии SCL был обнаружен простой. Флаг TOERR выставляется по обнулению основного счетчика времени простоя и может быть сброшен записью «1» в бит CLRST регистра CTL0

Поле	Биты	Описание
TOCDIV	2-1	Поле коэффициента делителя. Устанавливает коэффициент деления системного тактового сигнала, подаваемого на предделитель времени простоя линии SCL
		00 Тактовый сигнал отсутствует
		01 Деление на 4
		10 Деление на 8
		11 Деление на 16
BB	0	Флаг занятости шины. Если BB = 1b, это указывает на то, что шина занята. Устанавливается, как только шина переходит в активное состояние (одновременное появление низкого уровня сигнала на выводах I2C_SDA и I2C_SCL или хотя бы на одном из них) или в стартовое состояние. Сбрасывается при выключении интерфейса I2C либо при обнаружении состояния останова
–	31-8	Зарезервировано

### CTL0 – регистр управления 0

Смещение: + 0Ch

Сброс: 0h



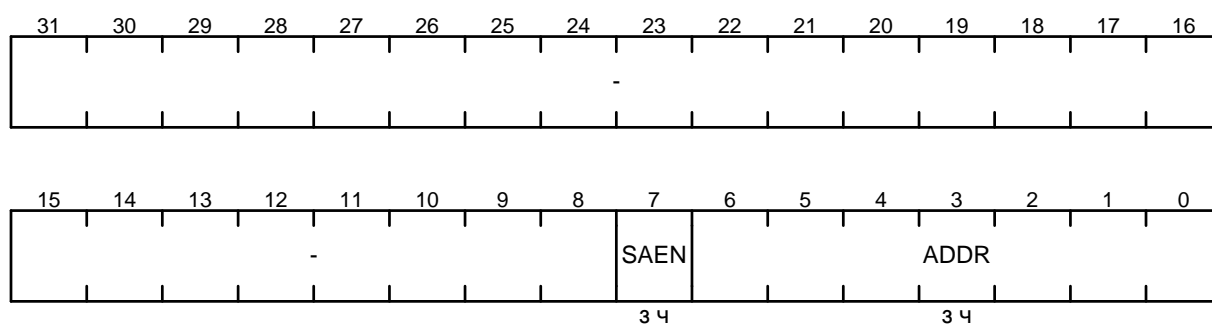
Поле	Биты	Описание	
CLRST	7	Бит сброса флага прерывания INT	
		Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает флаг INT в регистре ST
SMBARE	6	Бит управления реакцией на получение адреса отклика	
		0	Полученный адрес не проверяется на совпадение с адресом отклика
		1	Адрес, полученный сразу после старта, проверяется на совпадение с адресом отклика (0001100b)
		Бит очищается при выходе ведомого из режима IDLE	
GCMEN	5	Бит управления реакцией на получение адреса общего вызова	
		0	Полученный адрес не проверяется на совпадение с адресом общего вызова
		1	Адрес, полученный сразу после старта, проверяется на совпадение с адресом общего вызова (0000000b)
		Бит очищается при выходе ведомого из режима IDLE	

Поле	Биты	Описание
ACK	4	Бит квитирования приема. В режиме передатчика не используется. В режиме приемника (мастера/ведомого) содержит значение, которое передается в течение цикла отклика на запрос передатчика подтвердить прием. Передача нуля по окончании передачи байта (квитирование) означает, что данные успешно получены. Передача единицы (неквитирование) означает, что приемник не может продолжать работу по каким-либо причинам. Бит ACK очищается аппаратно по окончании цикла отклика
INTEN	2	Бит разрешения прерывания
		0   Запрещено
		1   Разрешено
STOP	1	Бит останова. В режиме мастера установка бита STOP генерирует состояние останова, которое завершает или прерывает текущую передачу. После прекращения передачи бит STOP очищается аппаратно
START	0	Бит старта. Этот бит устанавливается, когда требуется сформировать стартовое состояние на шине. Бит START очищается аппаратно по окончании цикла стартового состояния, а также при обнаружении ошибки на шине (состояние с кодом 1Fh)
–	31-8, 3	Зарезервировано

### ADDR – регистр собственного адреса

Смещение: + 10h

Сброс: 0h

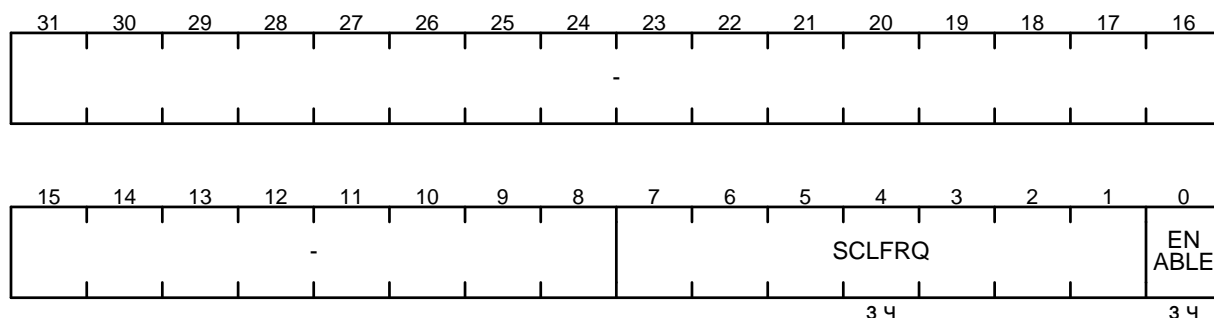


Поле	Биты	Описание
SAEN	7	Бит разрешения распознавания адреса
		0   Безадресный режим
		1   Включена функция распознавания принятого адреса
ADDR	6–0	Поле собственного 7-битного адреса. При работе в режиме ведомого первые 7 бит, принятые после стартового состояния, сравниваются со значением ADDR. Если обнаружено совпадение и установлен бит SAEN, ведомый переходит в режим приемника или передатчика (в зависимости от состояния бита направления R/W#)
–	31-8	Зарезервировано

## CTL1 – регистр управления 1

Смещение: + 14h

Сброс: 0h

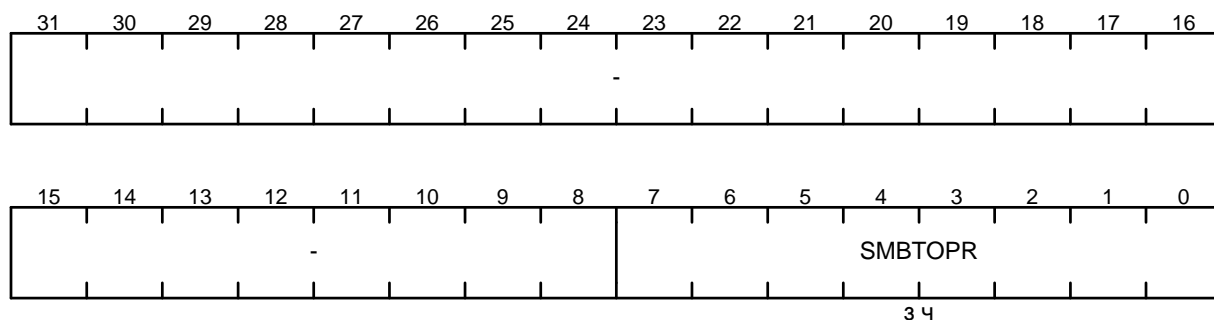


Поле	Биты	Описание
SCLFRQ	7-1	Младшие разряды поля выбора частоты $f_{SCL}$ сигнала на выводе I2Cx_SCL в режиме мастера. Длительности высокого $T_{SCLH}$ и низкого $T_{SCLL}$ уровней сигнала SCL зависят от тактовой частоты $F_{PCLK}$ модуля I2C и рассчитываются по формуле: $T_{SCLH} = T_{SCLL} = 2 \times SCLFRQ \times (1/F_{PCLK}).$ Таким образом, частота сигнала на выводе I2Cx_SCL равна $f_{SCL} = 1/(T_{SCLH} + T_{SCLL}).$ В поле SCLFRQ можно записать любое значение в диапазоне от 4h до 7FFFh (старшие разряды находятся в регистре CLT3). При попытке записи любого значения меньше 4h, оно будет записано со смещением 4h. Например, при записи числа 2h, к нему будет аппаратно добавлено смещение 4h и, в итоге, в поле SCLFRQ окажется значение 6h.
ENABLE	0	Бит включения модуля I2C 0   Модуль выключен. Тактирование не осуществляется. Регистры CTL0, ST, CST сброшены 1   Модуль включен
–	31-8	Зарезервировано

## TOPR – регистр загрузки делителя

Смещение: + 18h

Сброс: 0h

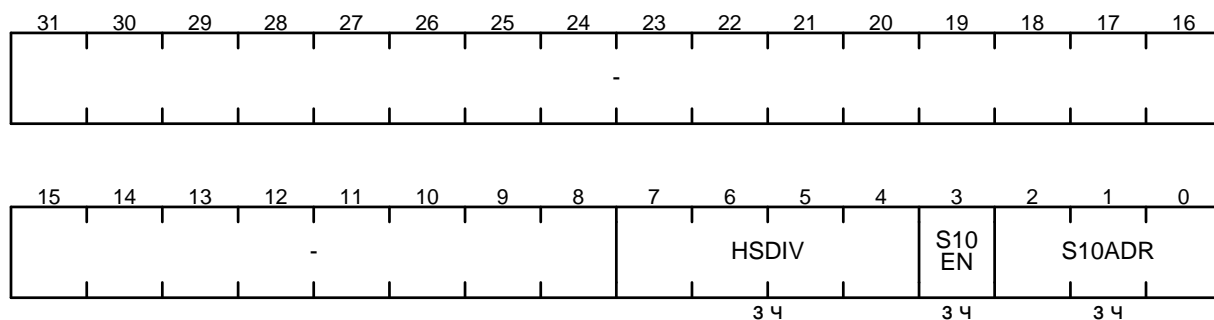


Поле	Биты	Описание
SMBTOPR	7-0	Поле значения перезагрузки делителя
–	31-8	Зарезервировано

## CTL2 – регистр управления 2

Смещение: + 1Ch

Сброс: 0h

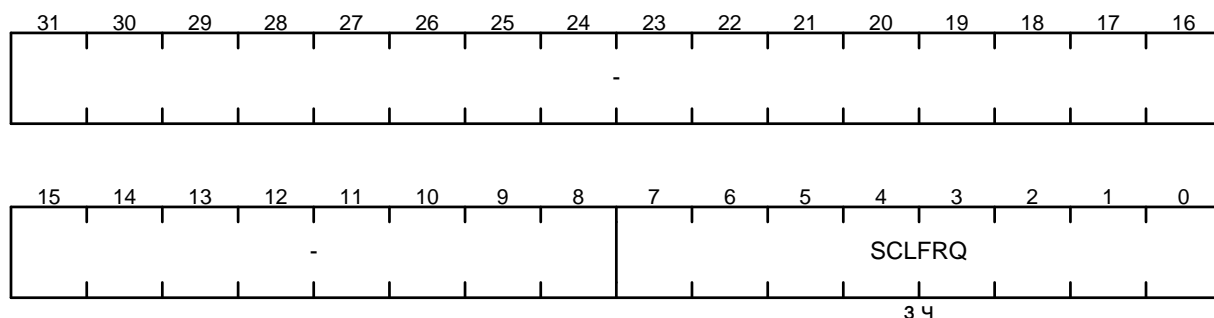


Поле	Биты	Описание		
HSDIV	7-4	<p>Младшие разряды поля выбора частоты <math>f_{SCL}</math> сигнала на выводе I2C_SCL в режиме HS мастера.</p> <p>Длительности высокого (<math>T_{HSCLH}</math>) и низкого (<math>T_{HSCLL}</math>) уровней сигнала на выводе I2C_SCL зависят от тактовой частоты <math>f_{PCLK}</math> модуля I2C и рассчитываются по формулам</p> $T_{HSCLH} = HSDIV \times (1 / f_{PCLK}),$ $T_{HSCLL} = 2 \times HSDIV \times (1 / f_{PCLK}).$ <p>Таким образом, частота сигнала на выводе I2C_SCL равна</p> $f_{SCL} = 1 / (T_{HSCLH} + T_{HSCLL}).$ <p>В поле HSDIV можно записать любое значение в диапазоне от 2h до 1000h (старшие разряды находятся в регистре CTL4). При попытке записи любого значения меньше 2 в поле HSDIV, оно будет записано со смещением 2. Например, при записи числа 1 к нему будет аппаратно добавлено смещение 2 и, в итоге, в поле HSDIV окажется значение 3.</p>		
S10EN	3	Бит разрешения 10-битной адресации ведомого		
		<table border="1"> <tr> <td>0</td> <td>Запрещено</td> </tr> <tr> <td>1</td> <td>Разрешена при условии, что установлен бит SAEN в регистре ADDR</td> </tr> </table>	0	Запрещено
0	Запрещено			
1	Разрешена при условии, что установлен бит SAEN в регистре ADDR			
S10ADR	2-0	<p>Поле старших битов 10-битного адреса ведомого.</p> <p>Поле содержит старшие три разряда адреса ведомого при 10-битной адресации.</p> <p>Первый принятый байт адреса сравнивается со значением [11110b, S10ADR[2:1]], второй байт адреса – со значением [S10ADR[0], ADDR]</p>		
–	31-8	Зарезервировано		

### CTL3 – регистр управления 3

Смещение: + 20h

Сброс: 0h

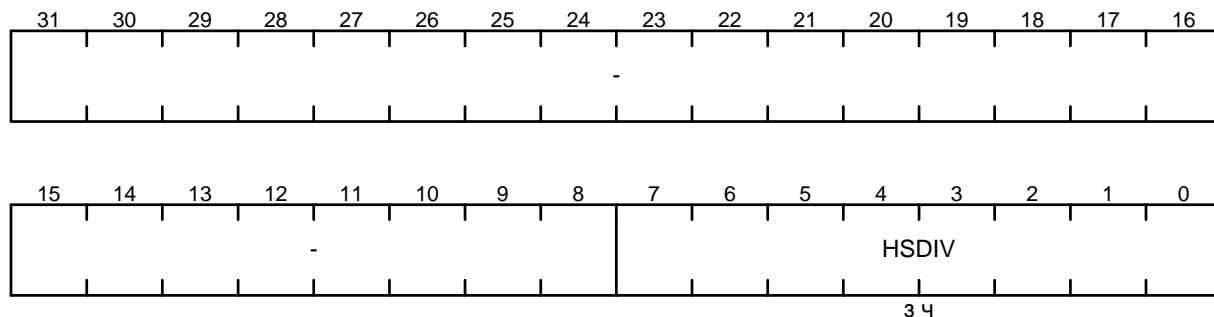


Поле	Биты	Описание
SCLFRQ	7-0	Старшие разряды делителя SCLFRQ. См. основное описание поля в регистре CTL1
–	31-8	Зарезервировано

### CTL4 – регистр управления 4

Смещение: + 24h

Сброс: 0h



Поле	Биты	Описание
HSDIV	7-0	Старшие разряды делителя HSDIV. См. основное описание поля в регистре CTL2
–	31-8	Зарезервировано

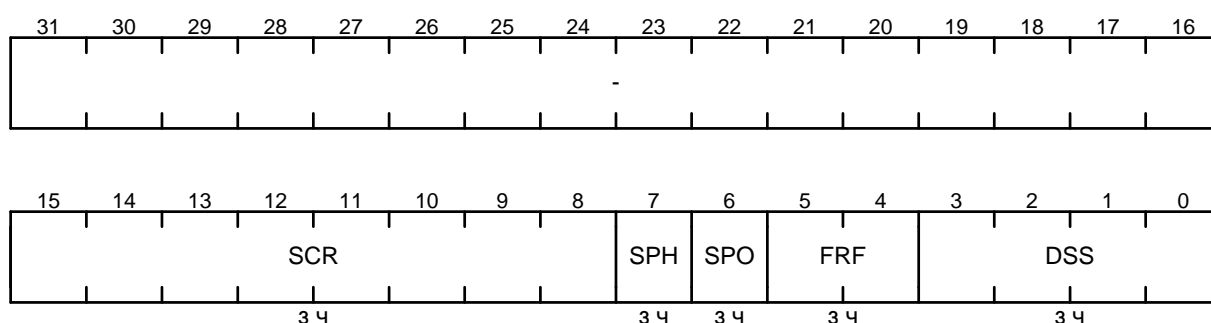
## А.26 Регистры контроллера интерфейса SPI

Базовый адрес:	4008_D000h	Регистры контроллера SPI 0
	4008_E000h	Регистры контроллера SPI 1
	4008_F000h	Регистры контроллера SPI 2
	4008_D000h	Регистры контроллера SPI 3

### CR0 – регистр управления 0

Смещение: + 00h

Сброс: 0h



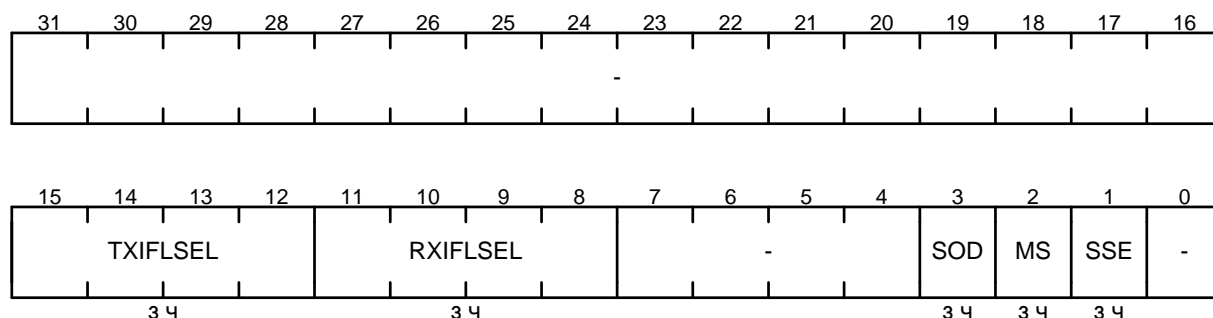
Поле	Биты	Описание
SCR	15-8	Коэффициент деления второго делителя. Может принимать значения 00h до FFh
SPH	7	0   Выборка данных по переднему фронту синхросигнала, а установка по заднему
		1   Выборка данных по заднему фронту синхросигнала, а установка по переднему
SPO	6	0   В режиме ожидания линия SPIx_SCK удерживается в состоянии логического нуля
		1   В режиме ожидания линия SPIx_SCK удерживается в состоянии логической единицы
FRF	5-4	Поле выбора протокола обмена информацией
		00   SPI
		01   SSI
		10   Microwire
		11   Зарезервировано
DSS	3-0	Размер слова данных
		0h-2h   Зарезервировано
		3h   4 бита
		4h   5 бит
		...   ...
		Eh   15 бит
Fh   16 бит		
–	31-16	Зарезервировано



## CR1 – регистр управления 1

Смещение: + 04h

Сброс: 4400h

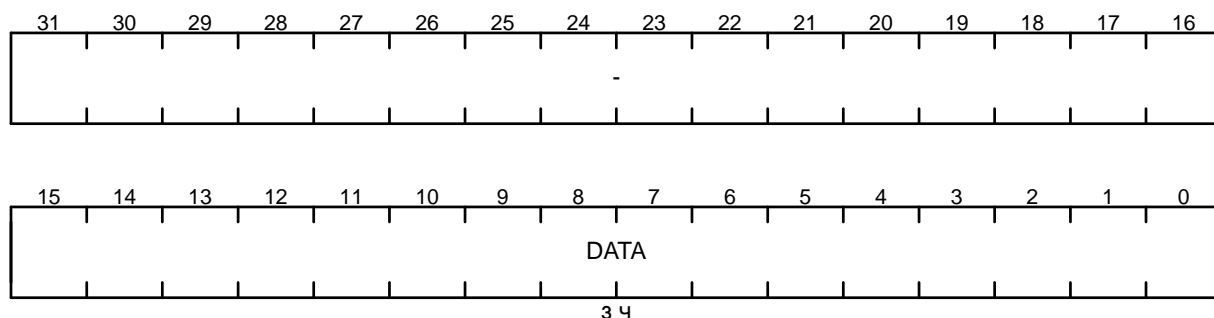


Поле	Биты	Описание
TXIFLSEL	15-12	Величина порога опустошения передающего FIFO. При опустошении до уровня порога или ниже может быть сгенерировано прерывание или запрос DMA. Допустимый диапазон значений 0h-8h (по умолчанию – 4h).
RXIFLSEL	11-8	Величина порога наполнения принимающего FIFO. При заполнении до уровня порога или выше может быть сгенерировано прерывание или запрос DMA. Допустимый диапазон значений 0h-8h (по умолчанию – 4h).
SOD	3	Бит запрета передачи данных. В режиме мастера значение бита игнорируется. В режиме ведомого бит контролирует выход данных. Пока бит сброшен передача и прием данных разрешены. Установка бита блокирует передачу данных и переводит вывод SPIx_TX в состояние слабой логической единицы, при этом прием тактового сигнала и прием данных не блокируется
MS	2	Бит выбора режима работы
		0   Мастер 1   Ведомый
SSE	1	Бит разрешения работы приемопередатчика
		0   Запрещено 1   Разрешено
–	31-16, 7-4, 0	Зарезервировано

## DR – регистр данных

Смещение: + 08h

Сброс: 0h

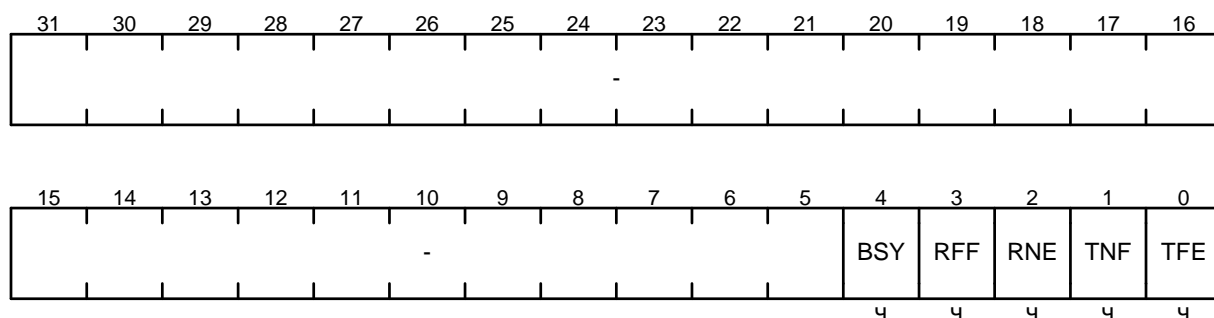


Поле	Биты	Описание
DATA	15-0	16-разрядный буфер FIFO приемника и передатчика. Данные для передачи записываются в регистр. Если размер данных менее 16 бит, они должны быть выравнены по правой границе. Принятые данные автоматически выравниваются по правой границе. Принятые данные возвращаются при чтении регистра.
–	31-16	Зарезервировано

## SR – регистр состояния

Смещение: + 0Ch

Сброс: 3h

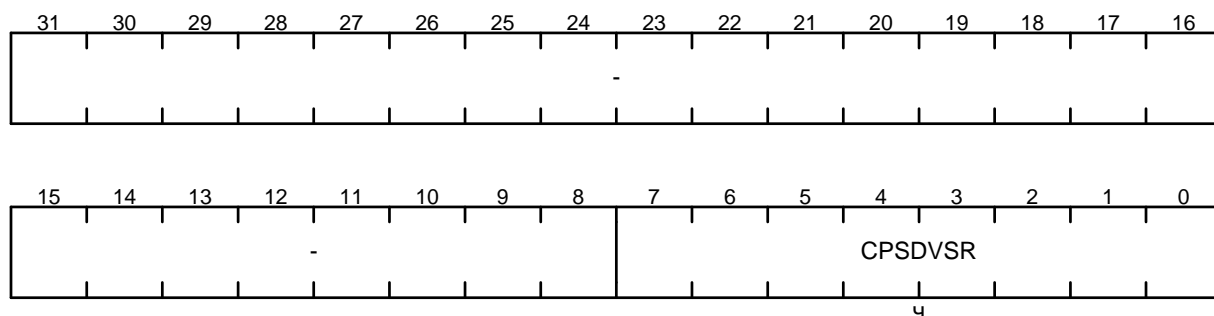


Поле	Биты	Описание	
BSY	4	0	Приемопередатчик не активен
		1	Передача/прием данных, либо буфер FIFO не пуст
RFF	3	0	Буфер FIFO приемника не заполнен
		1	Буфер FIFO приемника заполнен
RNE	2	0	Буфер FIFO приемника пуст
		1	Буфер FIFO приемника не пуст
TNF	1	0	Буфер FIFO передатчика заполнен
		1	Буфер FIFO передатчика не заполнен
TFE	0	0	Буфер FIFO передатчика не пуст
		1	Буфер FIFO передатчика пуст
–	31-5	Зарезервировано	

### CPSR – регистр делителя тактовой частоты

Смещение: + 10h

Сброс: 0h

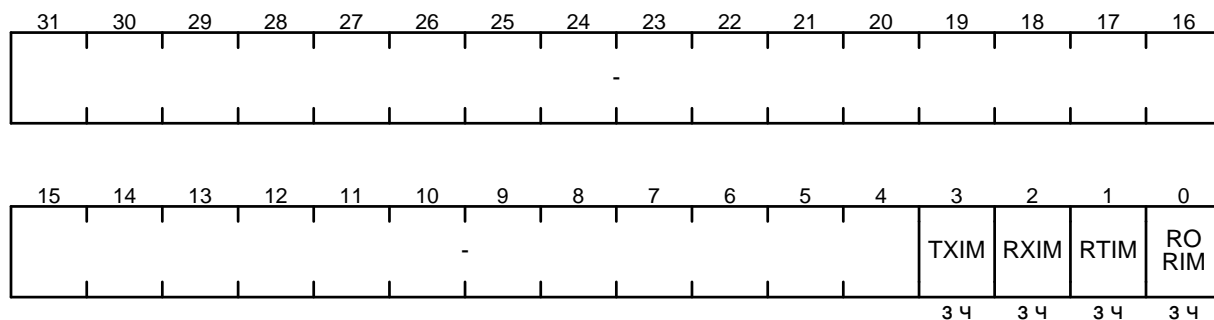


Поле	Биты	Описание
CPSDVSR	7-0	Коэффициент деления первого делителя. Может принимать четные значения от 02h до FEh
–	31-8	Зарезервировано

### IMSC – регистр маски прерываний

Смещение: + 14h

Сброс: 0h



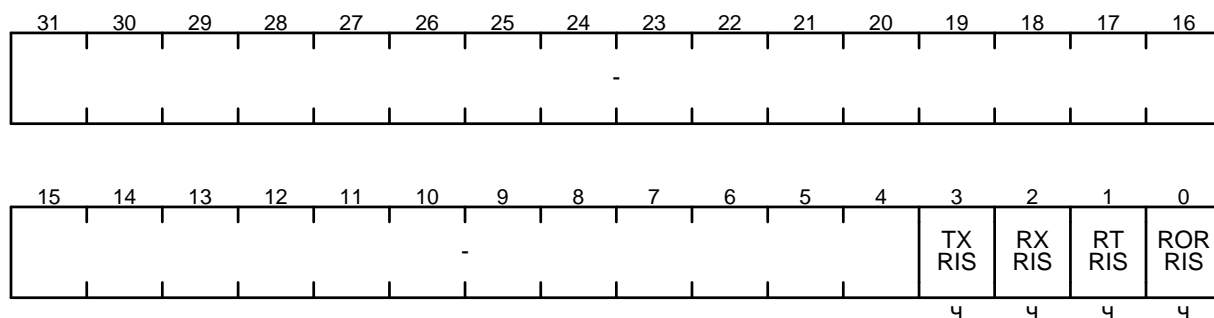
Поле	Биты	Описание
TXIM	3	Буфер передатчика опустошен до величины порога или ниже
RXIM	2	Буфер приемника заполнен на величину порога или выше
RTIM	1	Таймаут приема данных
RORIM	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – Установка/сброс бит формирует маску. По умолчанию, все биты сброшены, и установка флагов запрещена.

## RIS – регистр состояния прерываний

Смещение: + 18h

Сброс: 8h



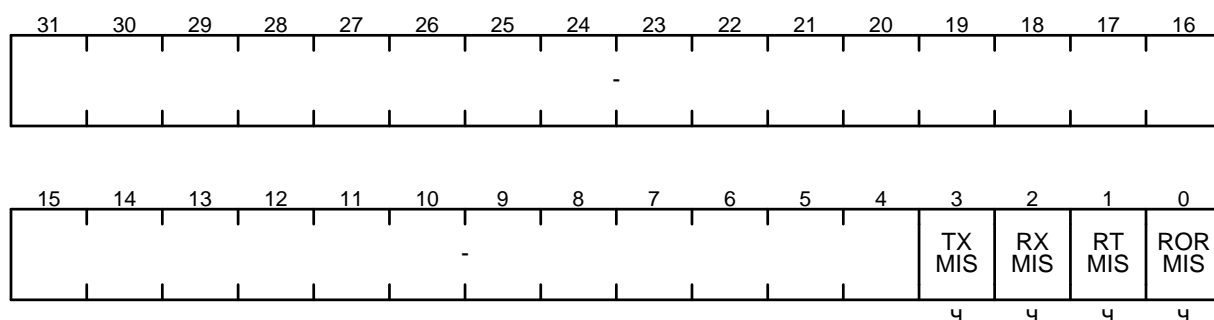
Поле	Биты	Описание
TXRIS	3	Буфер передатчика опустошен до величины порога или ниже
RXRIS	2	Буфер приемника заполнен на величину порога или выше
RTRIS	1	Таймаут приема данных
RORRIS	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – При возникновении прерываний устанавливаются соответствующие им немаскируемые флаги. Биты RTRIS также сбрасываются после чтения буфера приемника.

## MIS – регистр состояния прерываний с маскированием

Смещение: + 1Ch

Сброс: 0h



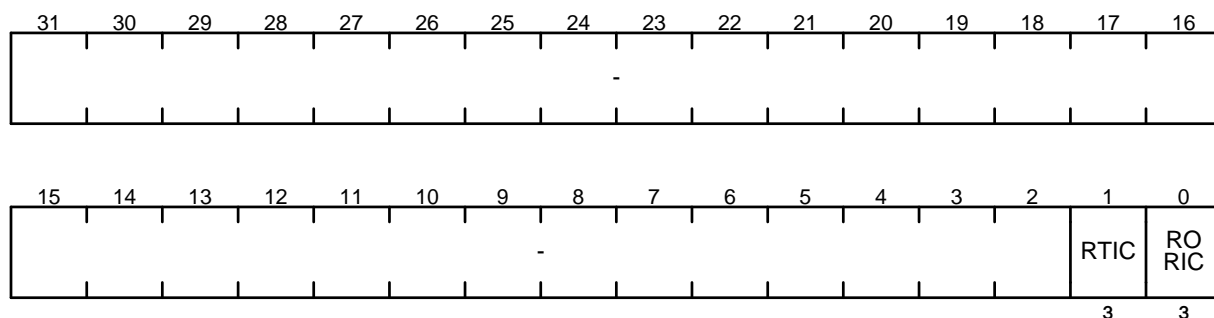
Поле	Биты	Описание
TXMIS	3	Буфер передатчика опустошен до величины порога или ниже
RXMIS	2	Буфер приемника заполнен на величину порога или выше
RTMIS	1	Таймаут приема данных
RORMIS	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – В регистре устанавливаются только те флаги, которые закрыты маской регистра IMSC. Бит RTMIS также сбрасывается после чтения буфера приемника.

## ICR – регистр сброса прерываний

Смещение: + 20h

Сброс: 0h



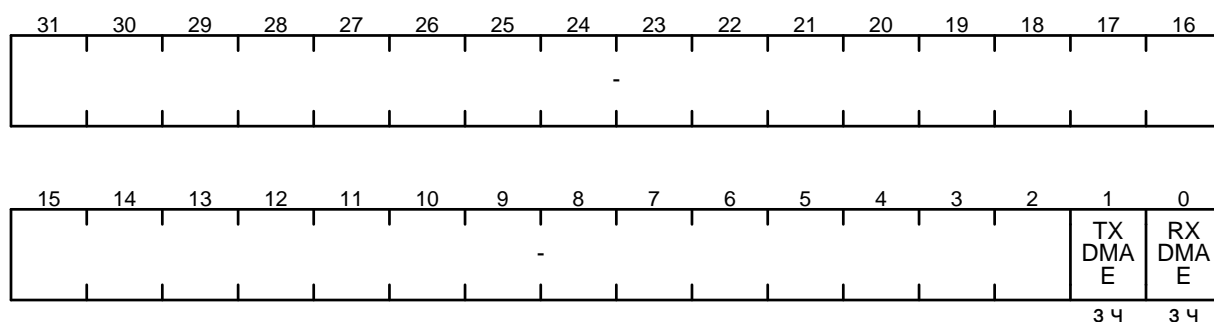
Поле	Биты	Описание
RTIC	1	Таймаут приема данных
RORIC	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – Запись единиц в биты регистра сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов.

## DMACR – регистр управления прямым доступом к памяти

Смещение: + 24h

Сброс: 0h



Поле	Биты	Описание
TXDMAE	1	Бит разрешения использования контроллера DMA при передаче
		0 Не используется
RXDMAE	0	Бит разрешения использования контроллера DMA при приеме
		0 Не используется
	1	Разрешено формирование запросов контроллера DMA для обслуживания буфера FIFO передатчика
		Разрешено формирование запросов контроллера DMA для обслуживания буфера FIFO приемника
–	31-2	Зарезервировано

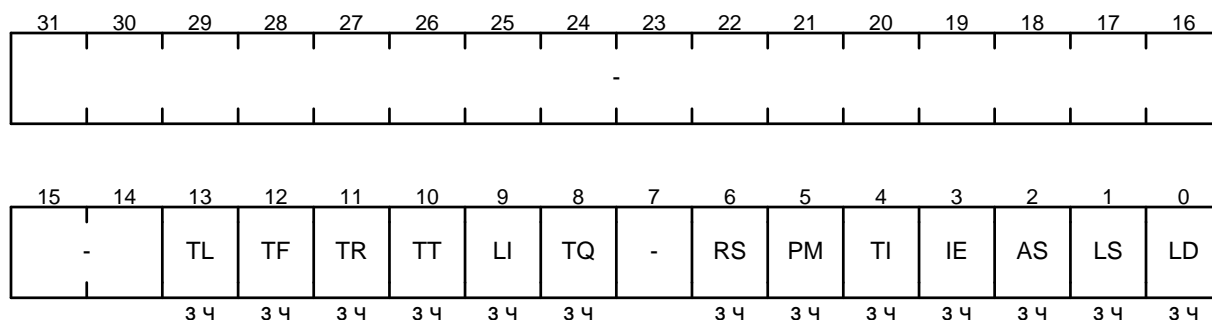
## A.27 Регистры контроллера интерфейса SpaceWire

Базовый адрес:	4007_0000h	Регистры SpaceWire0
	4007_1000h	Регистры SpaceWire1
	4007_4000h (SPWRTMR)	Регистры таймера временных отсчетов

### CTRL – регистр управления

Смещение: + 00h

Сброс: 0h



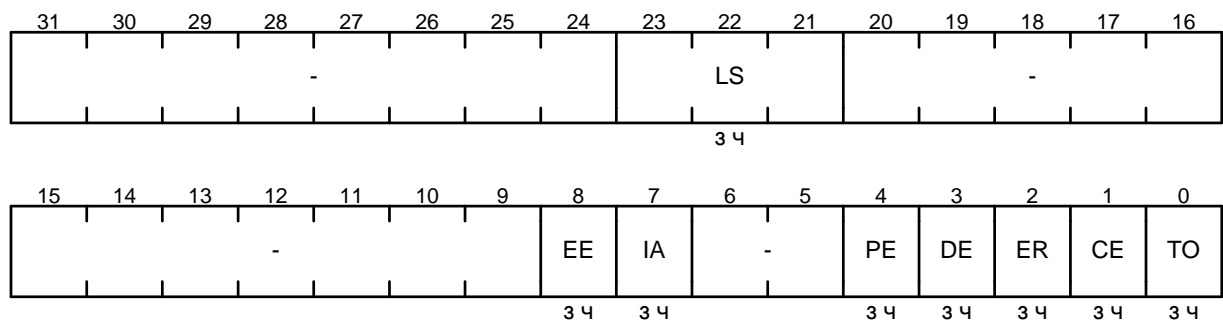
Поле	Биты	Описание	
TL	13	Включает или отключает функцию блокировки включения передатчика	
		0	Функция блокировки выключена
		1	Функция блокировки включена
TF	12	Фильтр флагов контроля тайм-кодов – если бит установлен, то достоверными считаются только тайм-коды с контрольными флагами «00». Если бит сброшен, то разрешены любые значения контрольных флагов	
TR	11	Прием тайм-кодов – разрешает прием кодов синхронизации	
TT	10	Передача тайм-кодов – разрешает передачу кодов синхронизации	
LI	9	Прерывание при ошибке связи – разрешение/запрет на генерацию прерывания при возникновении ошибки связи. Важно помнить, что бит IE также должен быть установлен, чтобы значение LI имело какое-либо влияние	
TQ	8	Прерывание при получении тайм-кода – разрешение/запрет на генерацию прерывания при приеме достоверного кода синхронизации (тайм-кода). Важно помнить, что бит IE также должен быть установлен, чтобы значение TQ имело какое-либо влияние	
RS	6	Сброс – полный сброс узла SpaceWire. С самоочисткой	
PM	5	«Неразборчивый» режим – разрешает «неразборчивый» режим приема сообщений	
TI	4	Входной отсчет – хост-система генерирует входной отсчет для интерфейса отправки таймкодов при записи «1» в данное поле. Инкрементируется счетчик синхронизации, и передается новое значение после передачи текущего символа	
IE	3	Разрешение прерывания – если установлено, то разрешена генерация прерываний для событий, которые индивидуально маскируются битами LI и TQ данного регистра	

Поле	Биты	Описание
AS	2	Автостарт – автоматический старт связи при приеме NULL
LS	1	Старт соединения – старт связи, т.е. разрешение передачи, перевод из режима готовности в режим запуска
LD	0	Запрет соединения
		0   Кодек SpaceWire разрешен
		1   Кодек SpaceWire запрещен
–	31-14, 7	Зарезервировано

### STAT – регистр статуса

Смещение: + 04h

Сброс: 0h

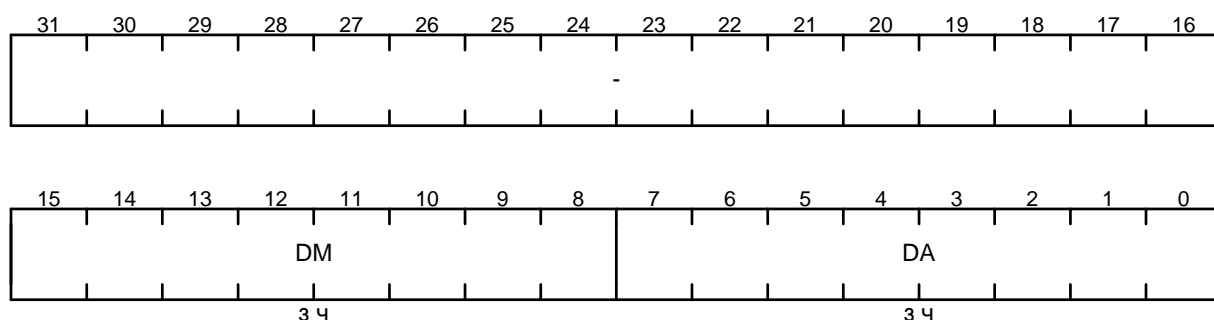


Поле	Биты	Описание
LS	23-21	Состояние соединения
		000   Сброс (ErrorReset)
		001   Ожидание после сброса (ErrorWait)
		010   Готово (Ready)
		011   Запущено (Started)
		100   Соединение (Connecting)
		101   Рабочий режим (Run)
EE	8	Преждевременный EOP/EEP – устанавливается, если пакет с EOP принят после первого байта пакета
IA	7	Недопустимый адрес – устанавливается, если пакет принят с недействительным полем адреса назначения, т.е. в случае несоответствия содержимому регистра адреса по умолчанию
PE	4	Ошибка четности
DE	3	Ошибка при разрыве соединения
ER	2	Ошибка ESC-последовательности
CE	1	Ошибка кредитования при разрешении на передачу
TO	0	Выходной отсчет – принято новое значение счетчика времени, сохранено в поле счетчика времени
–	31-24, 20-9, 6-5	Зарезервировано

## DEFADDR – регистр адреса по умолчанию

Смещение: + 08h

Сброс: 0000\_00FEh

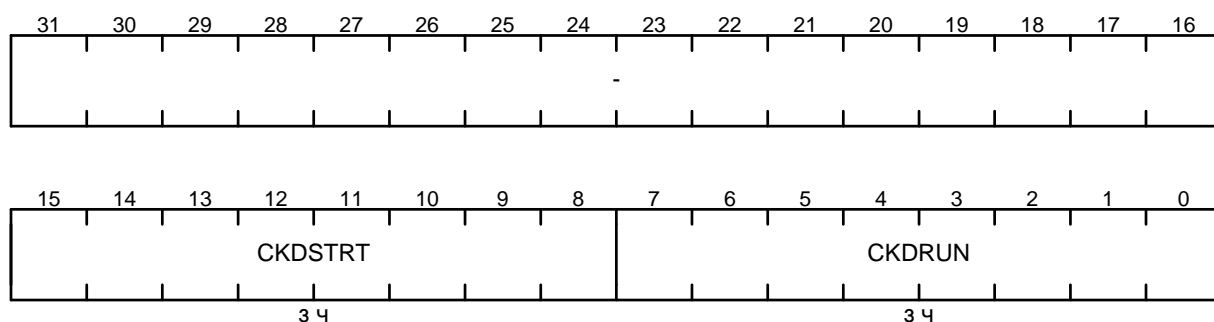


Поле	Биты	Описание
DM	15-8	Маска по умолчанию – Используется для идентификации узла в сети SpaceWire. Оба поля адреса – принятый адрес и DEFADDR логически умножаются с инверсией поля DEFMASK перед проверкой адреса
DA	7-0	Адрес узла по умолчанию – 8-битовый адрес узла, используемый для идентификации узла в сети SpaceWire
–	31-16	Зарезервировано

## CLKDIV – регистр тактового делителя

Смещение: + 0Ch

Сброс: 0000\_0404h



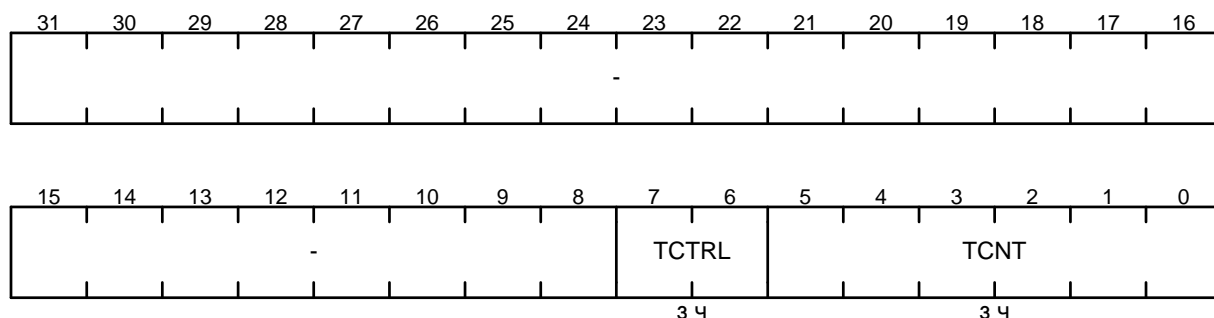
Поле	Биты	Описание
CKDSTRT	15-8	Значение тактового делителя при старте – значение, используемое для тактового делителя при запуске линии связи (любое из состояний соединения, кроме рабочего режима). Фактическое значение делителя: CKDSTRT + 1
CKDRUN	7-0	Значение тактового делителя в рабочем режиме – значение, используемое для тактового делителя, когда интерфейс связи находится в рабочем режиме. Фактическое значение делителя: CKDRUN + 1
–	31-16	Зарезервировано



## TIMECODE – регистр системного времени

Смещение: + 14h

Сброс: 0h

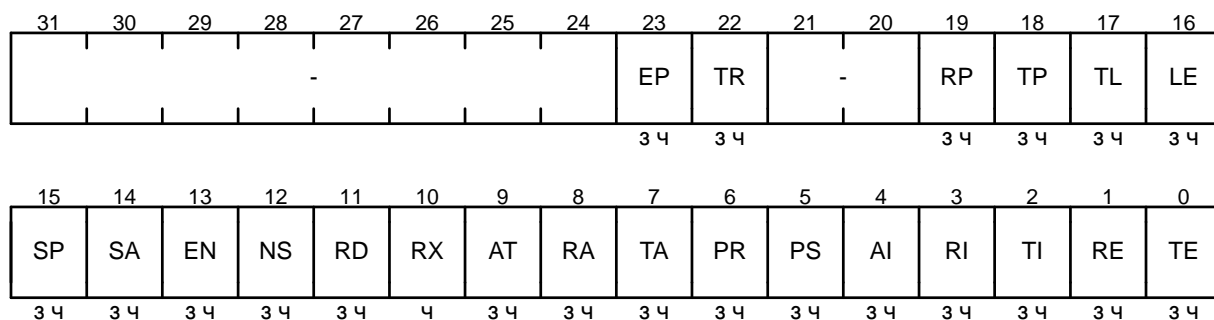


Поле	Биты	Описание
TCTRL L	7-6	Флаги контроля тайм-кодов – текущее значение флагов контроля системного времени (тайм-кодов). Они отсылаются в составе тайм-кода при установке бита TI регистра управления. Также данное поле обновляется при каждом получении тайм-кода
TCNT	5-0	Счетчик времени – текущее значение счетчика системного времени. Значение инкрементируется и передается в составе тайм-кода при установке бита TI регистра управления CTRL. Также данное поле обновляется при каждом получении тайм-кода. Важно понимать, что при непосредственной записи в данный регистр выставленное значение не будет передано, поскольку перед передачей оно будет проинкрементировано
–	31-8	Зарезервировано

## DMACSR – регистр управления/статуса канала DMA

Смещение: + 20h

Сброс: 0h



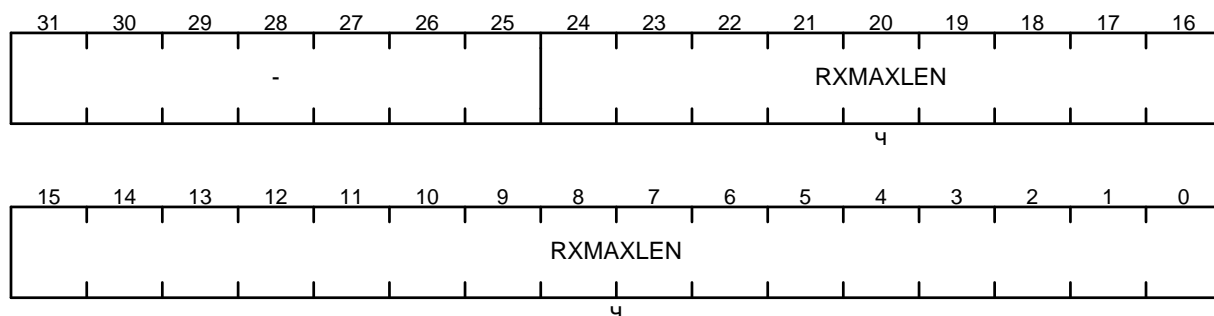
Поле	Биты	Описание
EP	23	EEP окончание – устанавливается, когда завершающим символом полученного пакета является EEP символ (пакет полчен с ошибкой)
TR	22	Усеченный – устанавливается, когда полученный пакет канала DMA сохраняется не целиком из-за превышения установленной максимальной длины
RP	19	Прерывание при получении пакета – бит устанавливается, когда генерируется прерывание, вызванное получением пакета
TP	18	Прерывание при получении пакета – бит устанавливается, когда генерируется прерывание, вызванное получением пакета
TL	17	Разрешение блокировки передатчика – бит устанавливается, если бит TL установлен, и передатчик соответствующего канала DMA запрещён из-за ошибки соединения (обеспечивается битом LE данного регистра). Пока данный бит установлен, невозможно перезапустить передатчик (т.е. нельзя установить бит TE данного регистра)
LE	16	Запрещение работы при ошибке соединения – запрет на дальнейшую передачу при возникновении ошибки соединения. До повторного установления бита разрешения передатчика, пакеты передаваться не будут
SP	15	Отделять PID – удаление PID-байта (второго байта) каждого пакета. Если данный бит установлен, то адресный байт (первый байт) будет также удален, вне зависимости от значения бита SA данного регистра
SA	14	Отделять адрес – удаление адресного байта (первого байта) каждого пакета
EN	13	Разрешение адреса – разрешение отдельного адреса узла для этого канала
NS	12	Без вытеснения – если бит сброшен, то получаемые пакеты будут отбракованы при отсутствии активированного дескриптора. Если бит установлен, то кодек будет ожидать активации дескриптора
RD	11	Доступен дескриптор приёмника – устанавливается для указания того, что в таблице дескрипторов имеются активированные дескрипторы. Сбрасывается аппаратно в случае обнаружения неактивированного дескриптора

Поле	Биты	Описание
RX	10	Осуществляется приём – устанавливается, если в текущий момент осуществляется приём по данному каналу DMA; в противном случае, бит сброшен
AT	9	Прервать передачу – установка данного бита прервет текущую передачу пакета и снимет разрешение на передачу. Если на текущий момент передача не производится, то произойдёт только снятие разрешения на передачу
RA	8	Ошибка АНВ при приёме – шина АНВ отреагировала ошибкой на попытку доступа при приёме через данный канал DMA
TA	7	Ошибка АНВ при передаче – шина АНВ отреагировала ошибкой на попытку доступа при передаче через данный канал DMA
PR	6	Пакет принят – этот бит устанавливается каждый раз, когда пакет получен
PS	5	Пакет отправлен – этот бит устанавливается каждый раз, когда пакет отправлен
AI	4	Прерывание из-за ошибки АНВ – если установлено, то при каждом возникновении ошибки на шине АНВ из-за попытки доступа по данному каналу DMA генерируется прерывание
RI	3	Прерывание при приеме – если установлено, то при наличии установленного бита разрешения прерывания (IE) соответствующего дескриптора при каждом приеме пакета генерируется прерывание. Это происходит вне зависимости от того заканчивается пакет EEP или EOP
TI	2	Прерывание при передаче – если установлено, то при наличии установленного бита разрешения прерывания (IE) соответствующего дескриптора при каждой передаче пакета генерируется прерывание. Это происходит вне зависимости от того, успешно прошла передача или нет
RE	1	Разрешение приёма – установка данного бита разрешает прием пакетов по данному каналу
TE	0	Разрешение передачи – разрешает передачу для соответствующего канала DMA. Запись «1» вызывает процедуру считывания узлом SpaceWire нового дескриптора и попытку отправки пакета адресату. Важно помнить, что бит TE можно установить, только если бит TL данного регистра сброшен. Бит TE автоматически сбрасывается при обнаружении узлом SpaceWire неактивированного дескриптора или если во время передачи пакета произошла ошибка соединения (устанавливается бит LE данного регистра)
–	31-24, 21-20, 7	Зарезервировано

### DMARXLEN – регистр максимального размера принимаемого пакета

Смещение: + 24h

Сброс: 0h

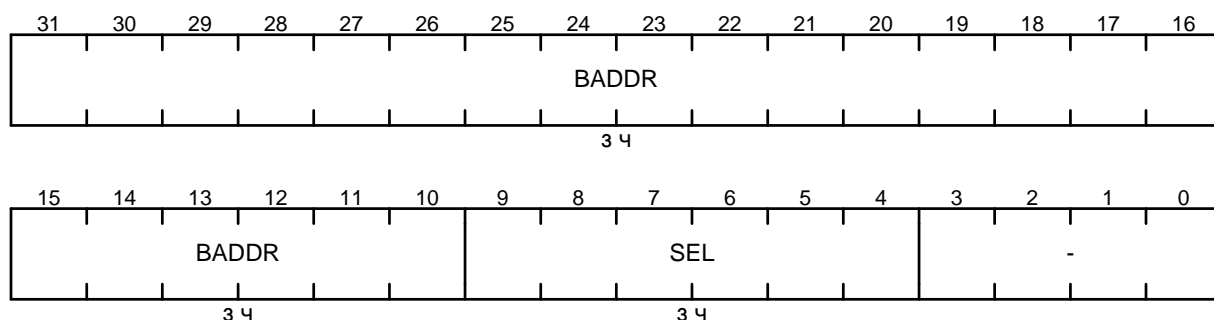


Поле	Биты	Описание
RXMAXLEN	24-0	Максимальный размер пакета приёмника в байтах. Минимальное значение поля RXMAXLEN равно 4
–	31-16	Зарезервировано

### DMATDTADDR – регистр адреса таблицы дескрипторов передатчика

Смещение: + 28h

Сброс: 0h

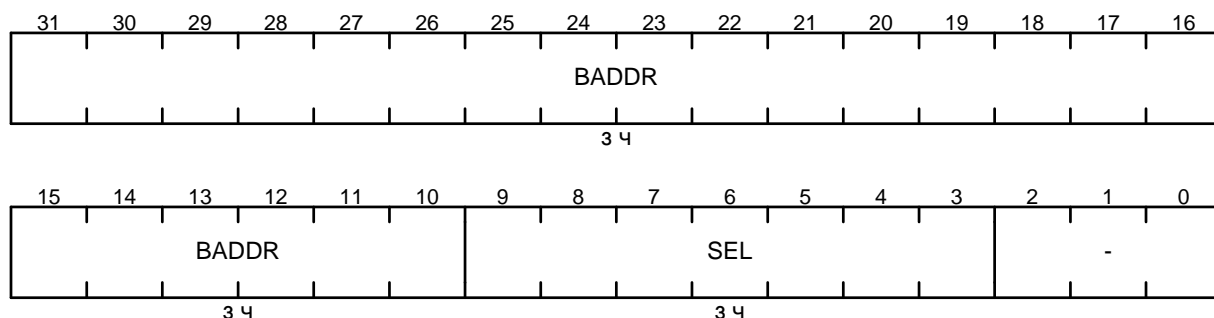


Поле	Биты	Описание
BADDR	31-10	Базовый адрес таблицы дескрипторов
SEL	9-4	Смещение в таблице дескрипторов. Показывает текущий используемый узлом SpaceWire дескриптор. Для чтения каждого нового дескриптора селектор увеличивается на 16(в расчет принимается и зарезервированное поле 3–0) и конечном итоге обнуляется
–	3-0	Зарезервировано

### DMARDTADDR – регистр адреса таблицы дескрипторов приемника

Смещение: + 2Ch

Сброс: 0h

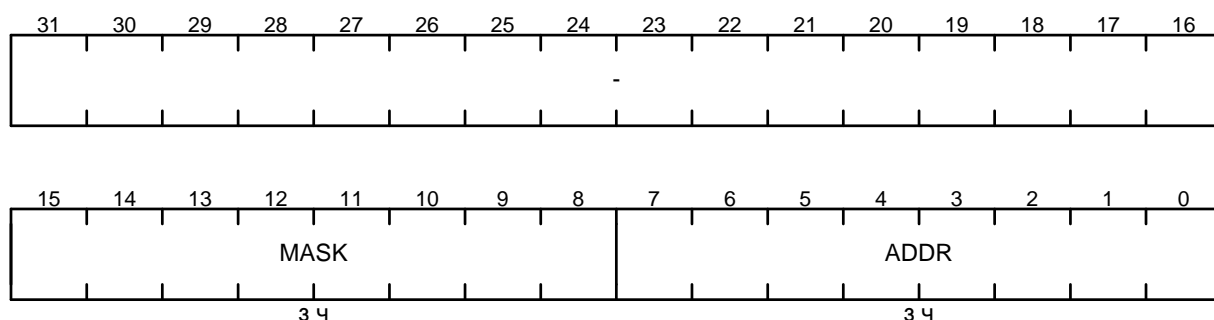


Поле	Биты	Описание
BADDR	31-10	Базовый адрес таблицы дескрипторов
SEL	9-3	Смещение в таблице дескрипторов. Показывает текущий используемый узлом SpaceWire дескриптор. Для чтения каждого нового дескриптора селектор увеличивается на 8(в расчет принимается и зарезервированное поле 2–0) и конечном итоге обнуляется
–	2-0	Зарезервировано

### DMAADDR – регистр адреса канала DMA

Смещение: + 30h

Сброс: 0h

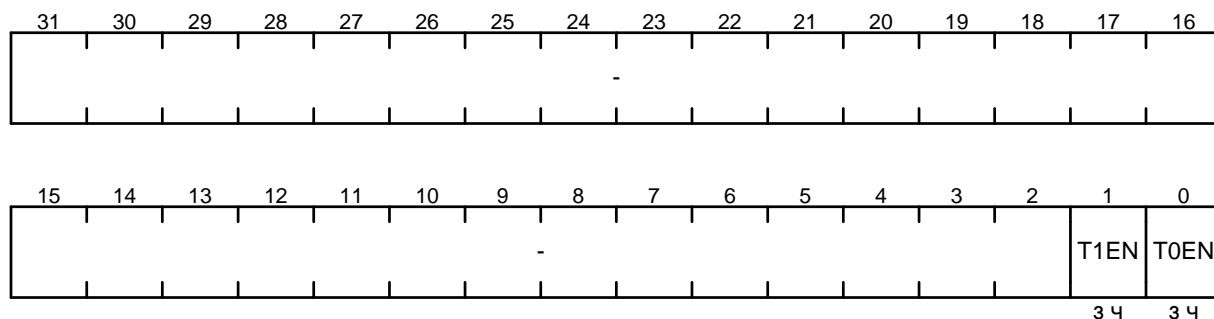


Поле	Биты	Описание
MASK	15-8	Маска – используется для идентификации узла в сети SpaceWire. Оба поля адреса – принятый адрес и ADDR логически умножаются с инверсией поля MASK перед проверкой адреса
ADDR	7-0	Адрес – если установлен бит EN регистра управления/статуса канала DMA (DMACSR), то данный адрес используется для идентификации узла в сети SpaceWire для данного канала DMA
–	31-16	Зарезервировано

### CFG – регистр конфигурации таймера временных отсчетов

Смещение: SPWRTMR + 00h

Сброс: 0h

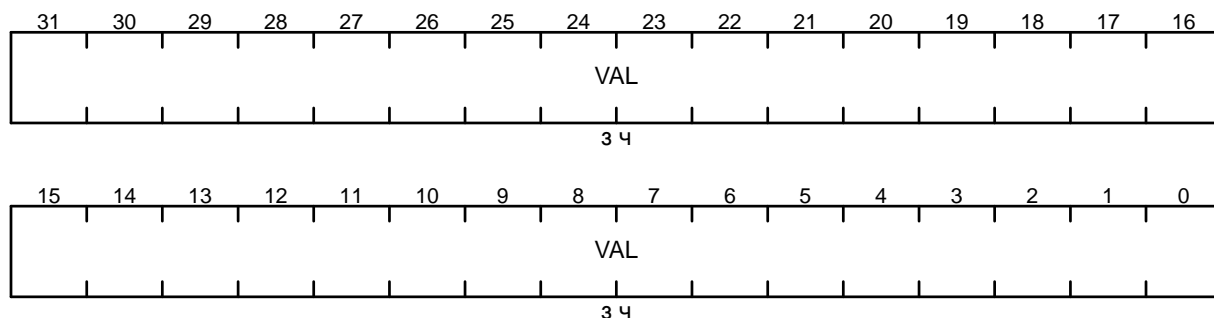


Поле	Биты	Описание
T1EN	1	Бит разрешения работы таймера подачи временных отсчетов для блока SpaceWire1
T0EN	0	Бит разрешения работы таймера подачи временных отсчетов для блока SpaceWire0
–	31-2	Зарезервировано

### REL0 – регистр начального значения счетчика таймера 0

Смещение: SPWRTMR + 04h

Сброс: 0h

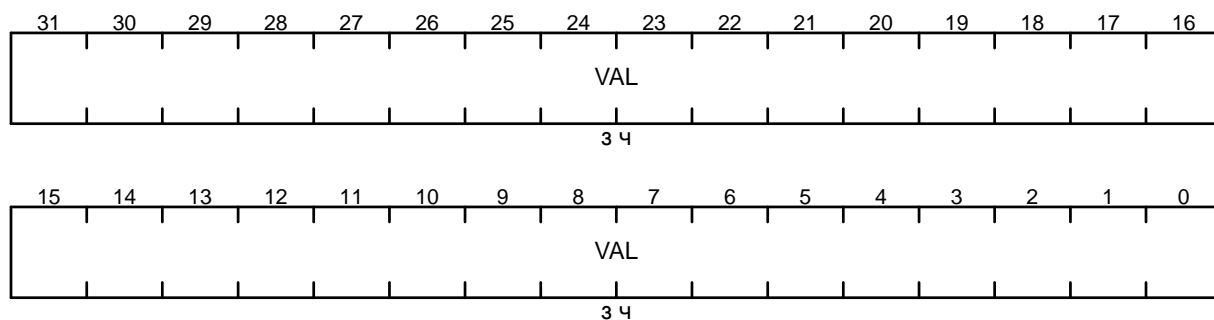


Поле	Биты	Описание
VAL	31-0	Значение для загрузки счетчика 0 после переполнения

## REL1 – регистр начального значения счетчика таймера 1

Смещение: SPWRTMR + 08h

Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Значение для загрузки счетчика 1 после переполнения

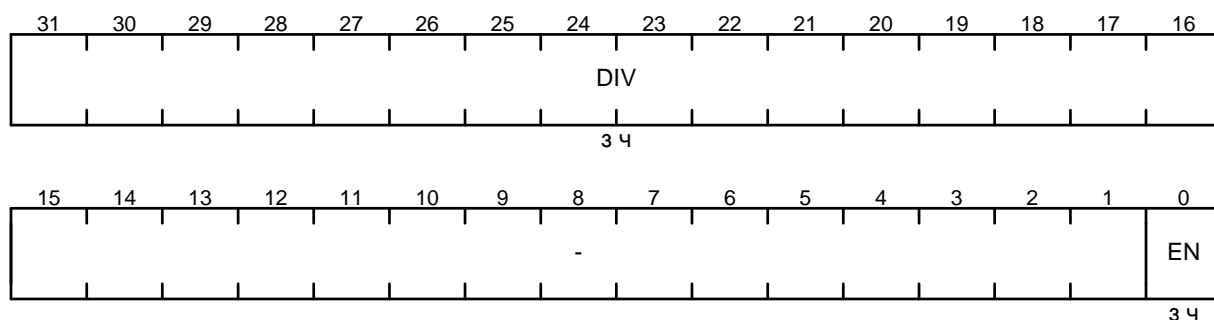
## A.28 Регистры контроллера интерфейса OWI

**Базовый адрес:** 400A\_F000h      Регистры контроллера OneWire 0  
 400B\_0000h      Регистры контроллера OneWire 1

### CTRL0 – регистр управления 0

Смещение: + 00h

Сброс: 0h

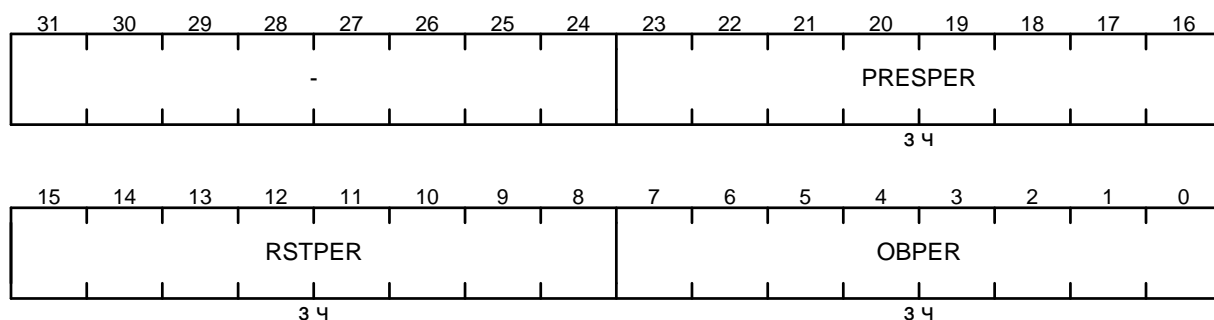


Поле	Биты	Описание
DIV	31-16	Значение коэффициента делителя частоты для формирования длительности одного стартового импульса (Tsp)
EN	0	Бит разрешения работы контроллера интерфейса
		0   Запрещено
		1   Разрешено
–	15-1	Зарезервировано

### CTRL1 – регистр управления 1

Смещение: + 04h.

Сброс: 0h



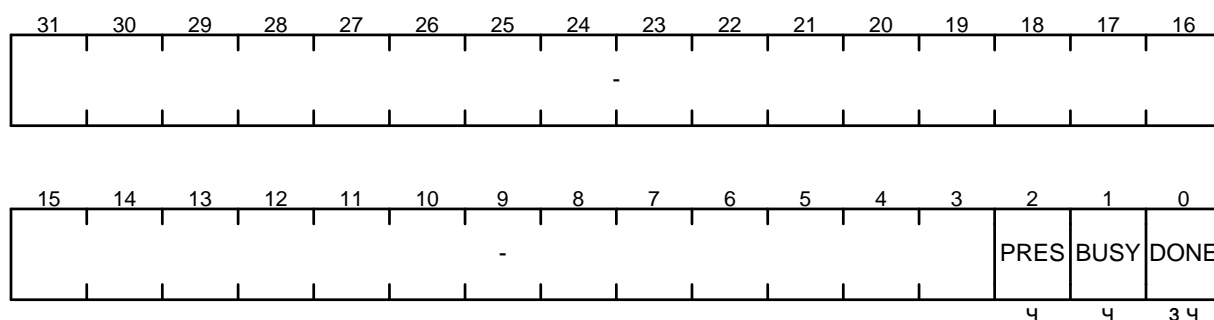
Поле	Биты	Описание
PRESER	23-16	Поле задания времени ожидания импульса присутствия от ведомого устройства в Tsp
RSTPER	15-8	Поле задания длительности последовательности сброса в Tsp
OPPER	7-0	Поле задания длительности одного принимаемого бита в Tsp
–	31-24	Зарезервировано



## STAT – регистр состояния

Смещение: + 08h

Сброс: 0h

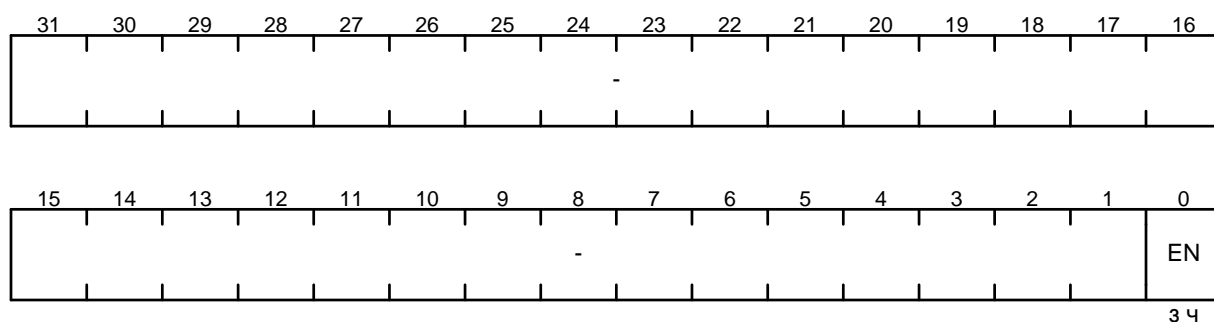


Поле	Биты	Описание
PRES	2	Флаг присутствия ведомого устройства
		0   Ведомое устройство не обнаружено
		1   Получен импульс присутствия от ведомого устройства
BUSY	1	Флаг занятости контроллера OneWire
		0   Контроллер не занят
		1   Контроллер выполняет транзакцию. Флаг снимается автоматически после завершения работы контроллера интерфейса
DONE	0	Флаг завершения транзакции
		0
		1   Транзакция завершена. Если в регистре IRQ установлен бит EN, то выставится сигнал прерывания. Для снятия сигнала прерывания следует записать «1» в бит DONE.
–	31-3	Зарезервировано

## IRQ – регистр прерываний

Смещение: + 0Ch.

Сброс: 0h

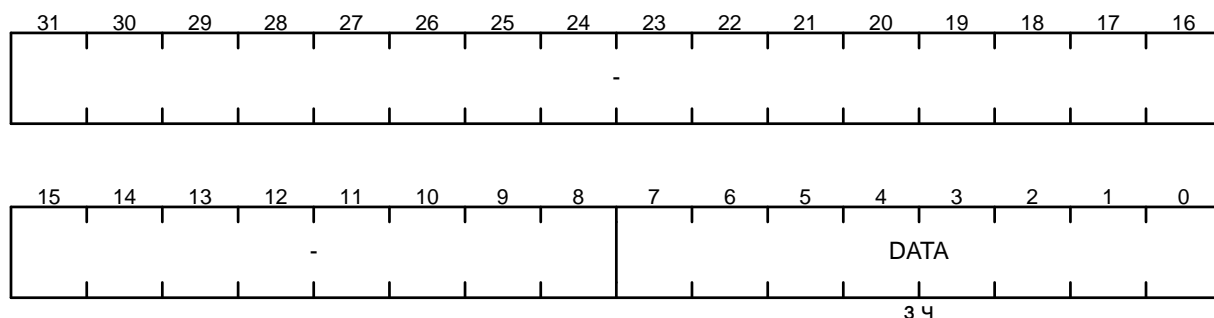


Поле	Биты	Описание
EN	0	Бит разрешения прерывания по завершению транзакции
		0   Запрещено
		1   Разрешено
–	31-1	Зарезервировано

## DATA – регистр данных

Смещение: + 10h

Сброс: 0h

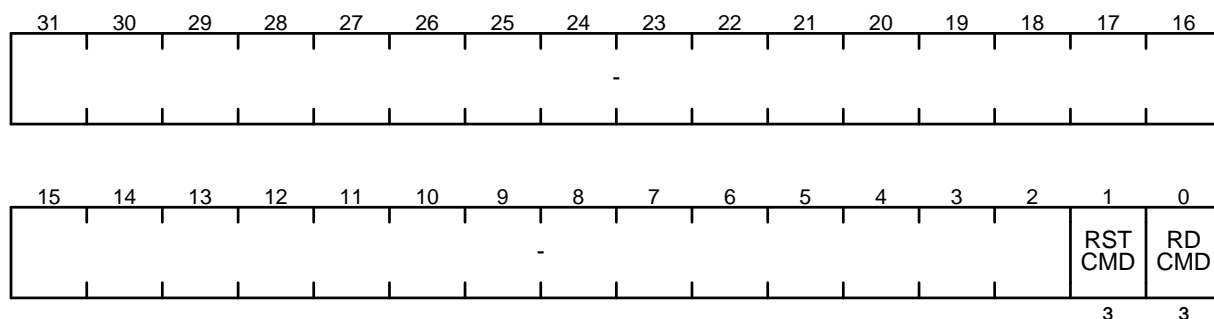


Поле	Биты	Описание
DATA	7-0	Поле данных. Результатом записи в поле DATA является размещение байта в буфере передатчика и начало передачи, а результатом чтения – считывание байта из буфера приемника
–	31-6	Зарезервировано

## CMD – регистр команд

Смещение: + 14h

Сброс: 0h



Поле	Биты	Описание
RSTCMD	1	Бит сброса
		0   Нет действий
	1	Контроллер начинает передачу последовательности сброса линии
RDCMD	0	Бит запуска чтения
		0   Нет действий
	1	Контроллер запускает процесс чтения 8 бит данных от ведомого устройства
–	31-2	Зарезервировано

## Приложение Б (обязательное) Коды состояний функционирования блока I2C

В таблицах Б.1 – Б.11 представлена информация о соответствии кодов и операций.

Условные обозначения, принятые в данных таблицах:

- [ADR, 0], [ADR, 1] – 8-разрядное значение, состоящее из 7-разрядного адреса ADR и бита направления передачи R/W#, значение которого «0» или «1» указывается непосредственно;

- DAT – байт данных;

- код мастера – 8-разрядное значение 0000\_1xxxh, где «xxx» – уникальный код каждого мастера в системе нескольких устройств;

- «с ACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает подтверждение передачи от ведомого (квитирование);

- «с NACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает неподтверждение передачи от ведомого (неквитирование);

- X – бит может быть установленным (1b) или сброшенным (0b), в зависимости от режима работы, состояния и дальнейших действий модуля I2C.

Таблица Б.1 – Исключительные состояния

Код	Описание состояния	Регистр SDA	Биты регистра CTLO				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
00h	IDLE	–	–	–	–	–	Ожидать завершения текущей передачи байта
1Fh	Ошибка на шине	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)

Таблица Б.2 – Режим FS мастера передатчика (дополнительно см. таблицу Б.4)

Код	Описание состояния	Регистр SDA	Биты регистра CTLO				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
01h	Старт	Код мастера	1	0	0	0	Передать код мастера и перейти в режим HS (0Ch/ 21h)
		[ADR, 0]					
02h	Повторный Старт	[ADR, 0]	1	0	0	0	Передать адрес ведомого (04h/ 05h)
		[ADR, 1]					Передать адрес ведомого, после чего перейти в режим приемника (08h/ 09h)

Окончание таблицы Б.2

1	2	3	4	5	6	7	8
03h	Потеря арбитража, мастер перешел в режим безадресного ведомого	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
04h	Отправлен адрес ведомого с ACK	DAT	1	0	0	0	Передать байт данных (06h/ 07h)
		–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
05h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
06h	Отправлен байт данных с ACK	DAT	1	0	0	0	Передать байт данных (06h/ 07h)
		–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
07h	Отправлен байт данных с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.3 – Режим FS мастера приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
08h	Отправлен адрес ведомого с ACK	–	1	0	0	0	Получить байт данных, квитировать прием (0Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (0Bh)
09h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
0Ah	Принят байт данных и квитирован	DAT	1	0	0	0	Получить байт данных, квитировать прием (0Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (0Bh)

Окончание таблицы Б.3

1	2	3	4	5	6	7	8
0Bh	Принят байт данных и не квити-рован	DAT	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.4 – Режим FS мастера передатчика (дополнительно см. таблицу Б.2)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
0Ch	Отправлен код мастера, обнаружена ошибка (ACK)	-	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.5 – Режим FS ведомого приемника (дополнительно см. таблицу Б.7)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
10h	Принят адрес и квити-рован	-	1	0	0	0	Получить байт данных, квити-ровать прием (12h)
			1	1	0	0	Получить байт данных, не квити-ровать прием (13h)
11h	Принят адрес после-потери арбитража и квити-рован	-	1	0	0	0	Получить байт данных, квити-ровать прием (12h)
			1	1	0	0	Получить байт данных, не квити-ровать прием (13h)
12h	Принят байт данных и квити-рован	DAT	1	0	0	0	Получить байт данных, квити-ровать прием (12h)
			1	1	0	0	Получить байт данных, не квити-ровать прием (13h)
13h	Принят байт данных и не квити-рован	DAT	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.6 – Режим FS ведомого передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
14h	Принят адрес и квитирован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
15h	Принят адрес после потери арбитража и квитирован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
16h	Отправлен байт данных с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
17h	Отправлен байт данных с NACK	–	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)
18h	Принят адрес отклика и квитирован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)
19h	Принят адрес отклика после потери арбитража и квитирован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)
1Ah	Отправлен байт данных в ответ на получение адреса отклика с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)

Окончание таблицы Б.6

1	2	3	4	5	6	7	8
1Bh	Отправлен байт данных в ответ на получение адреса отклика с NACK	–	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.7 – Режим FS ведомого приемника (дополнительно см. таблицу Б.5)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1Ch	Стоп	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)
1Dh	Принят адрес общего вызова и квитирован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)
1Eh	Принят адрес общего вызова после потери арбитража и квитирован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)

Таблица Б.8 – Режим HS мастера передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTLO				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
21h	Успешно отправлен код мастера, мастер перешел в режим HS	–	1	0	0	1	Сделать повторный старт (22h)
22h	Повторный старт	[ADR, 0]	1	0	0	0	Передать адрес ведомого (28h/29h)
		[ADR, 1]					Передать адрес ведомого, после квитирования/не квитирования переключиться в режим мастера приемника (28h/29h)
23h	Потеря арбитража, мастер перешел в режим HS безадресного ведомого	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
24h	Отправлен адрес ведомого с ACK	DAT	1	0	0	0	Передать байт данных (26h/27h)
		–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
25h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
26h	Отправлен байт данных с ACK	DAT	1	0	0	0	Передать байт данных (26h/27h)
		–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
27h	Отправлен байт данных с NACK	–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)



Таблица Б.9 – Режим HS мастера приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
28h	Отправлен адрес ведомого с ACK	–	1	0	0	0	Получить байт данных, квитировать прием (2Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (2Bh)
29h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
2Ah	Принят байт данных и квитирован	DAT	1	0	0	0	Получить байт данных, квитировать прием (2Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (2Bh)
2Bh	Принят байт данных и не квитирован	DAT	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.10 – Режим HS ведомого приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
30h	Принят адрес и квитирован	–	1	0	0	0	Получить байт данных, квитировать прием (32h)
			1	1	0	0	Получить байт данных, не квитировать прием (33h)
32h	Принят байт данных и квитирован	DAT	1	0	0	0	Получить байт данных, квитировать прием (32h)
			1	1	0	0	Получить байт данных, не квитировать прием (33h)
33h	Принят байт данных и не квитирован	DAT	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.11 – Режим HS ведомого передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
34h	Принят адрес и квитирован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (36h/37h)
36h	Отправлен байт данных с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (36h/37h)
37h	Отправлен байт данных с NACK	-	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
	1		X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)	

## Приложение В (обязательное) Регистры прерываний

Для управления прерываниями используются пять групп регистров ISERi, ICERi, ISPRi, ICPRi и IABRi, где индекс  $i = 0, 1, 2$  (см. таблицу В.1). Группы имеют идентичную структуру. Набор прерываний, которыми управляет регистр группы, зависит от индекса. На рисунке В.1 показана одна группа регистров и указано соответствие номеров векторов прерываний и бит регистров. Управление прерыванием осуществляется записью единицы или нуля в соответствующий бит. Допускается одновременное управление несколькими прерываниями.

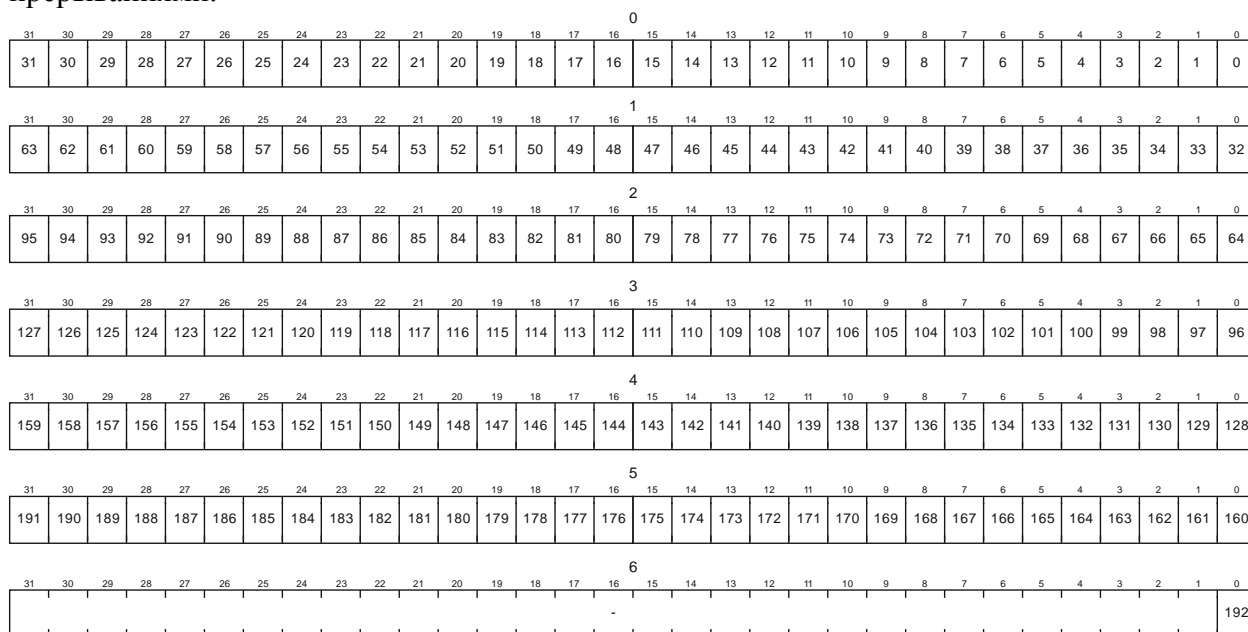


Рисунок В.1 – Соответствие векторов прерываний и бит управляющих регистров

Таблица В.1 – 32-разрядные регистры управления прерываниями

Мнемоника и назначение	Операция над битом	Влияние на соответствующее прерывание	
<b>ISERi</b> Регистр разрешения прерываний от источников	Чтение	0	Прерывание запрещено
		1	Прерывание разрешено
	Запись	0	Нет влияния
		1	Разрешение прерывания
<b>ICERi</b> Регистр сброса разрешения прерываний от источников	Чтение	0	Прерывание запрещено
		1	Прерывание разрешено
	Запись	0	Нет влияния
		1	Запрет прерывания
<b>ISPRi</b> Регистр ждущих прерываний	Чтение	0	Ждущего прерывания нет
		1	Есть ждущее прерывание
	Запись	0	Нет влияния
		1	Установка ждущего прерывания (программное прерывание)
<b>ICPRi</b> Регистр сброса ждущих прерываний	Чтение	0	Ждущего прерывания нет
		1	Есть ждущее прерывание
	Запись	0	Нет влияния
		1	Сброс ждущего прерывания
<b>IABRi</b> Регистр флагов прерываний	Чтение	0	Флага прерывания нет
		1	Флаг прерывания установлен. Сбрасывается аппаратно по окончании обслуживания прерывания

Для задания приоритетов прерываний используются регистры IPRi, где i индекс от 0 до 33. В таблице В.2 представлен формат регистра IPRi.

Таблица В.2 – Регистр приоритетов

IPRi		Сброс: 0000000h													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIn+3								PRIn+2							
3 ч								3 ч							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIn+1								PRIn+0							
3 ч								3 ч							
Поле	Бит		Описание												
PRIn+3	31-24		Поле задания приоритета (n+3)-ого вектора прерывания												
PRIn+2	23-16		Поле задания приоритета (n+2)-ого вектора прерывания												
PRIn+1	15-8		Поле задания приоритета (n+1)-ого вектора прерывания												
PRIn	7-0		Поле задания приоритета n-ого вектора прерывания $n = 4 \times i.$												
Примечание – Допустимые значения для записи в поля от 00h до 0Fh. Таким образом, каждому вектору может быть назначен один из 16 уровней приоритета. Значения 10h – FFh не используются.															

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Подпись	Дата
	изме- ненных	замененных	новых	анну- лиро- ванных				