

## Основные особенности

- Напряжение питания:  
5,0 В ± 10% – без использования  
встроенного линейного регулятора;  
8,0 В...15 В – при использовании  
встроенного линейного регулятора;
- Ток потребления не более 20 мА;
- Микроконтроллерное ядро  
архитектуры 8051:  
– машинный цикл 1 такт.
- Режим энергосбережения («SLEEP»);
- Режим энергосбережения  
с отключенными генераторами  
(«COLD\_SLEEP»);
- Температурный диапазон  
от –60°C до +125°C.

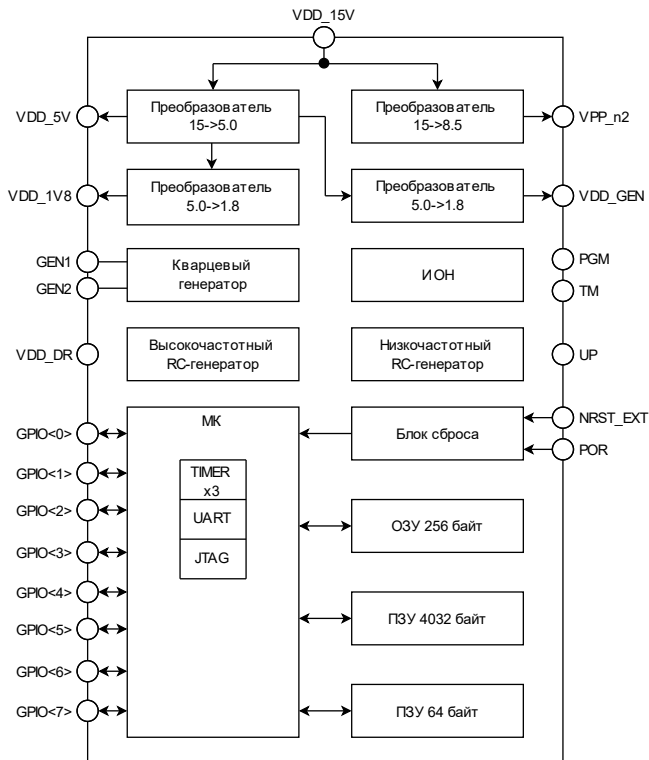


Рисунок 1. Структурная схема



ГГ – год выпуска  
НН – неделя выпуска

Рисунок 2. Внешний вид  
микросхемы 5400BK025

## Общее описание

Микросхема 5400BK025 – малопотребляющий контроллер с архитектурой ядра 8051, поддержкой энергосберегающего режима таймера и рабочим диапазоном частот до 8,0 МГц. Микроконтроллер в своем составе содержит низкочастотный RC-генератор (~12 кГц – 400 кГц) и высокочастотный RC-генератор (~5,0 МГц – 10 МГц). Тактирование осуществляется одним из трех способов: встроенный RC-генератор, кварцевый генератор с внешним резонатором или внешний источник.

Встроенные периферийные модули: JTAG, UART с настраиваемой скоростью передачи, 8 универсальных портов ввода / вывода, три 24-разрядных таймер / счетчика, сторожевой таймер. Возможен переход в энергосберегающий режим таймера («SLEEP») или энергосберегающий режим с полностью отключенными генераторами («COLD\_SLEEP»).

Память программ: 4032 Б + 64 Б ОППЗУ, память данных: 256 Б ОЗУ. Отличительная особенность микросхемы – допжог ПЗУ во время выполнения программы объемом 64 Б.

Микросхема выполнена в 28-ми выводном металлокерамическом корпусе МК 5123.28-1.01.

## Электрические параметры микросхемы

Таблица 1. Электрические характеристики (температурный диапазон от  $-60^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )

Параметр, единица измерения	Норма параметра		
	не менее	типовое	не более
Частота тактирования ядра, МГц			8,0
Выходное напряжение встроенных линейных регуляторов, В			
15 В → 5,0 В (вывод VDD_5V)	4,5	5,0	5,5
15 В → 8,5 В (вывод VPP_n2)	8,0	8,5	9,0
5,0 В → 1,8 В (вывод VDD_GEN)	1,62	1,8	1,98
5,0 В → 1,8 В (вывод VDD_1V8)	1,62	1,8	1,98
Нижняя граница диапазона настройки частоты встроенного низкочастотного RC-генератора, кГц			20
Верхняя граница диапазона настройки частоты встроенного низкочастотного RC-генератора, кГц	200		
Нижняя граница диапазона настройки частоты встроенного высокочастотного RC-генератора, МГц			5,0
Верхняя граница диапазона настройки частоты встроенного высокочастотного RC-генератора, МГц	10		
Ток потребления, мА			20
Ток потребления в режиме «SLEEP», мкА		60	100
Ток потребления в режиме «COLD_SLEEP», мкА		50	90
Напряжение высокого уровня выходных цифровых сигналов (GPIO в режиме выхода, UP), В			
при VDD_DR = 1,8 В	1,4	1,8	
при VDD_DR = 5,0 В	4,6	5,0	
Напряжение низкого уровня выходных цифровых сигналов (GPIO в режиме выхода, UP), В			
при VDD_DR = 1,8 В		0	0,4
при VDD_DR = 5,0 В		0	0,4

## Электростатическая защита

Микросхема имеет встроенную защиту от электростатического разряда до 1000 В по модели человеческого тела. Требуется мер предосторожности.

## Предельно-допустимые и предельные режимы эксплуатации

Таблица 2. Предельно-допустимые и предельные режимы эксплуатации

Параметр, единица измерения	Предельно-допустимый режим		Предельный режим	
	не менее	не более	не менее	не более
Напряжение питания (VDD_15V) <sup>(1)</sup> , В	8,0 12 <sup>(2)</sup>	16,5	-0,3	17,5
Напряжение питания (VDD_5V) <sup>(3)</sup> , В	4,5	5,5	-0,3	5,7
Напряжение питания интерфейсной части (VDD_DR), В	1,62	5,5	-0,3	5,7
Напряжение программирования ПЗУ (VPP), В	8,5	9,0	-0,3	9,5
Напряжение высокого уровня входных цифровых сигналов (GPIO в режиме входа, PGM, TM, NRST_EXT, GEN1), В	0,7× VDD_DR <sup>(5)</sup>  VDD_DR- 0,4 <sup>(6)</sup>	VDD_DR	-0,3	5,7
Напряжение низкого уровня входных цифровых сигналов (GPIO в режиме входа, PGM, TM, NRST_EXT, GEN1), В	0	0,3× VDD_DR <sup>(5)</sup>  0,4 <sup>(6)</sup>	-0,3	5,7
Нагрузочная способность цифровых выводов (GPIO в режиме выхода), мА	-	1,0 <sup>(4)</sup> 5,0	-	10
Частота внешнего тактового сигнала, МГц <sup>(7)</sup>	-	8,0	-	10
Температура эксплуатации, °С	-60	+125	-60	+150

Примечание:

- 1) В случае использования линейного регулятора 15 В → 5,0 В. Напряжение питания микросхемы подается на вывод VDD\_15V.
- 2) Для программирования 64 Б ПЗУ при исполнении программы. Используется линейный регулятор 15 В → 8,5 В. Напряжение питания микросхемы подается на вывод VDD\_15V.
- 3) В случае, когда линейный регулятор 15 В → 5,0 В не используется. Напряжение питания микросхемы подается на выходы VDD\_5V, VDD\_15V (в данном включении LDO-регулятор 15 В → 8,5 В не работоспособен).
- 4) При объединении VDD\_5V с VDD\_DR.
- 5) При напряжении питания VDD\_DR от 3,0 В до 5,5 В.
- 6) При напряжении питания VDD\_DR от 1,62 В до 3,0 В.
- 7) При подаче внешнего сигнала тактовой частоты или использовании внешнего кварцевого резонатора для тактирования микросхемы.

## Конфигурация и функциональное описание выводов

Таблица 3. Функциональное описание выводов

№ вывода	Тип вывода	Наименование вывода	Назначение вывода
1, 16	Power	VDD_DR	Вывод положительного напряжения питания универсальных портов ввода-вывода микроконтроллера
2	DI/DO	GPIO<3>	Порт ввода-вывода микроконтроллера, разряд №3 / выход TDO интерфейса JTAG в тестовом режиме (при TM = 1)
3	DI/DO	GPIO<2>	Порт ввода-вывода микроконтроллера, разряд №2 / вход TDI интерфейса JTAG в тестовом режиме (при TM = 1)
4	DI/DO	GPIO<1>	Порт ввода-вывода микроконтроллера, разряд №1 / вход TMS интерфейса JTAG в тестовом режиме (при TM = 1)
5	DI/DO	GPIO<0>	Порт ввода-вывода микроконтроллера, разряд №0 / вход TCK интерфейса JTAG в тестовом режиме (при TM = 1)
6	DI	PGM	Вход выбора режима программирования ПЗУ в тестовом режиме (при TM = 1): лог. «1» – перевод в режим программирования, от шины отключается CPU и подключается JTAG; лог. «0» – стандартная работа.
7	DI	TM	Вход для выбора режима работы микроконтроллера: лог. «1» – тестовый режим; лог. «0» – стандартная работа.
8	DI	POR	Вход для подключения внешнего конденсатора блока сброса. Встроенный резистор 180 кОм к VDD_5V.
9	DI	NRST_EXT	Вход внешнего сигнала сброса. Активный уровень – лог. «0»
10, 23	Power	VSSD	Вывод цифрового отрицательного напряжения питания, общий
11	AI	VPP	Вывод для подачи напряжения (~8,5 В) программирования ПЗУ и конфигурационной памяти
12	Power	VSSA	Вывод аналогового отрицательного напряжения питания, общий
13	AO / AI	VPP_n2	Выход регулятора напряжения (~8,5 В). Регулятор напряжения автоматически формирует напряжение для допрожига ПЗУ 64 Б
14	Power	VDD_15V	Вывод положительного напряжения питания 15 В
15, 28	AO / Power	VDD_5V	Вывод цифрового положительного напряжения питания (выходное напряжение LDO-регулятора 15 В → 5,0 В)
17	AO / Power	VDD_1V8	Вывод напряжения питания ядра (выходное напряжение LDO-регулятора 5,0 В → 1,8 В)
18	AO / Power	VDD_GEN	Вывод напряжения питания RC-генераторов (выходное напряжение LDO-регулятора 5,0 В → 1,8 В)
19	DI	GEN1	Вход подключения кварцевого резонатора / вход для подачи внешней тактовой частоты
20	DO	GEN2	Вход подключения кварцевого резонатора / выход контроля поданной внешней частоты
21	Power	VSSGEN	Отрицательное напряжение питания RC-генераторов, общий

№ вывода	Тип вывода	Наименование вывода	Назначение вывода
22	DO	UP	Сигнал разрешения подачи 8,5 В на вывод VPP_n2 при программировании ПЗУ 64 Б в режиме с внешним регулятором напряжения (при использовании встроенного регулятора вывод может использоваться для контроля момента программирования)
24	DI/DO	GPIO<7>	Порт ввода-вывода микроконтроллера, разряд №7 / вывод RCF контроля выходной частоты высокочастотного RC-генератора (при TM = 1)
25	DI/DO	GPIO<6>	Порт ввода-вывода микроконтроллера, разряд №6 / вывод RCS контроля выходной частоты низкочастотного RC-генератора (при TM = 1)
26	DI/DO	GPIO<5>	Порт ввода-вывода микроконтроллера, разряд №5 / выбор режима конфигурационной памяти (при TM = 1)
27	DI/DO	GPIO<4>	Порт ввода-вывода микроконтроллера, разряд №4
Примечание: DI – цифровой вход DO – цифровой выход AI – аналоговый вход AO – аналоговый выход Power – вывод напряжения питания			

## Рекомендуемая схема применения

Таблица 4. Таблица внешних компонентов

Компонент	Номинал
C1, C2, C6 – C8	0,1 мкФ – 1,0 мкФ
C3	1,0 нФ – 100 нФ. Выбирается пользователем в зависимости от требуемого времени первоначального сброса
C4, C5	16 пФ – 64 пФ
R1	1,0 МОм – 2,0 МОм
R2	2,0 кОм – 20 кОм
ZQ – кварцевый резонатор с частотой до 8 МГц	

Конденсаторы высокочастотные керамические, либо сдвоенные. В случае сдвоенных конденсаторов, один из них обязательно должен быть высокочастотный керамический емкостью не менее 10 нФ. Шунтирующие конденсаторы должны располагаться на плате в непосредственной близости к соответствующим выводам микросхемы.

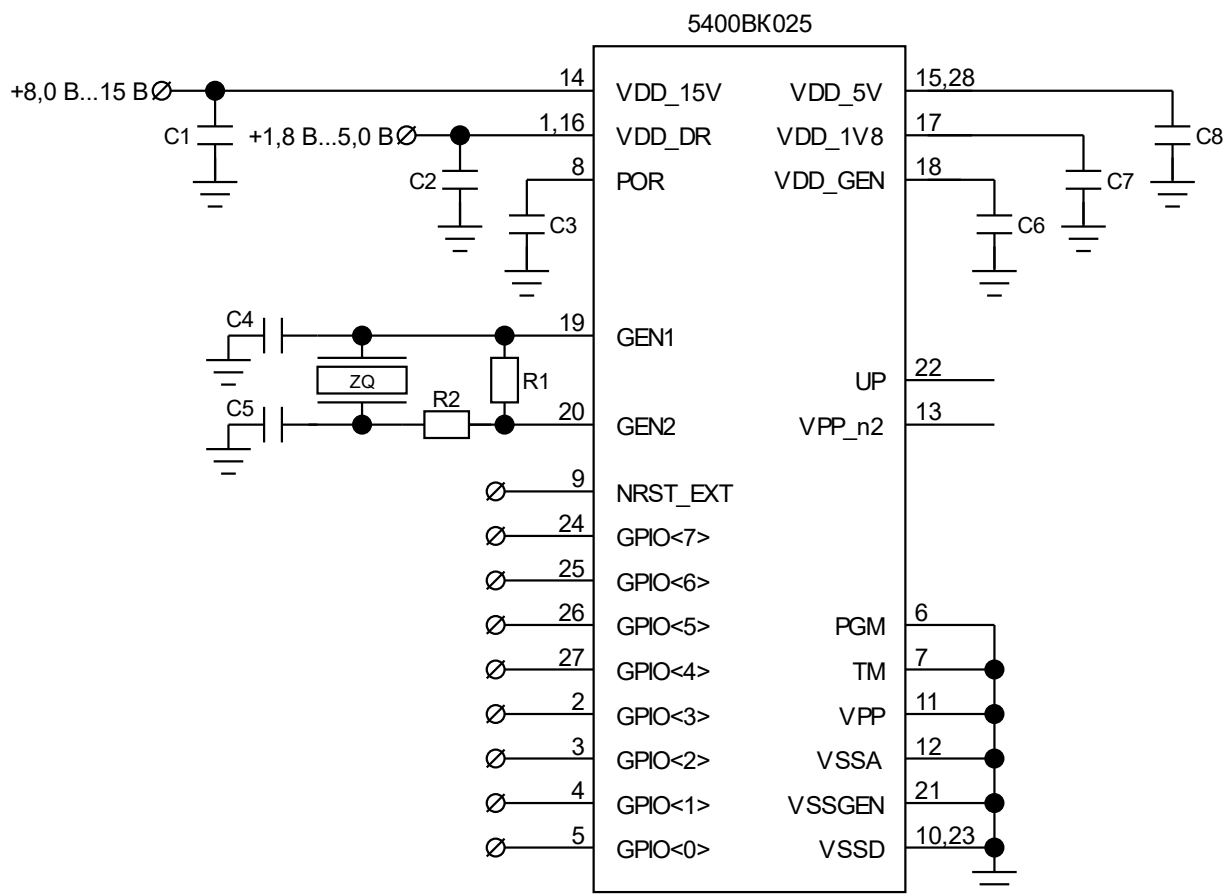


Рисунок 3. Рекомендуемая схема применения с использованием встроенных регуляторов напряжения

**Примечание:**

При использовании встроенного линейного регулятора для программирования пользовательской ПЗУ 64Б требуемое напряжение питания от 12 В до 15 В.

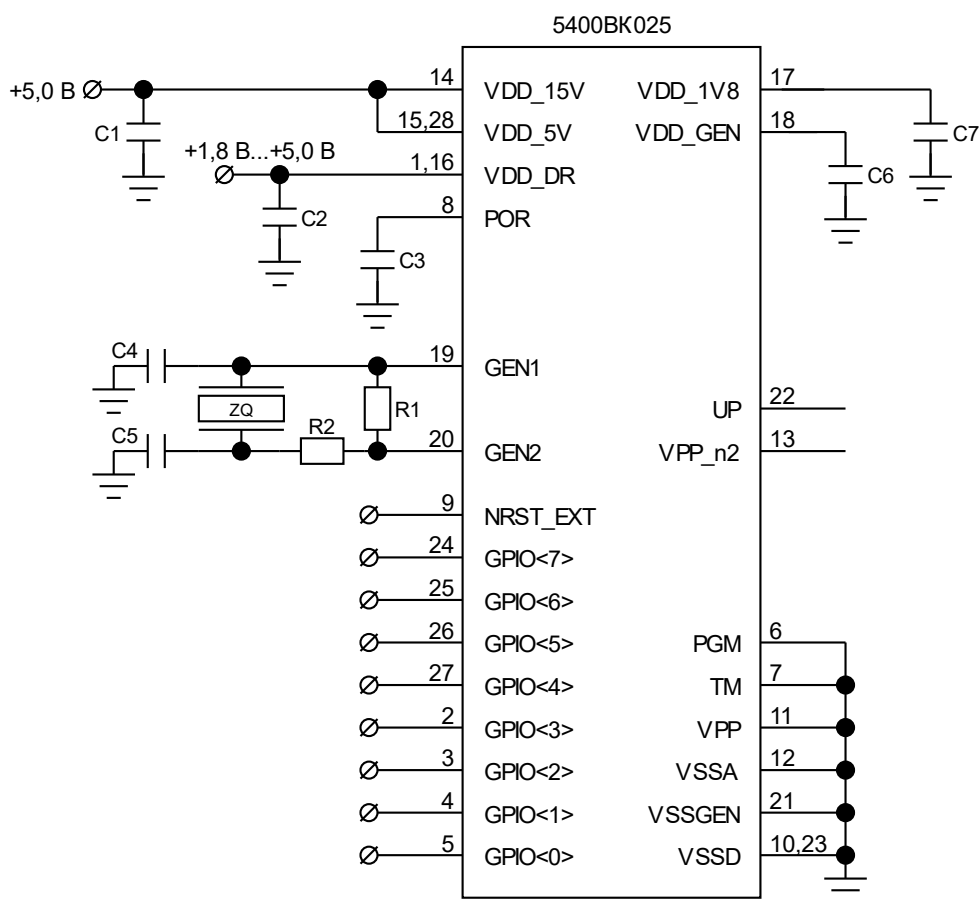


Рисунок 4. Рекомендуемая схема применения без использования встроенных регуляторов напряжения

#### Примечание:

При организации питания микроконтроллера от 5,0 В необходимо подавать напряжение 5,0 В на выводы VDD\_15V (14), VDD\_5V (15, 28).

При использовании внешнего генератора вывод GEN2 (20) необходимо оставить в обрыве, тактовая частота задается на вывод GEN1 (19).

Конденсатор C3 выбирается пользователем в зависимости от требуемого времени первоначального сброса:

Длительность сигнала первоначального сброса (POR), мс T = 25°C.	
для конденсатора 1,0 нФ	0,15 – 0,175
для конденсатора 10 нФ	1,5 – 1,75
для конденсатора 100 нФ	15 – 17,5

Расчетная длительность аналогового сигнала первоначального сброса относительно «резкого» (1,0 мкс) включения питания на выводе VDD\_5V = 5,0 В. Замедление включения питания будет соответственно затягивать сброс. После срабатывания аналогового сброса добавляется еще цифровая фильтр-задержка в течение 1000 периодов частоты, установленной в качестве системной.

## Конфигурационная память (ANALOG\_CFG)

В микросхеме реализована конфигурационная память (ANALOG\_CFG), которая может работать как в «SOFT», так и в «HARD» режиме. Регистры модуля ANALOG\_CFG подключены к конфигурационному однократно программируемому ПЗУ. В зависимости от состояния вывода ТМ (вывод №7) выводы №2, 3, 4, 5, 24, 25, 26 имеют разное назначение.

№ вывода	При ТМ = 0	При ТМ = 1
2	GPIO<3>	TDO
3	GPIO<2>	TDI
4	GPIO<1>	TMC
5	GPIO<0>	TCK
24	GPIO<7>	RC_F
25	GPIO<6>	RC_S
26	GPIO<5>	H_S

В зависимости от вывода H\_S (вывод №26 при ТМ = «1») источником данных конфигурационной памяти могут быть либо регистры (H\_S = «1», SOFT-режим), либо ПЗУ (H\_S = «0», HARD-режим).

Для прожига конфигурационной памяти необходимо в «SOFT» режиме подать 9,0 В на вывод VPP.

## Регистры модуля ANALOG\_CFG

№	Аббревиатура	Доступ	Описание
2A00h	ANALOG_O_RCF	W	Регистр настройки емкости конденсатора высокочастотного RC-генератора
2A01h	ANALOG_O_RCS	W	Регистр настройки емкости конденсатора низкочастотного RC-генератора
2A02h	ANALOG_O_RC_R	W	Регистр настройки сопротивления резистора низкочастотного RC-генератора
2A04h	ANALOG_O_GEN	W	Регистр выбора высокочастотного RC-генератора, внешнего генератора, источника для прожига ПЗУ 64Б

### ANALOG\_O\_RCF

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		RCF_C					
Начальное значение	0							

RCF\_C – настройка емкости конденсатора высокочастотного RC-генератора.

### ANALOG\_O\_RCS

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		RCS_C					
Начальное значение	0							

RCS\_C – настройка емкости конденсатора низкочастотного RC-генератора.



## ANALOG\_O\_RC\_R

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*		RC2_R			Резерв		
Начальное значение	0							

\* – Не изменять начальное значение.

RC2\_R – настройка сопротивления резистора низкочастотного RC-генератора.

Встроенный низкочастотный RC-генератор настраивается с помощью переменного конденсатора (биты RCS\_C регистра ANALOG\_O\_RCS) и переменного резистора (биты RC2\_R регистра ANALOG\_O\_RC\_R).

Таблица 5. Типовое значение частоты низкочастотного генератора.

Биты RC2_R*	Биты RCS_C**		
	111111	...	000000
000	~90 кГц	...	~420 кГц
001	~80 кГц	...	~380 кГц
010	~70 кГц	...	~330 кГц
011	~60 кГц	...	~280 кГц
100	~45 кГц	...	~225 кГц
101	~35 кГц	...	~170 кГц
110	~25 кГц	...	~115 кГц
111	~12 кГц	...	~60 кГц

\* – Биты RC2\_R регистра ANALOG\_O\_RC\_R.

\*\* – Биты RCS\_C регистра ANALOG\_O\_RCS.

Частота встроенного генератора может иметь первоначальный разброс от типового значения вследствие технологического разброса (от образца к образцу), напряжения питания и температуры. Гарантируемым диапазоном настройки низкочастотного генератора является частота от 20 кГц до 200 кГц. Точное требуемое значение выбирается подстройкой указанных выше бит.

Встроенный высокочастотный генератор настраивается только с помощью переменного конденсатора (биты RCF\_C регистра ANALOG\_O\_RCF). Однако, стоит учитывать, что частота тактирования микроконтроллера не более 8 МГц.

Таблица 6. Типовое значение частоты высокочастотного генератора.

Биты RCF_C*		
111111	...	000000
≤ 5,0 МГц	...	≥ 10 МГц

\* – Биты RCF\_C регистра ANALOG\_O\_RCF.

## ANALOG\_O\_GEN

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*					LDO	A_RCX	A_RC
Начальное значение	0							

\* – Не изменять начальное значение.

**LDO** – выбор источника для прожига ПЗУ 64Б:

- 1 – внешнее напряжение для прожига;
- 0 – встроенный линейный регулятор.

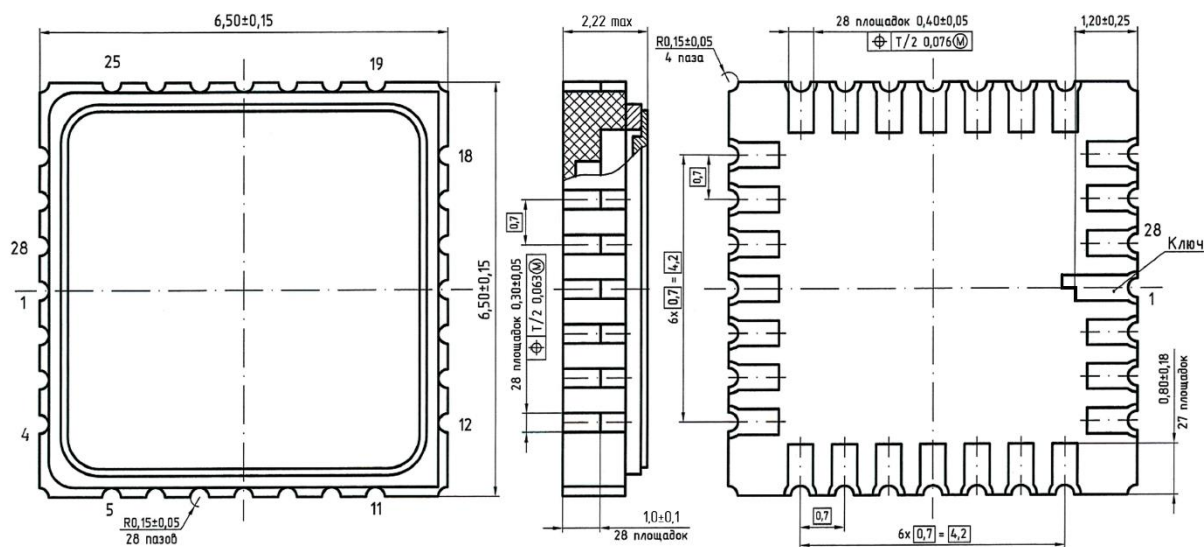
**A\_RCX** – выбор внешнего генератора:

- 1 – внешний генератор;
- 0 – кварцевый генератор с внешним резонатором.

**A\_RC** – выбор высокочастотного генератора:

- 1 – внешний генератор;
- 0 – встроенный генератор.

## Габаритный чертеж



1. \* Размеры для справок.  
2. Нумерация выводных площадок показана условно.

Рисунок 5. Габаритный чертеж корпуса МК 5123.28-1.01 (размеры в мм)

## Информация для заказа

Обозначение	Маркировка	Корпус	Температурный диапазон
5400BK025 АЕНВ.431290.610ТУ	BK025	МК 5123.28-1.01	-60°C...+125°C
К5400BK025 КФЦС.431000.001ТУ карта заказа КФЦС 431290.002.01СП	KBK025	МК 5123.28-1.01	-60°C...+125°C

Микросхемы категории качества «ВП» маркируются ромбом.

Микросхемы категории качества «ОТК» маркируются буквой «К».

## Оглавление

Основные особенности.....	<b>Ошибка! Закладка не определена.</b>
Общее описание .....	1
Электрические параметры микросхемы.....	2
Электростатическая защита.....	2
Предельно-допустимые и предельные режимы эксплуатации .....	3
Конфигурация и функциональное описание выводов.....	4
Рекомендуемая схема применения .....	6
Конфигурационная память (ANALOG_CFG) .....	8
Регистры модуля ANALOG_CFG.....	8
ANALOG_O_RCF .....	8
ANALOG_O_RCS .....	8
ANALOG_O_RC_R .....	9
ANALOG_O_GEN .....	10
Габаритный чертеж.....	<b>Ошибка! Закладка не определена.</b>
Информация для заказа .....	<b>Ошибка! Закладка не определена.</b>
Техническое описание .....	17
Блок схема.....	17
Состав изделия .....	18
Альтернативные функции .....	<b>Ошибка! Закладка не определена.</b>
Карта памяти.....	20
Процессор 8051 .....	21
Общие сведения .....	21
Поддержка IDE .....	21
Скорость выполнения команд .....	21
Прерывания.....	23
Подприоритеты прерываний.....	23
Регистры управления прерываниями .....	24
IE .....	24
IP .....	24
SCON .....	24
Ручной вызов прерываний .....	25
P1 .....	25
Таймер-счетчик .....	25
Типы статусов и прерывания .....	26
Система управления сбросом и питанием (PMM) .....	27
Структурная схема.....	27
Формирование сброса.....	27
Регистры модуля системы управления сбросом и питанием.....	28

PMM_CTRL .....	28
Модуль управления источниками тактовых сигналов системы (CMM) .....	29
Общая информация .....	29
Структурная схема .....	29
Выбор источника тактирования системы .....	30
Деление частоты тактирования .....	30
Отключение частоты тактирования .....	30
Выключение источника тактирования во время режима «Глубокий сон» .....	30
Регистры модуля управления источниками тактовых сигналов системы .....	30
CMM_CTRL .....	31
CMM_DIV .....	31
CMM_MSK .....	31
CMM_ST .....	32
Сторожевой таймер (WDT) .....	33
Общая информация .....	33
Регистры сторожевого таймера .....	33
WDT_LOAD0 .....	33
WDT_LOAD1 .....	33
WDT_LOAD2 .....	34
WDT_LOAD3 .....	34
WDT_VAL0 .....	34
WDT_VAL1 .....	34
WDT_VAL2 .....	34
WDT_VAL3 .....	34
WDT_CTRL .....	34
WDT_CLR .....	35
WDT_INTRAW .....	35
WDT_INT .....	35
WDT_LOCK .....	35
WDT_TCR .....	36
WDT_TOP .....	36
GPIO .....	37
Общая информация .....	37
Структурная схема .....	37
Статусы и прерывания .....	38
Регистры GPIO .....	38
GPIO_DIR_SET / GPIO_DIR_CLR .....	39
GPIO_INTEN_SET / GPIO_INTEN_CLR .....	39
GPIO_INTTYPE_SET / GPIO_INTTYPE_CLR .....	40

GPIO_INTPOL_SET / GPIO_INTPOL_CLR .....	40
GPIO_INT .....	40
UART .....	41
Общая информация .....	41
Структурная схема .....	41
Делитель частоты .....	42
Высокоскоростной режим .....	42
Приемник .....	43
Буфер приемника .....	43
Передатчик .....	44
Буфер передатчика .....	44
Прерывания .....	44
Таймер тайм-аута .....	44
Генерация и распознавание сигнала break .....	44
Особые режимы работы .....	45
Аппаратный контроль обмена .....	46
Регистры UART .....	46
UART_CFG0 .....	47
UART_CFG1 .....	47
UART_BDR0 .....	48
UART_BDR1 .....	48
UART_TXFIFOLVL0 .....	48
UART_TXFIFOLVL1 .....	48
UART_RXFIFOLVL0 .....	49
UART_RXFIFOLVL1 .....	49
UART_NBMSK .....	49
UART_NBADDR0 .....	49
UART_NBADDR1 .....	49
UART_MSK0 .....	50
UART_MSK1 .....	50
UART_CTRL .....	50
UART_TX .....	51
UART_RX0 .....	51
UART_RX1 .....	51
UART_ST0 .....	52
UART_ST1 .....	53
Постоянная память данных .....	54
Общая информация .....	54
Алгоритм записи данных .....	54

Алгоритм чтения данных.....	54
Регистры модуля управления OTP ПЗУ 64Б.....	54
OTP_ROM_DATA .....	54
OTP_PRESCCL .....	55
OTP_PRESCM.....	55
OTP_PRESCH .....	55
OTP_PREL.....	55
OTP_PREM.....	55
OTP_PREH .....	55
OTP_POSTL.....	56
OTP_POSTM.....	56
OTP_POSTH.....	56
Рабочий автомат (WORK_FSM).....	57
Общая информация .....	57
Особенности работы .....	57
Регистры «Рабочего автомата» .....	57
FSM_CTRL.....	57
FSM_PRDH.....	58
FSM_PRDM.....	58
FSM_PRDL.....	58
Таймер .....	59
Общая информация .....	59
Структурная схема.....	59
Работа в режиме «Простой таймер» .....	59
Принцип работы .....	59
Статусы и прерывания .....	60
Алгоритм работы .....	60
Работа в режиме «Таймер с внешней остановкой» .....	60
Принцип работы .....	60
Статусы и прерывания .....	60
Алгоритм работы .....	60
Работа в режиме «Межсобытийный таймер» .....	61
Принцип работы .....	61
Статусы и прерывания .....	61
Алгоритм работы .....	61
Работа в режиме «Таймер-счетчик» .....	61
Принцип работы .....	61
Статусы и прерывания .....	62
Алгоритм работы .....	62

Регистры таймеров .....	63
TMRx_CTRL.....	63
TMRx_CFG.....	64
TMRx_PRDH.....	64
TMRx_PRDL.....	65
TMRx_VALH.....	65
TMRx_VALM.....	65
TMRx_VALL.....	65
TMRx_MSK.....	65
TMRx_ST.....	66
Контроллер прерываний (INT_CTRL) .....	67
Регистры «Контроллера прерываний» .....	67
INT_FIX_CLR0.....	67
INT_FIX_CLR1.....	67
INT_FIX_CLR2.....	68
INT_FIX_CLR3.....	68
Режим «SLEEP» .....	69
Общая информация .....	69
Сон процессора .....	69
Глубокий сон .....	69
Режим «COLD_SLEEP» .....	69
Общая информация .....	69
Холодный сон.....	69
JTAG.....	70
Общее описание .....	70
Структурная схема.....	70
Инструкции «TEST_MODE», «EXTEST», «BYPASS», «SAMPLE/PRELOAD» .....	71
Инструкция «JTAG_RAM_ACCESS» .....	71
Инструкция «JTAG_ROM_ACCESS».....	72
Инструкция «IF_ACCESS» .....	73
Инструкция «DEBUGGER».....	73
Лист регистрации изменений .....	74



Техническое описание

Блок схема

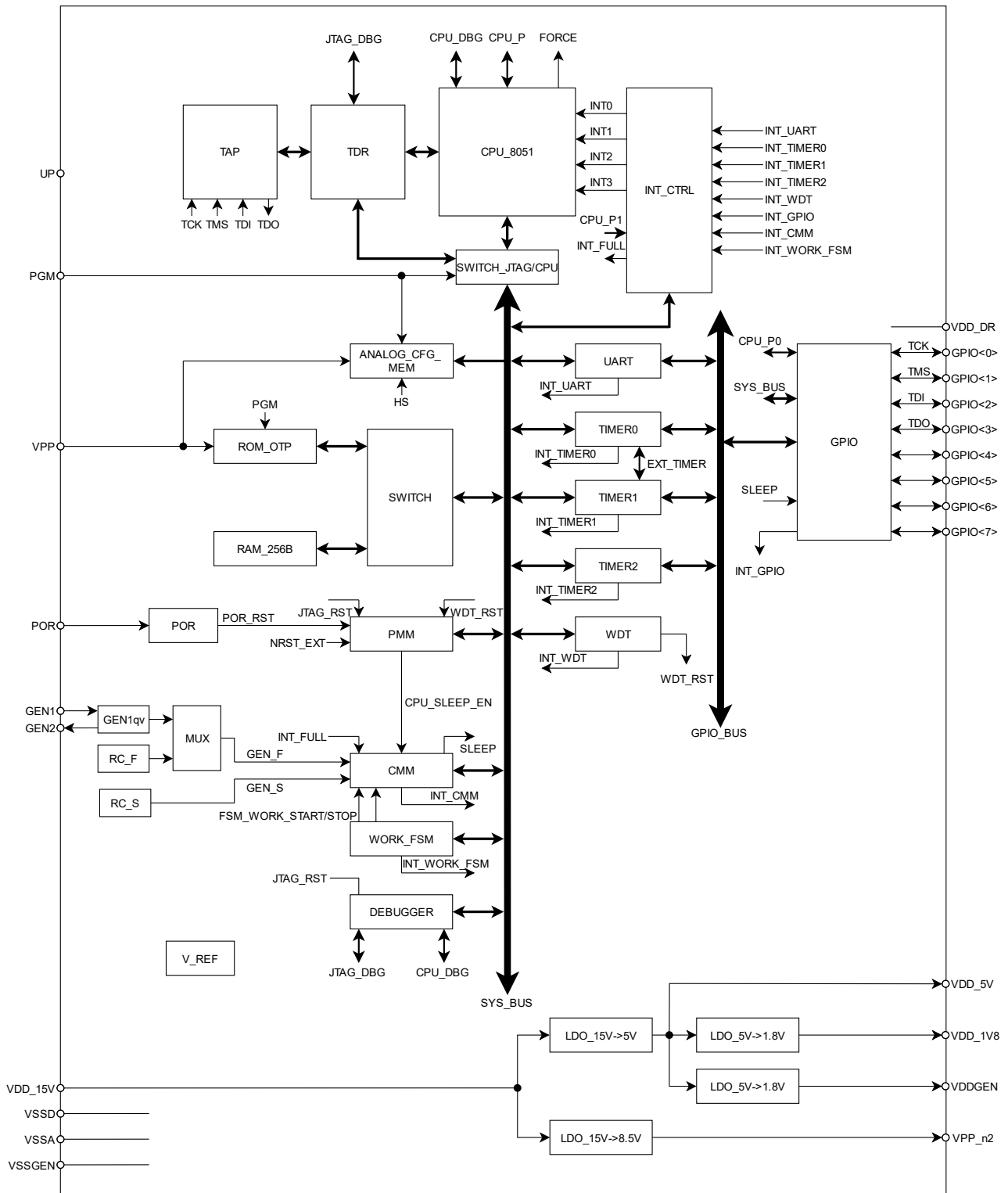


Рисунок 6. Блок схема 5400BK025

**Состав изделия**

Системные устройства:

- CPU\_8051 – процессорное ядро;
- TAP (Test Access Port) – автомат JTAG;
- TDR (Test Data Register) – регистры данных JTAG;
- MUX – мультиплексор;
- ANALOG\_CFG\_MEM – конфигурационная память;
- ROM\_OTP (One-Time Programmable) – однократно программируемое постоянное запоминающее устройство емкостью 4032 Б, память программ устройства;
- RAM\_256B – оперативное запоминающее устройство для хранения промежуточных данных, память данных устройства;
- SWITCH – модуль, разграничивающий доступ к ROM\_OTP\_4B, RAM;
- SWITCH JTAG/CPU – модуль, разграничивающий доступ к шине для JTAG, либо для CPU;
- PMM (Power Management Module) – модуль управления сбросом системы;
- POR (Power-On Reset) – модуль сброса при включении питания;
- RC\_F– высокочастотный RC-генератор;
- RC\_S – низкочастотный RC-генератор;
- CMM (Clock Management Module) – модуль регистров системы управления тактовой частоты устройства;
- DEBUGGER – отладчик;
- V\_REF – источник опорного напряжения (ИОН);
- GEN1qv – генератор на основе кварцевого резонатора;
- LDO\_15V → 5V, LDO\_5V → 1V8, LDO\_15V → 8.5V – линейные регуляторы напряжения.

Периферийные устройства:

- UART – контроллер интерфейса UART;
- GPIO – контроллер портов ввода-вывода;
- TIMER0, TIMER1, TIMER2 – универсальные таймеры;
- WDT – сторожевой таймер;
- WORK\_FSM – рабочий автомат;
- INT\_CTRL – контроллер прерываний.

## Альтернативные функции

№ вывода	Наименование вывода	Альтернативная функция	Пояснение
5	GPIO<0>	I_TIMER0_EXT	Порт ввода-вывода микроконтроллера, разряд №0: • TIMER0 – I_TIMER0_EXT (вход).
4	GPIO<1>	I_TIMER1_EXT	Порт ввода-вывода микроконтроллера, разряд №1: • TIMER1 – I_TIMER1_EXT (вход).
3	GPIO<2>	I_TIMER2_EXT	Порт ввода-вывода микроконтроллера, разряд №2: • TIMER2 – I_TIMER2_EXT (вход).
2	GPIO<3>	O_SLEEP	Порт ввода-вывода микроконтроллера, разряд №3: • режим «Глубокий сон» – O_SLEEP (выход).
27	GPIO<4>	UART_TX	Порт ввода-вывода микроконтроллера, разряд №4: • UART – TX (выход).
26	GPIO<5>	UART_RX	Порт ввода-вывода микроконтроллера, разряд №5: • UART – RX (вход).
25	GPIO<6>	CTS_n_UART	Порт ввода-вывода микроконтроллера, разряд №6: • UART – CTS (вход).
24	GPIO<7>	RTS_n_UART	Порт ввода-вывода микроконтроллера, разряд №7: • UART – RTS (выход).

Для сокращения выводов порты JTAG (TCK, TDI, TMS, TDO) мультиплексируются с портами GPIO. Выбор осуществляется выводом TM.

GPIO (при TM = 0)	JTAG (при TM = 1)
GPIO<0>	TCK
GPIO<1>	TMS
GPIO<2>	TDI
GPIO<3>	TDO

## Карта памяти

Адресное пространство памяти программ и данных разделено. В данное адресное пространство отображаются различные модули памяти и периферии.

Таблица 7. Адресное пространство памяти программ и данных

№	Выделенное пространство	Зона	Аббревиатура	Описание
0000h	4 КБ	ПЗУ	ROM	ПЗУ
0000h	256 Б	ОЗУ	RAM	ОЗУ
2000h	256 Б		PMM	Подсистема управления электропитанием
2100h	32 Б		CMM	Подсистема управления тактированием
2200h	256 Б		WDT	Сторожевой таймер
2300h	256 Б		GPIO	Универсальный порт ввода-вывода
2400h	256 Б		UART	Интерфейс UART
2500h	256 Б		ROM 64 Б	Постоянная память данных, 64 Б
2600h	256 Б		WORK_FSM	Рабочий автомат
2700h	32 Б		TIMER0	Таймеры
2720h	32 Б		TIMER1	
2740h	32 Б		TIMER2	
2800h	256 Б		INT_CTRL	Контроллер прерываний
2900h	64 Б		DEBUGGER	Отладчик
2A00h	8 Б		ANALOG_CFG	Конфигурационная память аналоговых блоков

Если CPU обратится по адресу вне выделенного пространства, то при чтении будут приняты нулевые данные, запись не будет иметь никакого эффекта.

## Процессор 8051

### Общие сведения

Текущая версия процессора 8051 соответствует стандартному ядру, за исключением некоторых особенностей, которые будут описаны в данной главе.

### Поддержка IDE

IDE – Keil. Потактовая симуляционная модель – Atmel AT89LP52.

### Скорость выполнения команд

Машинный цикл равен 1 такту. Уход на обработчика прерывания занимает 5 машинных циклов.

Таблица 8. Скорость выполнения команд

Команда	Кол-во машинных циклов	Команда	Кол-во машинных циклов
ACALL	3	MOV_ATRI_DIRECT	2
ADD_A_ATRI	2	MOV_ATRI_IMMEDIATE	2
ADD_A_DIRECT	2	MOV_BIT_C	2
ADD_A_IMMEDIATE	2	MOV_C_BIT	2
ADD_A_RN	1	MOV_DIRECT_A	2
ADDC_A_ATRI	2	MOV_DIRECT_ATRI	2
ADDC_A_DIRECT	2	MOV_DIRECT_DIRECT	3
ADDC_A_IMMEDIATE	2	MOV_DIRECT_IMMEDIATE	3
ADDC_A_RN	1	MOV_DIRECT_RN	2
AJMP	3	MOV_DPTR_IMMEDIATE:	3
ANL_A_ATRI	2	MOV_RN_A	1
ANL_A_DIRECT	2	MOV_RN_DIRECT	2
ANL_A_IMMEDIATE	2	MOV_RN_IMMEDIATE	2
ANL_A_RN	1	MOVC_A_ATDPTR	3
ANL_C_BIT	2	MOVC_A_ATPC	3
ANL_C_NBIT	2	MOVX_A_ATDPTR	3
ANL_DIRECT_A	2	MOVX_A_ATRI	3
ANL_DIRECT_IMMEDIATE	3	MOVX_ATDPTR_A	3
CJNE_A_DIRECT	4	MOVX_ATRI_A	3
CJNE_A_IMMEDIATE	4	MUL_AB	2
CJNE_ATRI_IMMEDIATE	4	NOP	1
CJNE_RN_IMMEDIATE	4	ORL_A_ATRI	2
CLR_A	1	ORL_A_DIRECT	2
CLR_BIT	2	ORL_A_IMMEDIATE	2
CLR_C	1	ORL_A_RN	1
CPL_A	1	ORL_C_BIT	2
CPL_BIT	2	ORL_C_NBIT	2
CPL_C	1	ORL_DIRECT_A	2
DA_A	1	ORL_DIRECT_IMMEDIATE	3
DEC_A	1	POP	2

Команда	Кол-во машинных циклов	Команда	Кол-во машинных циклов
DEC_ATRI	2	PUSH	2
DEC_DIRECT	2	RET	4
DEC_RN	1	RETI	4
DIV_AB	4	RL_A	2
DJNZ_DIRECT	4	RLC_A	1
DJNZ_RN	3	RR_A	1
INC_A	1	RRC_A	1
INC_ATRI	2	SETB_BIT	2
INC_DIRECT	2	SETB_C	1
INC_DPTR	2	SJMP	3
INC_RN	1	SUBB_A_ATRI	2
JB	4	SUBB_A_DIRECT	2
JBC	4	SUBB_A_IMMEDIATE	2
JC	3	SUBB_A_RN	1
JMP_A_DPTR	2	SWAP_A	1
JNB	4	XCH_A_ATRI:	2
JNC	3	XCH_A_DIRECT	2
JNZ	3	XCH_A_RN	1
JZ	3	XCHD_A_ATRI	2
LCALL	4	XRL_A_ATRI	2
LJMP	4	XRL_A_DIRECT	2
MOV_A_ATRI	2	XRL_A_IMMEDIATE	2
MOV_A_DIRECT	2	XRL_A_RN	1
MOV_A_IMMEDIATE	3	XRL_DIRECT_A	2
MOV_A_RN	1	XRL_DIRECT_IMMEDIATE	3
MOV_ATRI_A	1		

**Прерывания**

В текущей версии процессора 8051 предусмотрено 6 векторов прерываний.

Таблица 9. Описание прерываний и адрес перехода

Номер прерывания	Описание	Адрес перехода	Внутреннее название прерываний
0	WORK_FSM	0003h	IE0
	WDT		
	<i>Резерв</i>		
	CMM		
1	Внутренний таймер / счетчик T0	000Bh	TF0
2	GPIO	0013h	IE1
	TIMER0		
	UART		
3	Внутренний таймер / счетчик T1	001Bh	TF1
4	TIMER1	0023h	IS0
5	TIMER2	002Bh	IS1

Прерывания IS0 и IS1 всегда работают по уровню.

**Подприоритеты прерываний**

При одинаковом приоритете обработка прерываний выполняется в следующей последовательности

Таблица 10. Описание подприоритетов прерываний

Подприоритет	Название прерывания
1 – высший	IE0
2	TF0
3	IS0
4	IE1
5	TF1
6 – низший	IS1

**Регистры управления прерываниями**

При добавлении новых прерываний, в регистре приоритетов прерываний и регистре разрешения прерываний, были внесены изменения относительно стандартного ядра 8051.

**IE**

Измененный регистр разрешения прерываний представлен ниже.

Бит	7	6	5	4	3	2	1	0
Назначение	EA	Резерв	ES1	ES0	ET1	EX1	ET0	EX0
Начальное значение	0							

**EA** – включение системы прерываний.

Изменение заключается в добавлении битов ES0, ES1, отвечающих за разрешение прерываний IS0, IS1 (биты ET0, ET1, EX0, EX1 отвечают за разрешение прерываний TF0, TF1, IE0, IE1 соответственно).

**IP**

Измененный регистр приоритетов прерываний представлен ниже.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		PS1	PS0	PT1	PX1	PT0	PX0
Начальное значение	0							

Изменение заключается в добавлении битов PS0, PS1, отвечающих за установку приоритетов прерываний IS0, IS1. (биты PT0, PT1, PX0, PX1 отвечают за установку приоритетов прерываний TF0, TF1, IE0, IE1 соответственно).

**SCON**

Измененный регистр SCON представлен ниже.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв						IS1	IS0
Начальное значение	0							

Изменение заключается в удалении битов, отвечающих за управление работой последовательного порта и добавление флагов прихода прерываний IS0, IS1.



**Ручной вызов прерываний**

Пользователь может самостоятельно вызвать прерывания IE0, IE1, IS0, IS1. Вызвать прерывания можно через регистры управления портом P1.

**P1**

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>			EN_N	IS1	IS0	IE1	IE0
Начальное значение	255							

**EN\_N** – разрешение прерывания на основе битов текущего регистра:

1 – прерывания будут вызваны в соответствии с таблицей прерываний;

0 – разрешения вызова прерываний битами текущего регистра.

При разрешении вызова прерываний битами текущего регистра, прерывания от периферийных модулей будут отключены от входных линий прерываний IE0, IE1, IS0, IS1.

**IS1** – прерывание IS1:

1 – передать на вход линии прерывания IS1 высокий логический уровень;

0 – передать на вход линии прерывания IS1 низкий логический уровень.

**IS0** – прерывание IS0:

1 – передать на вход линии прерывания IS0 высокий логический уровень;

0 – передать на вход линии прерывания IS0 низкий логический уровень.

**IE1** – прерывание IE1:

1 – передать на вход линии прерывания IE1 высокий логический уровень;

0 – передать на вход линии прерывания IE1 низкий логический уровень.

**IE0** – прерывание IE0:

1 – передать на вход линии прерывания IE0 высокий логический уровень;

0 – передать на вход линии прерывания IE0 низкий логический уровень.

**Таймер-счетчик**

В текущей версии внешнее событие для таймера-счетчика всегда равно нулю.

В стандартной реализации таймера-счетчик инкрементируется каждый машинный цикл. В данном случае машинный цикл равен 1 такту системной частоты.

## Типы статусов и прерывания

Биты в регистрах статусов периферийных блоков подразделяются на флаги (FLAG) и события (EVENT). Разница между ними в том, что флаги сбрасываются, когда указанное условие перестает выполняться, а биты, соответствующие событиям, сбрасываются по чтению. При этом фиксация события имеет приоритет над сбросом.

Прерывание по биту-флагу формируется в момент перехода бита статуса из «0» в «1» (но не наоборот). Если бит статуса уже находился в «1», и в этот момент разрешается прерывание по этому биту, то прерывание тут же сработает.

Прерывание по биту-событию формируется в момент возникновения данного события. Следует иметь в виду, что бит в регистре статуса установится в «1» от первого события, и, если его не сбросить чтением, то он останется в «1». Прерывания же формируются и от последующих событий и неважно в «0» или в «1» статусный бит.

## Система управления сбросом и питанием (PMM)

### Структурная схема

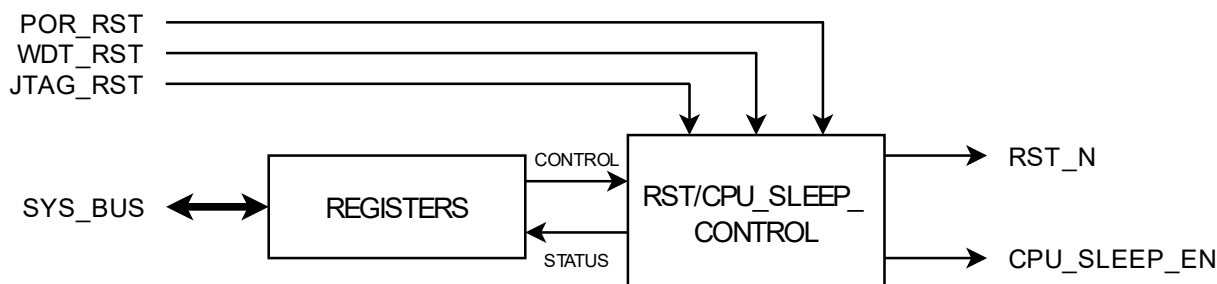


Рисунок 7. Структурная схема системы управления сбросом и питанием

Система управления сбросом и питанием состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- RST/CPU\_SLEEP\_CONTROL – блок управления сбросом и сном процессора, формирует системный сброс и разрешение на переход в режим «Сон процессора».

### Формирование сброса

Выходы POR и PGM имеют активный уровень «0». Вывод PGM работает, только когда вывод TM находится в «0». RST/CPU\_SLEEP\_CONTROL формирует синхронный сигнал сброса системы RST\_N из внешнего асинхронного сигнала POR\_RST. Микроконтроллер может быть сброшен программными методами от сторожевого таймера (сигнал WDT\_RST) или через JTAG (сигнал JTAG\_RST). Следует иметь введу, что JTAG и DEBUGGER сбрасывается только по сигналу POR\_RST. В модуле CMM регистры, связанные с выбором и настройкой текущей частоты сбрасываются только по POR\_RST, однако регистры CMM\_ST, CMM\_MSK и регистры, связанные с переходами в какой-либо режим «SLEEP», сбрасываются еще и по WDT\_RST и JTAG\_RST. Таким образом, после сброса от сторожевого таймера или JTAG система продолжит работу на прежней частоте или продолжит переход на другую частоту, а если находится в режиме «SLEEP», то выйдет из него.

Также блок RST/CPU\_SLEEP\_CONTROL на основании данных блока REGISTERS формирует сигнал разрешения перехода в режим «Сон процессора» для модуля CMM (сигнал CPU\_SLEEP\_EN).

## Регистры модуля системы управления сбросом и питанием

№	Аббревиатура	Доступ	Описание
2000h	PMM_CTRL	W	Регистр управления
2001h	PMM_ST	R	Регистр статусов

## PMM\_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>						COLD_SLEEP	CPU_SLEEP_EN
Начальное значение	0							

**CPU\_SLEEP\_EN** – переход в режим «Сон процессора»;

1 – перейти в режим «Сон процессора»;

0 – режим «Сон процессора» выключен.

**COLD\_SLEEP** – переход в режим «Холодный сон»

1 – перейти в режим «Холодный сон»;

0 – режим «Холодный сон» выключен.

## PMM\_ST

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>						JTAG_RST	WDT_RST
Тип статуса							EVENT	
Начальное значение	0							

**JTAG\_RST** – зафиксирован сброс по JTAG.

**WDT\_RST** – зафиксирован сброс по сторожевому таймеру.

## Модуль управления источниками тактовых сигналов системы (СММ)

### Общая информация

Модуль управления источника тактовых сигналов системы предназначен для включения, отключения источников тактирования системы, переключения между ними, а также для изменения периода синхросигнала.

Микроконтроллер может работать от двух различных источников тактирования:

- Низкочастотный (встроенный низкочастотный RC-генератор);
- Высокочастотный (встроенный высокочастотный RC-генератор, кварцевый резонатор, внешний генератор).

### Структурная схема

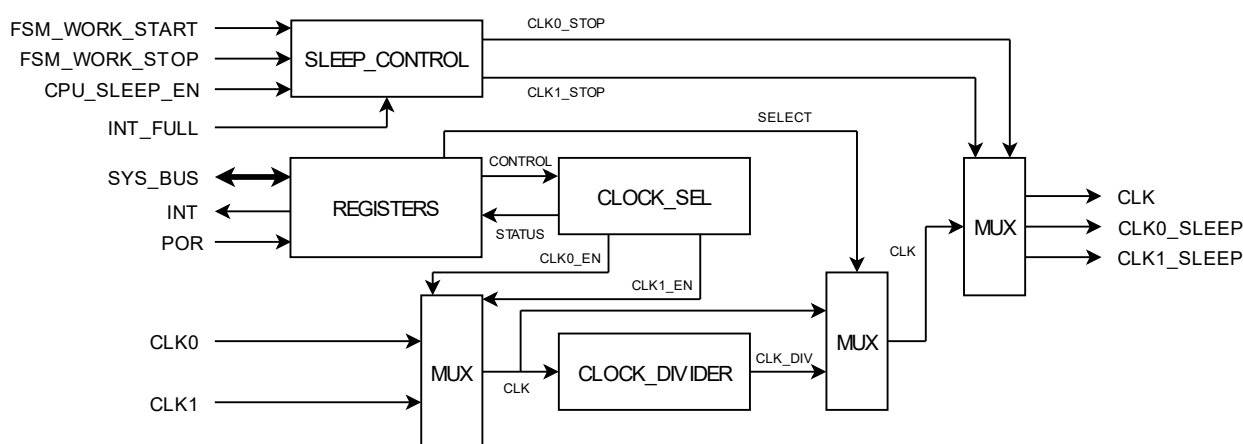


Рисунок 8. Структурная схема модуля управления источниками тактовых сигналов системы

Модуль управления источниками тактовых сигналов системы состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- SLEEP\_CONTROL – блок управления режимом «SLEEP», предназначен для запуска режимов «Сон процессора» и «Глубокий сон»;
- CLOCK\_SEL – блок выбора частоты тактирования, предназначен для выбора источника частоты тактирования системы;
- MUX – мультиплексор, предназначен для управления частотами, которые будут переданы в систему;
- CLOCK\_DIVIDER – делитель частоты, предназначен для деления частоты выбранного источника тактирования.

### Выбор источника тактирования системы

При подаче питания тактирование системы осуществляется с внутреннего высокочастотного RC-генератора. Пользователь может изменить источник тактирования путём записи необходимого значения бита RC\_FAST\_SLOW в регистре CMM\_CTRL. Переключение возможно только, если в регистре CMM\_CTRL установлены биты RC\_FAST\_EN и RC\_SLOW\_EN. В любом другом случае попытка переключения будет проигнорирована, биты не изменят своего значения. На время операции переключения в регистре CMM\_ST в активном уровне будет находиться бит SWITCH. Когда операция переключения будет завершена, бит SWITCH примет значение 0.

Процесс переключения частот состоит из нескольких этапов. После записи RC\_FAST\_SLOW команда старта синхронизируется на переключаемую частоту – 2 такта (частоты, на которую происходит переключение). Производится проверка частоты, на которую переключается пользователь – 4 такта (частоты, на которую происходит переключение). Сигнал подтверждения наличия частоты синхронизируется на текущую частоту – 2 такта (текущей частоты). Синхронизация сигнала об отключении текущей частоты – 2 такта (частоты, на которую происходит переключение). Затем снимается команда переключения – 1 такт. Вырабатывается прерывание – 1 такт. Обновляется регистр статуса – 1 такт. При большой разнице частот, процесс переключения на медленную частоту займет порядка 12 тактов медленной частоты.

### Деление частоты тактирования

Текущая частота тактирования системы может быть поделена. Для этого необходимо записать коэффициент в регистр CMM\_DIV.

### Отключение частоты тактирования

Пользователь имеет возможность отключить незадействованный источник тактирования путём записи в биты RC\_FAST\_EN или RC\_SLOW\_EN регистра CMM\_CTRL нулевого значения, таким образом понизив электропотребление системы. Отключение текущей рабочей частоты невозможно. Отключение частоты невозможно во время процедуры переключения источника тактирования.

### Выключение источника тактирования во время режима «Глубокий сон»

Модуль «CMM» имеет два бита настройки отключения источника тактирования во время режима «Глубокий сон». Подробно режим описан в пункте «Режим SLEEP». Запись «1» в один из битов RC\_F\_SLEEP\_STOP или RC\_S\_SLEEP\_STOP регистра CMM\_CTRL означает, что при переходе системы в режим «Глубокий сон», соответствующий источник тактирования автоматически будет отключен. При выходе из режима «Глубокий сон» источники тактирования будут работать в соответствии с битами RC\_SLOW\_EN и RC\_FAST\_EN. Данная процедура снижает энергопотребление системы во время режима «Глубокий сон».

Отключение источника тактирования, на котором происходит работа во время режима «Глубокий сон», приведет к полной остановке системы, без возможности программного восстановления работоспособности.

### Регистры модуля управления источниками тактовых сигналов системы

№	Аббревиатура	Доступ	Описание
2100h	CMM_CTRL	RW	Регистр управления
2101h	CMM_DIV	RW	Регистр значения делителя частоты
2102h	CMM_MSK	RW	Регистр маски
2103h	CMM_ST	R	Регистр статуса

## CMM\_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*			RC_S_SLEEP_STOP	RC_F_SLEEP_STOP	RC_SLOW_EN	RC_FAST_EN	RC_FAST_SLOW
Начальное значение	0			0	0	1	1	0

\* – Не изменять начальное значение.

**RC\_S\_SLEEP\_STOP** – выключение низкочастотного RC-генератора в режиме «Глубокий сон»:

1 – отключить низкочастотный RC-генератор;

0 – не отключать низкочастотный RC-генератор.

**RC\_F\_SLEEP\_STOP** – выключение высокочастотного RC-генератора в режиме «Глубокий сон»:

1 – отключить высокочастотный RC-генератор;

0 – не отключать высокочастотный RC-генератор.

**RC\_SLOW\_EN** – бит управления включением низкочастотного RC-генератора:

1 – включен;

0 – выключен.

**RC\_FAST\_EN** – бит управления включением высокочастотного RC-генератора:

1 – включен;

0 – выключен.

**RC\_FAST\_SLOW** – бит управления источником тактирования системы ( $F_{src}$ ):

1 – тактирование от низкочастотного RC-генератора;

0 – тактирование от высокочастотного RC-генератора.

## CMM\_DIV

Бит	7	6	5	4	3	2	1	0
Назначение	COEF							
Начальное значение	0							

**COEF** – значение регистра CMM\_DIV.

При  $COEF > 0$  частота тактирования системы рассчитывается по формуле:

$$f_{sys} = \frac{f_{src}}{2 * COEF}$$

где  $f_{src}$  – исходная частота RC-генератора.

При  $COEF = 0$ :

$$f_{sys} = f_{src}$$

## CMM\_MSK

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							NSWITCH
Начальное значение	0							

**NSWITCH** – разрешить прерывание по статусу NSWITCH:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

Прерывания работают независимо от того, включен генератор или нет.

## СММ\_ST

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>						SWITCH	NSWITCH
Начальное значение	0						0	0

**SWITCH** – переключение между частотой высокочастотного и низкочастотного RC-генератора:

1 – идет переключение на выбранную частоту;

0 – система работает на выбранной частоте.

**NSWITCH** – окончание процесса переключения частоты тактирования:

1 – процесс переключения частоты тактирования окончен;

0 – процесс переключения не окончен или не начинался.



## Сторожевой таймер (WDT)

### Общая информация

Сторожевой таймер предназначен для принудительной перезагрузки системы в случае ее зависания. В основе таймера лежит счетчик разрядностью 32 бита со счетом «вниз». Модуль формирует регулярное прерывание INT\_WDT в зависимости от запрограммированного значения. Каждый такт синхросигнала значение счетчика уменьшается на единицу. Когда значение счетчика достигает 0, формируется сигнал прерывания. Затем счетчик перезагружается и заново начинает отсчет к нулю. Если к моменту, когда счетчик достиг заново значения 0, прерывание не очищено, то в систему формируется сигнал сброса WDT\_RST. Таким образом, сторожевой таймер предоставляет возможность восстановления системы после сбоя программного обеспечения. При необходимости модуль может быть выключен.

### Регистры сторожевого таймера

№	Аббревиатура	Доступ	Описание
2200h	WDT_LOAD0	RW	Период счета сторожевого таймера 0
2201h	WDT_LOAD1	RW	Период счета сторожевого таймера 1
2202h	WDT_LOAD2	RW	Период счета сторожевого таймера 2
2203h	WDT_LOAD3	RW	Период счета сторожевого таймера 3
2204h	WDT_VAL0	R	Текущее значение сторожевого таймера 0
2205h	WDT_VAL1	R	Текущее значение сторожевого таймера 1
2206h	WDT_VAL2	R	Текущее значение сторожевого таймера 2
2207h	WDT_VAL3	R	Текущее значение сторожевого таймера 3
2208h	WDT_CTRL	RW	Регистр управления
220Ch	WDT_CLR	W	Регистр сброса прерывания
2210h	WDT_INTRAW	R	Регистр исходного прерывания
2214h	WDT_INT	R	Регистр маскируемого прерывания
2218h	WDT_LOCK	RW	Регистр блокировки доступа к сторожевому таймеру
221Ch	WDT_TCR	RW	Регистр перехода в тестовый режим
2220h	WDT_TOP	W	Регистр управления в тестовом режиме

### WDT\_LOAD0

Регистр содержит значение, с которого счетчик начнет уменьшаться. При перезаписи данного регистра таймер немедленно стартует с записанного значения. Минимальное допустимое значение для WDT\_LOAD равно 1. 32-х разрядное значение поместится в регистр сторожевого таймера в момент записи в данный регистр.

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – период счета сторожевого таймера, младшая часть.

### WDT\_LOAD1

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – период счета сторожевого таймера, биты <15:8>.

## WDT\_LOAD2

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – период счета сторожевого таймера, биты <23:16>.

## WDT\_LOAD3

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – период счета сторожевого таймера, старшая часть.

## WDT\_VAL0

Бит	7	6	5	4	3	2	1	0
Назначение	CURRENT_VALUE							
Начальное значение	0							

CURRENT\_VALUE – текущее значение таймера, младшая часть.

## WDT\_VAL1

Бит	7	6	5	4	3	2	1	0
Назначение	CURRENT_VALUE							
Начальное значение	0							

CURRENT\_VALUE – текущее значение таймера, биты <15:8>.

## WDT\_VAL2

Бит	7	6	5	4	3	2	1	0
Назначение	CURRENT_VALUE							
Начальное значение	0							

CURRENT\_VALUE – текущее значение таймера, биты <23:16>.

## WDT\_VAL3

Бит	7	6	5	4	3	2	1	0
Назначение	CURRENT_VALUE							
Начальное значение	0							

CURRENT\_VALUE – текущее значение таймера, старшая часть.

## WDT\_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв						RST_EN	INT_EN
Начальное значение	0							

RST\_EN – разрешение на формирование сигнала сброса по завершению счета таймера:

1 – сброс разрешен;

0 – сброс запрещен.

**INT\_EN** – разрешение на формирование прерывания по окончании счета таймера:

- 1 – прерывание разрешено;
- 0 – прерывание запрещено.

Сторожевой таймер начинает счет, если установлен бит INT\_EN. Сторожевой таймер перестает считать, если бит INT\_EN сброшен в ноль. После разрешения прерывания, если пользователь до этого его запрещал, счетчик автоматически перезагрузится со значения из регистра WDT\_LOAD.

#### WDT\_CLR

Запись любого значения в данный регистр очищает прерывание сторожевого таймера и перезагружает счетчик значением из регистра WDT\_LOAD.

Бит	7	6	5	4	3	2	1	0
Назначение	CLEAR							
Начальное значение	0							

#### WDT\_INTRAW

Данный регистр указывает на необработанное прерывание от счетчика. На основании данного сигнала формируется маскируемое прерывание.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							RAW_WDT
Начальное значение	0							

**RAW\_WDT** – исходное прерывание таймера:

- 1 – прерывание произошло;
- 0 – прерывание отсутствует.

#### WDT\_INT

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							INT_WDT
Начальное значение	0							

**INT\_WDT** – маскируемое прерывание таймера:

- 1 – прерывание произошло;
- 0 – прерывание отсутствует.

Данное прерывание формируется на основании битов RAW\_WDT и INT\_EN и передается в систему.

#### WDT\_LOCK

Данный регистр блокирует доступ на запись во все остальные регистры сторожевого таймера. Запись значения 55h обеспечивает доступ на запись ко всем регистрам. Запись любого другого значения блокирует доступ. По умолчанию запись во все регистры сторожевого таймера заблокирована.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							LOCK_WDT
Начальное значение	0							1

**LOCK\_WDT** – блокирование записи в регистры:

- 1 – запись во все регистры сторожевого таймера заблокирована;
- 0 – запись во все регистры сторожевого таймера разрешена.

## WDT\_TCR

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>							TEST_EN
Начальное значение	0							

**TEST\_EN** – тестовый режим:

- 1 – сторожевой таймер в тестовом режиме;
- 0 – сторожевой таймер в рабочем режиме.

В тестовом режиме пользователь непосредственно управляет маскируемым прерыванием и сбросом от сторожевого таймера через регистр WDT\_TOP.

## WDT\_TOP

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>						TST_INT	TST_RES
Начальное значение	0							

**TST\_INT** – формирование сигнала прерывания от таймера в тестовом режиме:

- 1 – прерывание активно;
- 0 – прерывание не активно.

**TST\_RES** – формирование сигнала сброса от таймера в тестовом режиме:

- 1 – сброс активен;
- 0 – сброс не активен.

## GPIO

### Общая информация

Мультиплексор GPIO\_MUX для каждого вывода микроконтроллера позволяет либо соединить его с портом P процессора (использовать как вывод общего назначения), либо соединить его с одним из периферийных устройств (использовать альтернативную функцию порта). Выбор альтернативной функции осуществляется записью в регистр GPIO\_ALTFL.

Если вывод используется как вывод общего назначения, то блок GPIO позволяет настроить его на вход или на выход. Выбор направления для порта осуществляется записью в регистры GPIO\_DIR\_SET / GPIO\_DIR\_CLR.

Когда порт настроен как выход общего назначения, то передаваемое во вне значение определяется значением порта P процессора. Когда порт настроен как вход общего назначения, то считать значение порта можно также через порт P процессора. Порт P0 процессора соединён с блоками GPIO.

Блок GPIO может сформировать прерывание при определенном уровне или изменении уровня на порту микроконтроллера.

Блок GPIO может зафиксировать фронт сигнала на выводе микроконтроллера, даже когда система находится в режиме «Глубокого сна» (с помощью асинхронного детектора фронта), и вывести систему из режима «SLEEP».

Аналогичным и единственным способом вывода системы из режима «COLD\_SLEEP» является подача положительного фронта на вход любого GPIO. Через такт системной частоты система продолжит работать с той инструкцией, на которой остановилась.

### Структурная схема

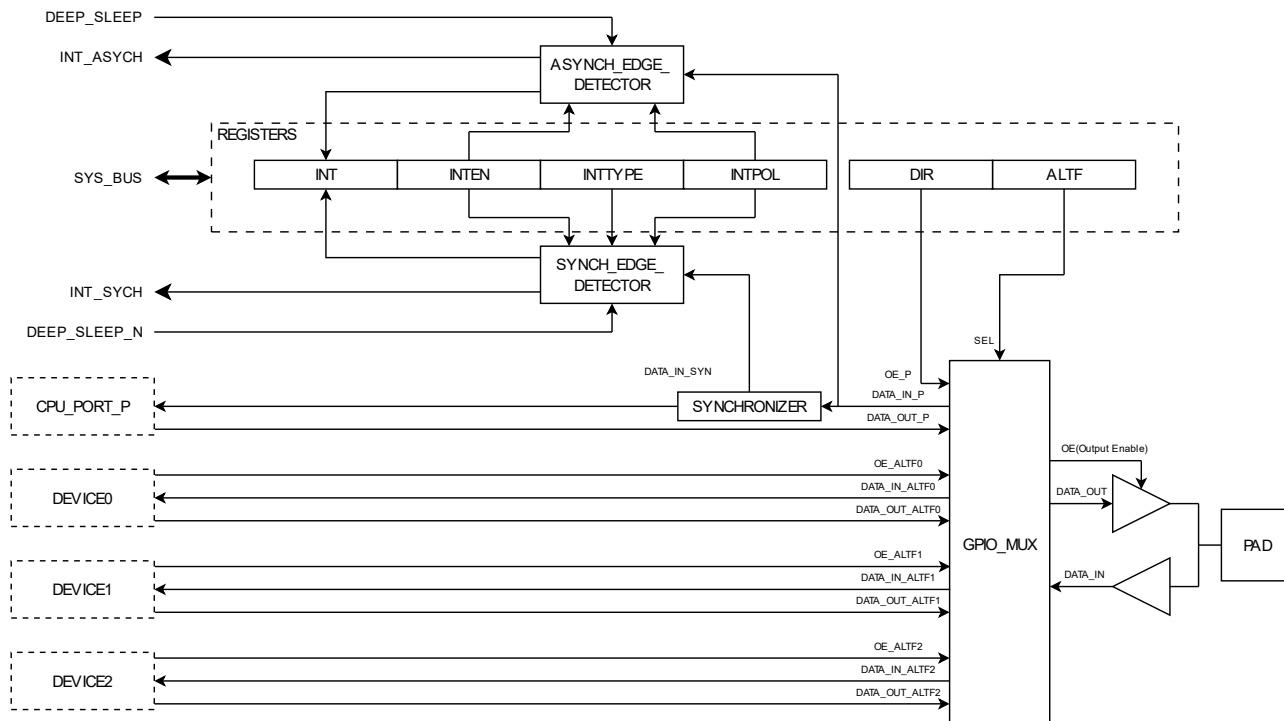


Рисунок 9. Структурная схема GPIO для одного из выводов микроконтроллера

GPIO\_MUX соединяет PAD (вывод микроконтроллера) всегда соединенный с портом P процессора (CPU\_PORT\_P), с одной из альтернативных функций этого вывода. GPIO\_MUX управляется регистрами GPIO\_ALTFL. Если вывод соединен с портом процессора, то направление (вход/выход) определяется регистром GPIO\_DIR\_SET / GPIO\_DIR\_CLR. Если вывод соединен с альтернативной функцией, то направление (вход/выход) определяется этой альтернативной функцией (периферийным устройством DEVICE<sub>x</sub>).

В блоке GPIO присутствуют два детектора, способных формировать прерывания – синхронный (SYNCH\_EDGE\_DETECTOR) и асинхронный (ASYNCH\_EDGE\_DETECTOR). Работа детекторов управляются регистрами GPIO\_INTEN\_SET / GPIO\_INTEN\_CLR, GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR и GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR. Статус прерываний сохраняется в регистре GPIO\_INT.

### Статусы и прерывания

GPIO поддерживает два режима регистрации событий – синхронный и асинхронный. Синхронный детектор работает в рабочем режиме и в режиме «Сон процессора», но не работает в режиме «Глубокий сон». Асинхронный детектор, наоборот работает только в режиме «Глубокий сон» и предназначен для вывода системы из него по внешнему сигналу.

Прерывание для каждого из выводов разрешается и запрещается записью в регистры GPIO\_INTEN\_SET / GPIO\_INTEN\_CLR.

Синхронное прерывание может быть сформировано как по фронту сигнала, так и по уровню, выбор типа прерывания осуществляется записью в регистры GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR. Регистры GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR определяют, какой уровень (низкий/высокий) или какой фронт (возрастающий/спадающий) вызовет прерывание.

Асинхронный детектор не использует системную частоту, поэтому может работать в режиме «Глубокого сна» микроконтроллера. Асинхронное прерывание выводит микроконтроллер из режима «SLEEP», таким образом блок GPIO можно использовать, чтобы выйти из режима «Глубокого сна» по внешнему событию. Асинхронный детектор фронта работает с несинхронизированным на системную частоту входным сигналом, поэтому даже короткий глитч входного сигнала будет гарантированно зарегистрирован как фронт.

Асинхронное прерывание может быть сформировано только по фронту. При переходе в режим «Глубокого сна» (где работает асинхронный детектор) значение в регистрах GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR игнорируется, прерывание срабатывает по фронту сигнала (возрастающему или спадающему, в зависимости от GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR).

Какой именно вывод вызвал прерывание можно выяснить, прочитав регистр статуса прерываний GPIO\_INT. Соответствующий бит в регистре GPIO\_INT выставляется в «1», только если прерывание по этому выводу разрешено.

### Регистры GPIO

№	Аббревиатура	Доступ	Описание
2300h	GPIO_DIR_SET	RW	Установка режима работы выходного буфера
2301h	GPIO_DIR_CLR	RW	
2304h	GPIO_ALTFL	RW	Выбор альтернативной функции
2305h	GPIO_INTEN_SET	RW	Разрешение прерываний
2306h	GPIO_INTEN_CLR	RW	
2307h	GPIO_INTTYPE_SET	RW	Выбор типа прерывания (фронт/уровень)
2308h	GPIO_INTTYPE_CLR	RW	
2309h	GPIO_INTPOL_SET	RW	Выбор полярности входного сигнала, при которой формируются прерывания
230Ah	GPIO_INTPOL_CLR	RW	
230Bh	GPIO_INT	R	Статус прерываний

**GPIO\_DIR\_SET / GPIO\_DIR\_CLR**

GPIO\_DIR\_SET / GPIO\_DIR\_CLR – парные регистры управления режимом работы выходных буферов порта.

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_DIR	IO6_DIR	IO5_DIR	IO4_DIR	IO3_DIR	IO2_DIR	IO1_DIR	IO0_DIR
Начальное значение	0							

Запись в **IOx\_DIR** регистра GPIO\_DIR\_SET:

- 1 – включить выходной буфер на передачу;
- 0 – не меняет текущую настройку.

Запись в **IOx\_DIR** регистра GPIO\_DIR\_CLR:

- 1 – выключить выходной буфер;
- 0 – не меняет текущую настройку.

Чтение **IOx\_DIR** регистров GPIO\_DIR\_SET/GPIO\_DIR\_CLR:

- 1 – выходной буфер включен на передачу;
- 0 – выходной буфер выключен.

**GPIO\_ALTFL**

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_ALTFL	IO6_ALTFL	IO5_ALTFL	IO4_ALTFL	IO3_ALTFL	IO2_ALTFL	IO1_ALTFL	IO0_ALTFL
Назначение	IO3_ALTFL		IO2_ALTFL		IO1_ALTFL		IO0_ALTFL	
Начальное значение	0							

**IOx\_ALTFL** – альтернативная функция:

- 1 – включена альтернативная функция ALTFL;
- 0 – альтернативные функции выключены, выходным буфером управляет GPIO.

**GPIO\_INTEN\_SET / GPIO\_INTEN\_CLR**

GPIO\_INTEN\_SET / GPIO\_INTEN\_CLR – парные регистры установки разрешения генерации прерываний по событиям на входах GPIO.

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_IE	IO6_IE	IO5_IE	IO4_IE	IO3_IE	IO2_IE	IO1_IE	IO0_IE
Начальное значение	0							

Запись в **IOx\_IE** регистра GPIO\_INTEN\_SET:

- 1 – разрешить генерацию прерывания по событиям на данном входе;
- 0 – не меняет текущую настройку.

Запись в **IOx\_IE** регистра GPIO\_INTEN\_CLR:

- 1 – запретить генерацию прерывания по событиям на данном входе;
- 0 – не меняет текущую настройку.

Чтение **IOx\_IE** регистров GPIO\_INTEN\_SET/GPIO\_INTEN\_CLR:

- 1 – разрешена генерация прерывания по событиям на данном входе;
- 0 – запрещена генерация прерывания по событиям на данном входе.

**GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR**

GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR – парные регистры установки типа прерывания (по фронту/уровню) генерируемого GPIO.

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_ITYPE	IO6_ITYPE	IO5_ITYPE	IO4_ITYPE	IO3_ITYPE	IO2_ITYPE	IO1_ITYPE	IO0_ITYPE
Начальное значение	0							

Запись в **IOx\_ITYPE** регистра GPIO\_INTTYPE\_SET:

- 1 – установить генерацию прерывания по фронту;
- 0 – не меняет текущую настройку.

Запись в **IOx\_ITYPE** регистра GPIO\_INTTYPE\_CLR:

- 1 – установить генерацию прерывания по уровню;
- 0 – не меняет текущую настройку.

Чтение **IOx\_ITYPE** регистров GPIO\_INTTYPE\_SET/GPIO\_INTTYPE\_CLR:

- 1 – генерация прерывания осуществляется по фронту;
- 0 – генерация прерывания осуществляется по уровню.

**GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR**

GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR – парные регистры установки полярности события GPIO, по которому генерируется прерывание.

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_IPOL	IO6_IPOL	IO5_IPOL	IO4_IPOL	IO3_IPOL	IO2_IPOL	IO1_IPOL	IO0_IPOL
Начальное значение	0							

Запись в **IOx\_IPOL** регистра GPIO\_INTPOL\_SET:

- 1 – установить генерацию прерывания по положительному фронту или высокому уровню (зависит от GPIO\_INTTYPE\_SET/CLR);
- 0 – не меняет текущую настройку.

Запись в **IOx\_IPOL** регистра GPIO\_INTPOL\_CLR:

- 1 – установить генерацию прерывания по отрицательному фронту или низкому уровню (зависит от GPIO\_INTTYPE\_SET/CLR);
- 0 – не меняет текущую настройку.

Чтение **IOx\_IPOL** регистров GPIO\_INTPOL\_SET/GPIO\_INTPOL\_CLR:

- 1 – генерация прерывания осуществляется по положительному фронту или высокому уровню;
- 0 – генерация прерывания осуществляется по отрицательному фронту или низкому уровню.

**GPIO\_INT**

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_INT	IO6_INT	IO5_INT	IO4_INT	IO3_INT	IO2_INT	IO1_INT	IO0_INT
Тип статуса	EVENT							
Начальное значение	0							

**IOx\_INT** – статус прерывания соответствующего вывода GPIO:

- 1 – был зафиксирован фронт или уровень (согласно заданным в регистрах GPIO\_INTPOL\_SET / GPIO\_INTPOL\_CLR и GPIO\_INTTYPE\_SET / GPIO\_INTTYPE\_CLR условиям) на данном выводе;
- 0 – фронт или уровень не был зафиксирован.



## UART

### Общая информация

UART (универсальный асинхронный приемо-передатчик) осуществляет асинхронный полнодуплексный обмен данными по последовательным линиям RX и TX с другими устройствами UART.

Основные характеристики:

- изменение скорости передачи заданием коэффициента делителя частоты;
- изменение формата посылки. От 1 до 8 бит в слове данных, 1 или 2 стоп-бита, бит контроля четности (4 режима: odd, parity, space, mark);
- входной и выходной FIFO буферы позволяют снизить количество прерываний от UART. Количество слов в буфере (глубина буфера), при которой формируется прерывание, задается программно. Глубина буферов 8 слов;
- тестовые режимы:
  - эхо-режим;
  - режим внутренней петли;
  - режим внешней петли.
- 9-битный режим с автоматической сверкой адреса для систем из нескольких UART;
- высокоскоростной режим (четыре семпла на бит вместо шестнадцати);
- аппаратный контроль обмена через сигналы RTS и CTS;
- детектирование и формирование break-сигнала;
- тайм-аут программируемой длительности;
- возможность инвертирования логических уровней передачи сигнала.

### Структурная схема

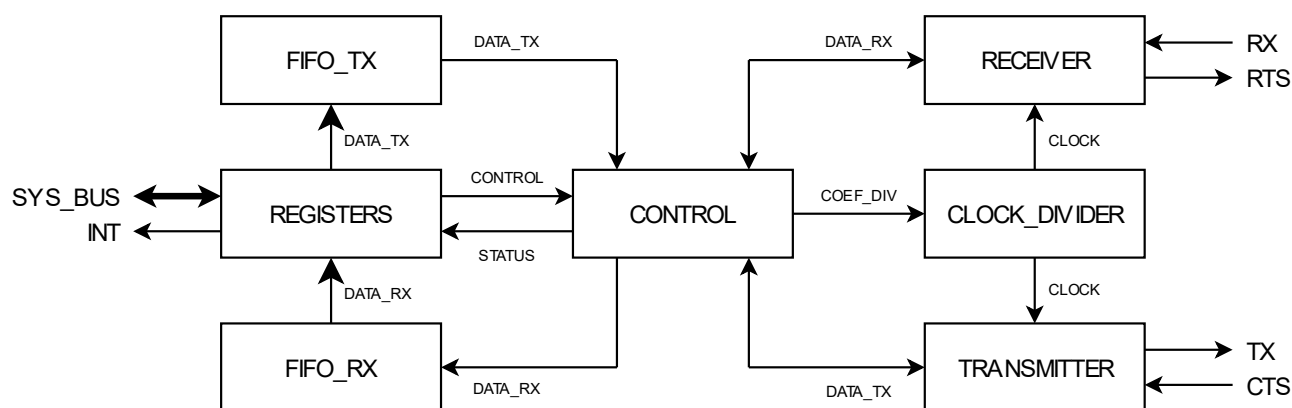


Рисунок 10. Структурная схема UART

UART состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO\_TX – буфер передатчика;
- FIFO\_RX – буфер приемника;
- CONTROL – управляющий автомат модуля UART;
- RECEIVER – приемник, реализующий функцию приема интерфейса;
- TRANSMITTER – передатчик, реализующий функцию передачи интерфейса;
- CLOCK\_DIVIDER – делитель частоты.

## Делитель частоты

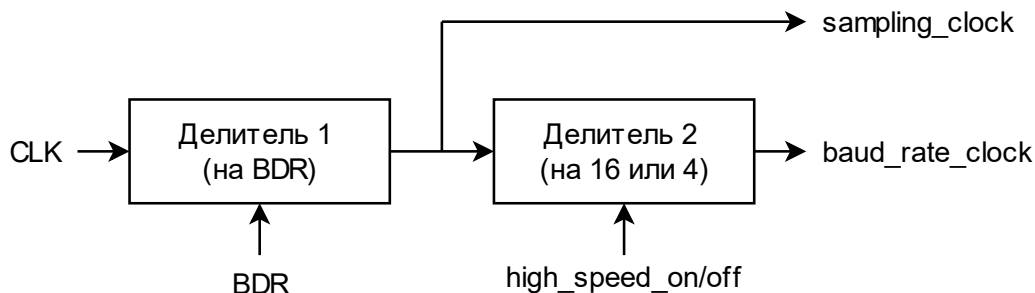


Рисунок 11. Структурная схема делителя частоты

Делитель частоты модуля UART состоит из двух делителей. Первый делитель осуществляет деление системной частоты на BDR – содержимое специальных регистров UART\_BDR0 и UART\_BDR1 (от 0 до 65535). Этот делитель определяет скорость обмена. Полученная частота (*sampling\_clock*) – это частота, с которой приемник сканирует линию RX до приема валидного стартового бита.

Второй делитель делит частоту *sampling\_clock* на 16 или на 4 в высокоскоростном режиме. С полученной частотой (*baud\_rate\_clock*) передатчик выдает биты на линию TX, а приемник сканирует линию RX в процессе приема посылки. На время передачи одного бита приходится несколько (16 или 4) семпла линии RX, для того чтобы приемник мог синхронизироваться по изменению сигнала RX в процессе работы. Частота работы взаимодействующих UART пересинхронизируется (счетчик-делитель сбрасывается) по стартовому биту и каждый раз, когда меняется уровень на линии RX в процессе приема посылки. Это позволяет справиться с «уходом» тактовых частот UART. При BDR = 0 делитель частоты не работает, соответственно не работают приемник и передатчик.

Для установления необходимой скорости обмена необходимо записать в регистры UART\_BDR0 и UART\_BDR1 соответствующий коэффициент деления, который рассчитывается по формуле:

$$BDR = \frac{f_{clk}}{16 * desired\_baud\_rate},$$

где  $f_{clk}$  – частота тактирования системы, *desired\_baud\_rate* – желаемая скорость передачи (в бодах или в бит/с).

При этом если по формуле получилось не целое число, то в результате округления реальная частота обмена будет несколько отличаться от желаемой. Реальная частота обмена (*baud\_rate\_clock*) в бодах рассчитывается по формуле:

$$baud\_rate\_clock = \frac{f_{clk}}{16 * BDR}.$$

Для самого длинного из возможных форматов посылки (12 бит), если ресинхронизации в процессе происходить не будет (то есть посылка состоит из одних единиц/нулей кроме стартового/стоп-бита), максимально допустимая разница частот, взаимодействующих UART – 3,6% (2% в высокоскоростном режиме). При меньшем формате посылки максимально допустимая разница частот соответственно увеличивается.

## Высокоскоростной режим

При выставлении бита «Высокоскоростной режим» регистра UART\_CFG0 в «1» меняется коэффициент второго делителя частоты с 16 на 4. Таким образом, в высокоскоростном режиме делитель частоты осуществляет деление на 4\*BDR, скорость передачи увеличивается в 4 раза. Формулы выше остаются справедливыми, если заменить число 16 на число 4. При этом синхронизация происходит менее точно. Этот режим рекомендуется использовать только при полном отсутствии помех на линии.

## Приемник

После сброса устройства приемник выключен, для включения необходимо записать «1» в бит RE регистра UART\_CTRL. После этого приемник начинает сканировать линию RX с частотой *sampling\_clock*, ожидая старт-бита (уровня логического нуля). При выключении приемника записью в RE «0», UART перестает сканировать линию, однако если в этот момент шла операция приема данных, она будет закончена, и данные будут помещены в буфер приемника.

UART считает старт-битом логический «0» длительностью больше, чем половина времени передачи одного бита ( $9/16 \text{ baud\_rate\_clock}$ , то есть 9 семплов *sampling\_clock*). Если уровень логического «0» держится меньшее время, это считается помехой, и UART продолжает сканировать линию на наличие старт-бита. Если логический «0» держится 9 семплов, то приемник начинает сканировать линию с меньшей в 16 раз частотой *baud\_rate\_clock* (таким образом, что сканирование на временной диаграмме происходит в предполагаемой середине каждого бита), помещая в регистр сдвига значение на линии RX. Когда вся посылка принята (длина посылки определяется форматом посылки, задающимся в регистрах UART\_CFG0 и UART\_CFG1), приемник формирует статус наличия break, ошибки четности/совпадения адреса и стоп-бита (а также соответствующие прерывания, если их формирование разрешено в регистре UART\_MSK0) и помещает их и принятое слово данных в буфер приемника. Если битов в слове данных выставлено меньше восьми в регистре UART\_CFG1, то старшие разряды данных заполняются нулями. Если буфер заполнен, формируется прерывание переполнения буфера приемника. Временная диаграмма работы приемника представлена на рисунке ниже.

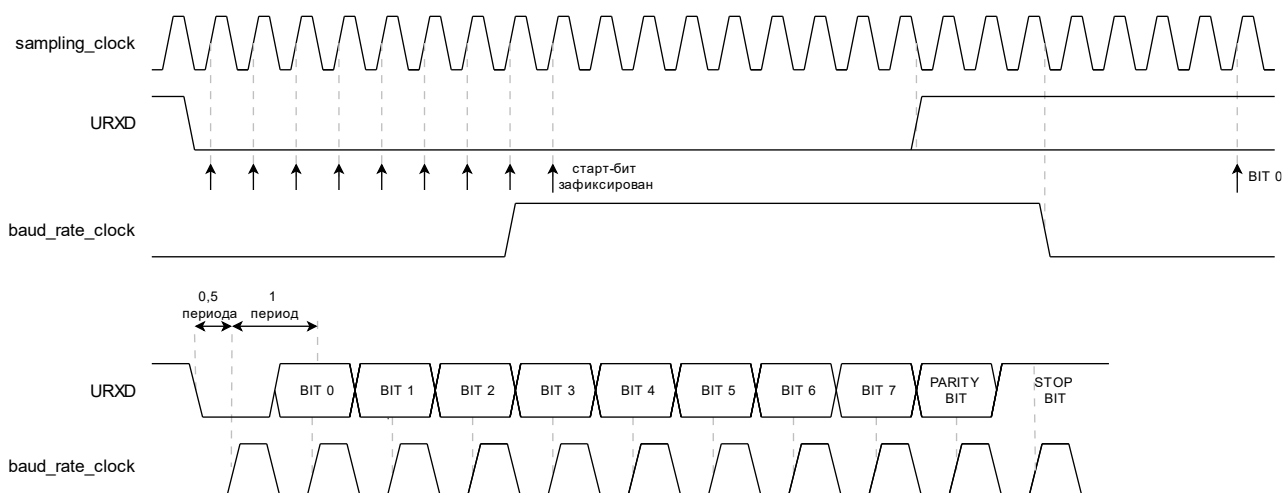


Рисунок 12. Временная диаграмма работы приемника

Между приемником и выводом RX находится простейший фильтр нижних частот: три последних значения сигнала RX (взятых с частотой *sampling\_clock*) попадают в мажоритар 2 из 3, затем результат поступает в приемник. Таким образом, одиночные помехи линии длиной меньше периода *sampling\_clock* не поступают в приемник. Помехи длиной больше периода *sampling\_clock* могут привести к ошибочной ресинхронизации и ошибке приема.

## Буфер приемника

Принятые данные попадают в буфер приемника, откуда могут быть вычитаны программно через регистр UART\_RX0. Буфер имеет 11 разрядов, 8 бит данных и 3 разряда под статус наличия break, ошибки четности/совпадения адреса и стоп-бита. Биты RBRPL, RBF, RBNE регистров UART\_ST0 и UART\_ST1 позволяют контролировать заполненность буфера. Все эти биты равны «0», если приемник выключен.

### Передатчик

После сброса устройства передатчик выключен, для включения необходимо записать «1» в бит TE в UART\_CTRL. При выключении передатчика записью в TE «0» UART сначала завершает передачу текущего слова данных.

Передатчик выдает биты слова данных, полученного из буфера передатчика, с частотой *baud\_rate\_clock* в соответствии с форматом посылки, указанным в UART\_CFG0 и UART\_CFG1. Примечание: если передатчик и буфер передатчика пусты, от момента записи в UART\_TX до начала передачи, может пройти время меньше или равное периоду *baud\_rate\_clock* (время передачи одного бита). Это происходит из-за того, что передатчик ждет следующего импульса *baud\_rate\_clock*, чтобы начать передачу посылки.

### Буфер передатчика

Передатчик берет данные для отправки из буфера передатчика, куда их можно поместить записью в регистр UART\_TX. Буфер передатчика имеет 9 разрядов, 8 из них для передаваемого слова данных. 9-ый бит используется только в 9-битном режиме как идентификатор адреса и передается вместо бита четности. Биты TBNF, TBRPL, TBE, TI регистра UART\_ST1 позволяют контролировать заполненность буфера. Все эти биты равны 0, если передатчик выключен. Если после начальной конфигурации и задания маски прерываний включить передатчик, сработают все эти прерывания.

### Прерывания

По любому из битов UART\_ST0 или UART\_ST1 можно разрешить формировать прерывание, записав «1» в соответствующий бит регистра маски прерываний UART\_MSK0 или UART\_MSK1 (расположение битов в регистрах одинаково).

Прерывание возникает, как только соответствующее событие было зафиксировано. Из-за наличия буфера может быть не очевидно, при приеме какого именно слова возникла ошибка, поэтому статус наличия break, ошибки четности/совпадения адреса и ошибки стоп-бита данного слова доступны также через младшие разряды регистра UART\_RX1.

### Таймер тайм-аута

В режиме сканирования линии приемником каждый период *baud\_rate\_clock* инкрементируется счетчик-таймера тайм-аута. Он не работает, если приемник выключен. Таймер сбрасывается в «0», как только зафиксирован валидный старт-бит и остается в «0» во время приема данных. После приема стоп-бита, он начинает считать снова. Таймер сбрасывается в «0», при записи «1» в бит RTT регистра UART\_CTRL и остается в «0» до момента записи «0» в RTT. Максимальное значение, до которого досчитывает таймер, определяется в разрядах TV регистра UART\_CFG0. Возможные значения: 2, 4, 8, 16, 32, 64, 128, 256 периодов *baud\_rate\_clock*. Досчитав до этого значения, таймер останавливается, статусный бит RTO переходит в «1» и генерируется соответствующее прерывание, если его формирование разрешено в регистре UART\_MSK0.

### Генерация и распознавание сигнала break

Сигнал break, то есть уровень логического «0» на линии на время большее времени передачи посылки, используется как индикатор серьезного сбоя в работе. Даже если никакая информация не может быть передана из-за слишком большой разности в частотах двух UART, долгий логический «0» на линии может быть правильно принят и интерпретирован приемником. Также в большинстве физических реализаций при разрыве линии сигнал RX также принимает значение логического «0».

Приемник засчитывает за сигнал break посылку, в которой старт-бит, все биты данных, бит четности (если, конечно, бит четности включен в регистре UART\_CFG1) и стоп-бит равны «0» (ошибка стоп-бита при этом не возникает). В момент приема последнего из битов бит BD переходит в «1» и генерируется соответствующее прерывание, если его формирование включено в регистре UART\_MSK0.

В буфер приемника при этом записывается одно слово из восьми нулей с выставленным в «1» битом BD. Break сигнал может длиться и дольше, приемник ждет конца сигнала, то есть перехода линии RX в «1», и только потом продолжает работу.

Передатчик выдает сигнал break при записи «1» в бит SB регистра UART\_CTRL. Если запись произошла в момент передачи слова, то передача будет закончена перед выдачей сигнала break. Линия TX удерживается в состоянии логического «0» до момента записи «0» в разряд SB, таким образом продолжительность сигнала определяется программно. После окончания сигнала линия TX удерживается в «1» на 12 периодов *baud\_rate\_clock*, чтобы удаленный приемник смог корректно определить конец сигнала, предпринять необходимые действия и начать прием следующей посылки.

### Особые режимы работы

Выбор режима работы происходит записью в разряды MODE регистра UART\_CFG1.

**Эхо-режим.** В этом режиме все данные принятые UART с линии RX побитово ретранслируются на линию TX. Передатчик при переходе в этот режим работы автоматически выключается, программно включить его в регистре UART\_CTRL нельзя.

**Режим внутренней петли.** Пины UART TX и RX замыкаются внутри устройства, таким образом, выдаваемая передатчиком информация может быть принята приемником. При этом на линию TX выдается высокий уровень. Этот режим применяется для тестирования работы UART без использования линии и другого UART.

**Режим внешней петли.** Линии TX и RX замыкаются внутри устройства, таким образом вся информация, принятая по линии RX, попадает на линию TX. Отличие от эхо-режима в том, что эта информация не фиксируется приемником UART. Приемник и передатчик при переходе в этот режим работы автоматически выключаются, программно включить их записью в UART\_CTRL нельзя.

**9-битный режим.** 9-бит режим, или режим с аппаратным детектированием и сверкой адреса, необходим для соединения по UART одного ведущего устройства с несколькими ведомыми. Этот режим работы может быть полезен при реализации сетевых протоколов, например Modbus RTU. При работе в этом режиме формат посылки тот же, что и в обычном режиме, но вместо бита четности ведущий UART передает бит, определяющий тип данных в посылке: 1 - в посылке содержится адрес ведомого устройства, 0 - в посылке обычное слово данных (настройки бита четности PE и PT в регистрах конфигурации игнорируются в 9-бит режиме). Ведомое устройство распознает свой адрес, принимает последующие данные и/или передает ведущему устройству свои данные.

По умолчанию после включения 9-битного режима приемник не принимает слова данных, а адреса (слова данных с «1» в девятом бите) сравнивает с заданными в регистрах UART\_NBADDR0 и UART\_NBADDR1 (обычно один адрес индивидуальный для ведомого, а второй – широковещательный, общий для всех ведомых). При этом приемник сравнивает только биты адреса незамаскированные регистром UART\_NBMSK. Если адрес совпадает, то приемник посылает его в буфер вместе с 3-мя статусными битами (при чтении этого слова бит PE/AM регистра UART\_RX1 установлен в «1», показывая, что это слово – адрес), устанавливает соответствующий статус (PE/AM) в регистре UART\_ST0 и формирует прерывание по совпадению адреса, если его формирование разрешено в регистре UART\_MSK0. После этого приемник начинает принимать последующие слова данных (с «0» вместо бита четности) и продолжает делать это, пока не получит новую посылку с адресом («1» вместо бита четности). Если в формате посылки выставлено меньше восьми битов в слове, то перед сверкой идет дополнение нулями до восьми разрядов (в 9-битном режиме рекомендуется все же выставить 8 бит в слове, хотя это не обязательно). Передатчик вместо бита четности отправляет значение бита SADDR регистра UART\_CTRL.

### Аппаратный контроль обмена

Для установления аппаратного контроля обмена необходимо соединить вывод RTS (Request To Send, запрос на отправку данных, выходной сигнал) с выводом CTS того UART, с которым происходит обмен, и соединить вывод CTS (Clear To Send, запрос на получение данных, входной сигнал) с выводом RTS того UART, с которым происходит обмен. При работе в этом режиме передатчик начинает отправку посылки, только если сигнал CTS равен «0».

При работе в этом режиме при приеме старт-бита оценивается состояние буфера приемника. Если в результате приема данного слова буфер приемника будет полностью заполнен, то сигнал RTS переходит из «0» в «1», запрещая удаленному передатчику отправлять следующие данные. Чтение регистра RX переводит RTS обратно из «1» в «0», разрешая удаленному передатчику дальнейшую отправку. Если приемник выключен, RTS равно «1».

В других режимах работы сигнал RTS управляется битом RTS регистра UART\_CTRL. Этот сигнал можно использовать для управления внешним приемопередатчиком физического уровня, например, RS-485.

### Регистры UART

№	Аббревиатура	Доступ	Описание
2400h	UART_CFG0	RW*	Регистр конфигурации 0
2401h	UART_CFG1	RW*	Регистр конфигурации 1
2404h	UART_BDR0	RW*	Регистр настройки скорости обмена 0
2405h	UART_BDR1	RW*	Регистр настройки скорости обмена 1
2408h	UART_TXFIFOLVL0	RW	Регистр контроля уровня заполнения буфера передатчика 0
2409h	UART_TXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера передатчика 1
240Ch	UART_RXFIFOLVL0	RW	Регистр контроля уровня заполнения буфера приемника 0
240Dh	UART_RXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера приемника 1
2410h	UART_NBMSK	RW	Регистр маски адреса 9-битного режима
2414h	UART_NBADDR0	RW	Регистр фильтра адреса в 9-битном режиме 0
2415h	UART_NBADDR1	RW	Регистр фильтра адреса в 9-битном режиме 1
2418h	UART_MSK0	RW	Регистр маски прерываний 0
2419h	UART_MSK1	RW	Регистр маски прерываний 1
241Ch	UART_CTRL	RW	Регистр управления
2420h	UART_TX	W	Регистр буфера передатчика
2424h	UART_RX0	R	Регистр буфера приемника 0
2425h	UART_RX1	R	Регистр буфера приемника 1
2428h	UART_ST0	R	Регистр статусов 0
2429h	UART_ST1	R	Регистр статусов 1

\* – Запись в регистры UART\_CFG0, UART\_CFG1, UART\_BDR0, UART\_BDR1 **запрещена аппаратно** во время работы UART, т.к. изменение содержимого этих регистров во время работы может привести к ошибкам передачи или приема. Поэтому запись в регистры производится один раз в начале работы. Если необходимо изменить содержимое регистров, то приемник и передатчик должны быть выключены, и текущие операции приема/передачи должны закончиться. Когда эти условия выполнены, бит CRWE регистра UART\_ST1 переходит в «1», показывая, что запись в эти регистры разрешена.

## UART\_CFG0

Бит	7	6	5	4	3	2	1	0
Назначение	PT		NSB	HSE	Резерв	TV		
Начальное значение	0					4		

**PT (Parity Type)** – тип контроля четности:

11b – mark, бит четности всегда равен «1»;

10b – space, бит четности всегда равен «0»;

01b – odd, бит четности формируется как XNOR всех битов данных в слове;

00b – even, бит четности формируется как XOR всех битов данных в слове.

**NSB (Number of Stop Bits)** – количество стоп-битов в посылке:

1 – два стоп-бита. Приемник не проверяет наличие второго стоп-бита, эта настройка влияет только на передатчик;

0 – один стоп-бит.

**HSE (High Speed Enabled)** – высокоскоростной режим:

1 – высокоскоростной режим включен. Коэффициент делителя частоты равен 4;

0 – высокоскоростной режим выключен. Коэффициент делителя частоты равен 16.

**TV (Time-out Value)** – длительность тайм-аута в периодах *baud\_rate\_clock*:

111b – 256;

110b – 128;

101b – 64;

100b – 32;

011b – 16;

010b – 8;

001b – 4;

000b – 2.

## UART\_CFG1

Бит	7	6	5	4	3	2	1	0
Назначение	INVE	MODE			CHRL			PE
Начальное значение	0				7			0

**INVE (Inversion Enabled)** – инверсия сигналов линии:

1 – сигналы линии обмена (RX, TX, RTS, CTS) инвертированы. Активный уровень для RX и TX – «0», для RTS и CTS – «1»;

0 – сигналы линии обмена не инвертированы.

**MODE** – режим работы:

101b – аппаратный контроль обмена;

100b – 9-битный режим;

011b – режим внешней петли;

010b – режим внутренней петли;

001b – эхо-режим;

000b – обычный режим работы.

**CHRL (Character Length)** – размер слова данных:

111b – 8 бит;  
 110b – 7 бит;  
 101b – 6 бит;  
 100b – 5 бит;  
 011b – 4 бита;  
 010b – 3 бита;  
 001b – 2 бита;  
 000b – 1 бит.

**PE (Parity Enabled)** – контроль четности:

1 – контроль четности включен, бит четности передается после слова данных;  
 0 – контроль четности выключен, бит четности не формируется.

#### UART\_BDR0

Бит	7	6	5	4	3	2	1	0
Назначение	BDR							
Начальное значение	0							

**BDR** – коэффициент деления, младшая часть.

#### UART\_BDR1

Бит	7	6	5	4	3	2	1	0
Назначение	BDR							
Начальное значение	0							

**BDR** – коэффициент деления, старшая часть.

#### UART\_TXFIFOLVL0

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*				TXFIFOLVL			
Начальное значение	0							

\* – Не изменять начальное значение.

**TXFIFOLVL (Transmitter FIFO Buffer Level)** – заданный уровень буфера передатчика. Когда количество слов в буфере передатчика меньше или равно заданному уровню, разряд TBRPL регистра UART\_ST1 равен «1». Таким образом, в момент опустошения буфера до заданного уровня, бит TBRPL регистра UART\_ST1 переходит из «0» в «1».

#### UART\_TXFIFOLVL1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*							
Начальное значение	0							

\* – Не изменять начальное значение.



## UART\_RXFIFOLVL0

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*				RXFIFOLVL			
Начальное значение	0							

\* – Не изменять начальное значение.

**RXFIFOLVL (Receiver FIFO Buffer Level)** – заданный уровень буфера приемника. Когда количество слов в буфере приемника больше или равно заданному уровню, разряд RBRPL регистра UART\_ST0 равен «1». Таким образом, в момент заполнения буфера до заданного уровня, бит RBRPL регистра UART\_ST0 переходит из «0» в «1».

## UART\_RXFIFOLVL1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв*							
Начальное значение	0							

\* – Не изменять начальное значение.

## UART\_NBMSK

Содержимое этого регистра имеет эффект только в 9-битном режиме.

Бит	7	6	5	4	3	2	1	0
Назначение	NBMSK							
Начальное значение	255							

**NBMSK (Nine Bit Mode Address Mask)** – маска адреса в 9-битном режиме:

1 – этот разряд сравнивается при приеме адреса в 9-битном режиме;

0 – этот разряд не сравнивается при приеме адреса в 9-битном режиме.

## UART\_NBADDR0

Содержимое этого регистра имеет эффект только в 9-битном режиме.

Бит	7	6	5	4	3	2	1	0
Назначение	NBA2							
Начальное значение	0							

**NBA2 (Nine Bit Mode Address 2)** – адрес устройства в 9-битном режиме №2.

## UART\_NBADDR1

Содержимое этого регистра имеет эффект только в 9-битном режиме.

Бит	7	6	5	4	3	2	1	0
Назначение	NBA1							
Начальное значение	0							

**NBA1 (Nine Bit Mode Address 1)** – адрес устройства в 9-битном режиме №1.

**UART\_MSK0**

Возможно формирование прерывания по любому биту статусного регистра UART\_ST0. Расположение битов в UART\_ST0 и UART\_MSK0 аналогично.

Бит	7	6	5	4	3	2	1	0
Назначение	RBRPL	RBF	OE	RTO	BD	PE/AM	FE	CTSIC
Начальное значение	0							

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**UART\_MSK1**

Возможно формирование прерывания по любому биту статусного регистра UART\_ST1. Расположение битов в UART\_ST1 и UART\_MSK1 аналогично.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв	CTSI	CRWE	TBNF	TBRPL	TBE	TI	RBNE
Начальное значение	0							

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

**UART\_CTRL**

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		RE	TE	SB	RTS	RTT	SADDR
Начальное значение	0							

**RE (Receiver Enabled):**

- 1 – приемник включен;
- 0 – приемник выключен, если в момент выключения происходил прием слова, он будет завершен перед выключением.

**TE (Transmitter Enabled):**

- 1 – передатчик включен;
- 0 – передатчик выключен, если в момент выключения происходила передача слова, она будет завершена перед выключением.

**SB (Send Break)** – устанавливает линию TX в «0». Если в этот момент происходила передача посылки, она будет завершена перед выдачей break сигнала. Сигнал заканчивается при записи «0» в этот бит, таким образом длительность сигнала break задается программно. После конца сигнала break линия TX устанавливается в «1» на 12 периодов *baud\_rate\_clock*. После этого передатчик продолжает работу.

**RTS (Request To Send)** – управляет сигналом RTS:

- 1 – пассивный уровень сигнала RTS (RTS установлен в «1», если инверсия сигналов линии не включена битом INVE регистра UART\_CFG1);
- 0 – активный уровень сигнала RTS (RTS установлен в «0», если инверсия сигналов линии не включена битом INVE регистра UART\_CFG1).

Значение этого бита не имеет эффекта в режиме работы «Аппаратный контроль обмена», так как в этом режиме UART сам управляет сигналом RTS.

**RTT (Reset Timeout Timer)** – сбросить таймер тайм-аута:

1 – таймер тайм-аута сброшен в «0» и не работает;

0 – таймер тайм-аута работает.

**SADDR (Send Address)** – отправить адрес, этот бит имеет эффект только в 9-битном режиме:

1 – все записываемые в TX слова отправляются как адреса (с «1» вместо бита четности);

0 – все записываемые в TX слова отправляются как слова данных (с «0» вместо бита четности).

## UART\_TX

Запись в этот регистр запрещена, если буфер передатчика полон (бит TBNF регистра ST равен «0»). Стоит избегать записи в заполненный буфер.

Бит	7	6	5	4	3	2	1	0
Назначение	CHTBT							
Начальное значение	0							

**CHTBT (Character To Be Transmitted)** – слово данных для передачи. Если количество битов в слове (биты CHRL в регистре UART\_CFG1) выбрано меньше 8, то передаваемые биты – младшие. Например, если выбрано 5 битов в слове, а в CHTBT записано 00001111b, то передатчик отправит на линию TX 01111b (младшими битами вперед).

## UART\_RX0

Чтение буфера приемника UART\_RX0, если тот пуст (бит RBNE регистра UART\_ST1 равен «0») приводит к выдаче «0» во всех разрядах. Стоит избегать чтения пустого буфера.

Бит	7	6	5	4	3	2	1	0
Назначение	RCH							
Начальное значение	0							

**RCH (Received Character)** – принятое слово данных. Если количество битов в слове (биты CHRL в регистре UART\_CFG1) выбрано меньше 8, то принятые биты дополняются нулями слева до 8 перед записью в RCH. Например, если выбрано 5 битов данных в слове и приемник принял 10110b, тогда RCH = 00010110b.

## UART\_RX1

Если необходимо считать UART\_RX1 и UART\_RX0, то сначала нужно считать данные регистра UART\_RX1, а затем UART\_RX0. Если нужно считать только принятое слово данных, то в чтение регистра UART\_RX1 необходимости нет.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв					BD	PE/AM	FE
Начальное значение	0							

**BD (Break Detected)** – это слово данных состоит только из «0», даже вместо стоп-бита, и поэтому было интерпретировано как сигнал break.

**PE/AM (Parity Error/Address Match)** – значение этого бита зависит от режима работы:

1 – обнаружена ошибка четности или это слово – адрес.

0 – ошибка четности не обнаружена или это слово – слово данные.

Во всех режимах кроме 9-битного значение бита PE/AM – ошибка четности при приеме данного слова. В 9-битном режиме, где бит четности не используется, этот бит показывает содержимое девятого бита принятого слова. Слово вместе с битами статуса отправляются в буфер приемника, только если адрес совпал при сверке.

**FE (Frame Error)** – при приеме этого слова произошла ошибка стоп-бита, то есть вместо «1» на месте старт бита принят «0».

#### UART\_ST0

Биты BD, PE/AM, FE при приеме слова выставляются в «1» даже в случае, если в буфере приемника не было места, чтобы сохранить принятое слово.

Бит	7	6	5	4	3	2	1	0
Назначение	RBRPL	RBF	OE	RTO	BD	PE/AM	FE	CTSIC
Тип статуса	FLAG	FLAG	EVENT					
Начальное значение	0							

**RBRPL (Receiver Buffer Reached Preprogrammed Level)** – в буфере приемника количество слов больше или равно заданному в регистре RXFIFOLVL уровню (и приемник включен).

**RBF (Receiver Buffer Full)** – буфер приемника полон (и приемник включен). Если приемник закончит прием еще одного слова данных, это приведет к ошибке Overrun Error.

**OE (Overrun Error)** – произошла ошибка переполнения приемника, то есть стоп-бит был принят в момент, когда буфер приемника был полон, в результате чего принятое слово было утеряно.

**RTO (Receiver Timeout)** – произошел тайм-аут приемника, то есть приемник не зафиксировал старт-бита после приема последнего слова по истечению количества периодов baud\_rate\_clock, заданного в разрядах TV регистра UART\_CFG0.

**BD (Break Detected)** – обнаружен сигнал break, то есть линия RX удерживалась в «0» в течение времени, большего времени передачи посылки (иными словами, принята посылка, состоящая только из нулей, даже в стоп-бите). При этом приемник записывает в FIFO-буфер одно слово из восьми нулей, вне зависимости от того, как долго держится «0» на RX. Прием следующей посылки начнется только после перехода линии в «1», а затем приема старт-бита. Сигнал break не считается ошибкой стоп-бита.

**PE/AM (Parity Error/Address Match)** – зафиксирована ошибка четности/совпадение адреса в 9-битном режиме. Значение этого бита зависит от режима работы UART. Во всех режимах работы, кроме 9-битного режима, значение этого бита – ошибка четности, то есть содержимое бита четности принятой посылки не соответствовало биту, сформированному приемником для проверки в соответствии с выбранным типом контроля. В 9-битном режиме значение этого бита – совпадение адреса, то есть приемник принял адрес (посылка с «1» вместо бита четности), совпадающий с одним из адресов, заданных в регистрах UART\_NBADDR0 и UART\_NBADDR1. При этом сравниваются только разряды адреса незамаскированные в UART\_NBMSK.

**FE (Frame Error)** – зафиксирована ошибка стоп-бита, то есть в принятой посылке значение стоп-бита было равно «0». Этот бит не устанавливается в «1», если все остальные биты слова также равны «0», так как это воспринимается как сигнал break.

**CTSIC (Clear To Send Input Change)** – произошло изменение сигнала CTS. Это событие не возникает, если передача запрещена, чтобы избежать ложной генерации этого события после сброса.

## UART\_ST1

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>	CTSI	CRWE	TBNF	TBRPL	TBE	TI	RBNE
Тип статуса	FLAG							
Начальное значение	0	1	0					

**CTSI (Clear To Send Image)** – текущее значение сигнала CTS:

1 – удаленный приемник готов принимать данные;

0 – удаленный приемник не готов принимать данные.

Значение этого бита берется из входного сигнала CTS. Если инверсия сигналов линии (бит INVE регистра UART\_CFG1) выключена, то он равен инверсии CTS, если включена, то самому сигналу CTS.

**CRWE (Config Registers Write Enable)** – запись в регистры UART\_CFG0, UART\_CFG1, UART\_BDR0, UART\_BDR1 разрешена. Этот бит равен «1» при условиях:

- приемник и передатчик выключены;
- приемник и передатчик закончили свои операции.

**TBNF (Transmitter Buffer Not Full)** – буфер передатчика не полон, в нем есть место еще как минимум для одного слова данных (и передатчик включен).

**TBRPL (Transmitter Buffer Reached Preprogrammed Level)** – в буфере передатчика осталось количество слов меньше или равно заданному в регистрах UART\_TXFIFOLVL0 и UART\_TXFIFOLVL1 уровню (и передатчик включен).

**TBE (Transmitter Buffer Empty)** – буфер передатчика пуст (и передатчик включен).

**TI (Transmitter Idle)** – буфер передатчика пуст и у него нет больше слов для отправки (и передатчик включен). Формируется до завершения последнего стоп-бита. Таким образом, если новое слово будет записано до завершения последнего стоп-бита, то новое слово начнет отправляться сразу после его завершения (при CTS в активном уровне в режиме аппаратного контроля обмена).

**RBNE (Receiver Buffer Not Empty)** – в буфере приемника есть как минимум одно неп прочитанное слово данных (и приемник включен).

## Постоянная память данных

### Общая информация

Постоянная память данных представляет собой однократно программируемую память ёмкостью 64Б. Данные записываются побайтно. Требуемое время записи одного байта 200 мс.

Область памяти размещена в старших 64 байтах ПЗУ OTP, но доступна программно по адресам 0x2500. Поскольку память расположена физически в памяти программ, то после команды записи в нее CPU переходит в режим ожидания на интервал, заданный в регистрах OTP\_PRESC (номинально 200 мс).

### Алгоритм записи данных

- Задать в регистрах OTP\_PRESC (L / M / H) интервал, соответствующий не менее 200 мс в тактах частоты системы;
- В регистры OTP\_PRE (L / M / H) и OTP\_POST (L / M / H) задать защитные интервалы не менее 100 мс для включения напряжения программирования;
- Записать байт данных по адресам 0x2500 – 0x253F.

### Алгоритм чтения данных

- Данные доступны для чтения по адресам 0x2500 – 0x253F.
- Данные доступны в области программ по старшим 64 адресам ПЗУ OTP.

### Регистры модуля управления OTP ПЗУ 64Б

№	Аббревиатура	Доступ	Описание
2500h	OTP_ROM_DATA	RW	Регистр для записи/считывания данных
2540h	OTP_PRESC_L	RW	Интервал программирования 1
2544h	OTP_PRESC_M	RW	Интервал программирования 2
2548h	OTP_PRESC_H	RW	Интервал программирования 3
2550h	OTP_PRE_L	RW	Начальный защитный интервал 1
2554h	OTP_PRE_M	RW	Начальный защитный интервал 2
2558h	OTP_PRE_H	RW	Начальный защитный интервал 3
2560h	OTP_POST_L	RW	Конечный защитный интервал 1
2564h	OTP_POST_M	RW	Конечный защитный интервал 2
2568h	OTP_POST_H	RW	Конечный защитный интервал 3

### OTP\_ROM\_DATA

Бит	7	6	5	4	3	2	1	0
Назначение	Данные на запись или чтение [7:0]							
Начальное значение	0							

Данный регистр предназначен для записи и чтения данных. При записи в регистр инициализируется процесс записи в ПЗУ по соответствующему адресу.

## OTP\_PRESC\_L

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – интервал программирования, младшая часть.

## OTP\_PRESC\_M

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – интервал программирования, биты <15:8>.

## OTP\_PRESC\_H

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		VALUE					
Начальное значение	0							

VALUE – интервал программирования, старшая часть.

## OTP\_PREL

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – начальный защитный интервал, младшая часть.

## OTP\_PREM

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – начальный защитный интервал, биты <15:8>.

## OTP\_PREN

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		VALUE					
Начальное значение	0							

VALUE – начальный защитный интервал, старшая часть.

## OTP\_POSTL

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – конечный защитный интервал, младшая часть.

## OTP\_POSTM

Бит	7	6	5	4	3	2	1	0
Назначение	VALUE							
Начальное значение	0							

VALUE – конечный защитный интервал, биты <15:8>.

## OTP\_POSTH

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		VALUE					
Начальное значение	0							

VALUE – конечный защитный интервал, старшая часть.

Диаграмма записи имеет следующий вид. Команда записи в ROM\_DATA (WR) приводит к переводу процессора в режим ожидания (READY) и началу отсчета интервала до включения напряжения программирования (VPP\_CTRL). Диаграмма напряжения программирования состоит из 3 фаз – PRE, PRESC, POST.

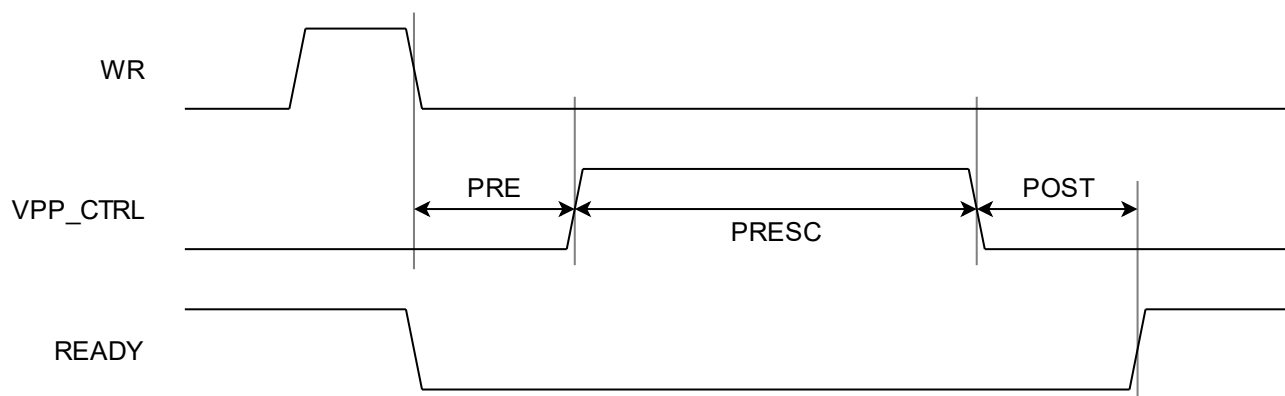


Рисунок 13. Временная диаграмма записи



## Рабочий автомат (WORK\_FSM)

### Общая информация

Модуль «Рабочий автомат» представляет собой 24-разрядный таймер. Данный таймер считает «вверх» на частоте системы. Начало работы данного модуля автоматически означает переход системы в режим «Глубокий сон» (более подробно смотрите пункт «Режим «SLEEP»»). По окончании счета будет сформировано немаскируемое прерывание, которое выведет систему из режима «Глубокий сон».

### Особенности работы

Актуальное значение периода счета обязательно должно быть записано перед включением модуля «Рабочий автомат». Таймер модуля «Рабочий автомат» работает на частоте системы. Это значит, что перед запуском данного модуля для снижения энергопотребления пользователь обязан переключить систему на частоту тактирования от RC-генератора в регистре CMM\_CTRL.

Существует два способа запуска модуля «Рабочий автомат»:

- классический запуск;
- запуск с ожиданием.

При классическом запуске пользователь переключает частоту тактирования (модуль CMM). Пользователю необходимо узнать о завершении данного процесса по прерыванию, либо по чтению из статусного регистра CMM\_ST (бит SWITCH). И только после этого пользователь может запустить модуль «Рабочий автомат». По записи сигнала EN в регистр FSM\_CTRL модуль начнет счет «вверх» на частоте системы и переведет микроконтроллер в режим «Глубокий сон».

Для запуска с ожиданием пользователь должен записать «1» в бит EN\_WSW регистра FSM\_CTRL. Модуль «Рабочий автомат» будет ожидать события переключения частоты (модуль CMM). Пользователю необходимо записать команду на переключение частоты в регистр CMM\_CTRL. По окончании процесса переключения модуль «Рабочий автомат» автоматически начнет счет и инициирует процесс перехода в режим «Глубокий сон».

По окончании счета «Рабочий автомат» сформирует прерывание, по которому микроконтроллер начнет выход из режима «Глубокий сон».

### Регистры «Рабочего автомата»

№	Аббревиатура	Доступ	Описание
2600h	FSM_CTRL	W	Регистр управления
2601h	FSM_PRDH	RW	Регистр периода счета 2
2602h	FSM_PRDM	RW	Регистр периода счета 1
2603h	FSM_PRDL	RW	Регистр периода счета 0

### FSM\_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв						EN_WSW	EN
Начальное значение	0							

**EN\_WSW (Enable Wait Switch and Work)** – разрешение работы модуля по переключению частоты:

- 1 – ожидать переключения;
- 0 – модуль выключен.

**EN** – разрешение работы модуля:

- 1 – модуль включен;
- 0 – модуль выключен.

## FSM\_PRDH

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

PRD – период счета таймера, старшая часть.

## FSM\_PRDM

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

PRD – период счета таймера, средняя часть.

## FSM\_PRDL

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

PRD – период счета таймера, младшая часть.

Значение, до которого будет считать модуль определяется как:

$$N_{clk} = PRD + 1,$$

где  $N_{clk}$  – количество тактов системной частоты.

## Таймер

### Общая информация

В системе присутствует 3 модуля таймера. Каждый таймер поддерживает 3 различных независимых режима работы. Кроме того, TIMER0 и TIMER1 дополнительно поддерживают 4-ый режим работы, в котором они взаимодействуют между собой. Каждый модуль имеет внешний вход, который, в зависимости от режима, управляет таймером.

### Структурная схема

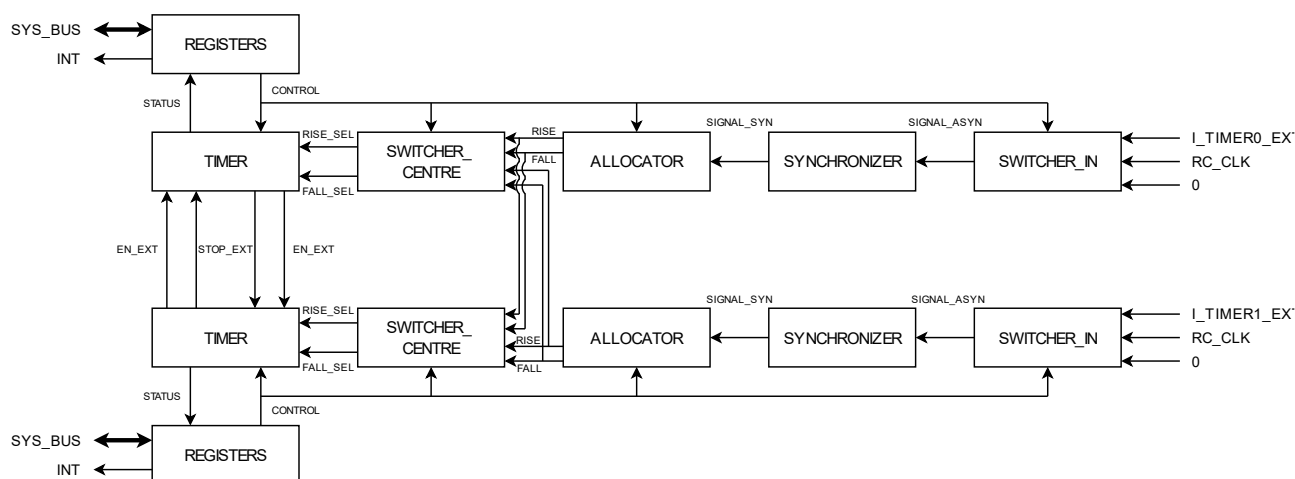


Рисунок 14. Структурная схема двух соединенных модулей таймера

Таймер состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- TIMER – блок управления счетом;
- SWITCHER\_CENTRE – переключатель, определяет будут ли использованы внутренние сигналы или сигналы с соседнего таймера;
- ALLOCATOR – выделитель, выделяет фронт и спад внешнего сигнала;
- SYNCHRONIZER – синхронизатор, синхронизирует внешний сигнал к системной частоте;
- SWITCHER\_IN – переключатель, определяет источник внешнего сигнала.

### Работа в режиме «Простой таймер»

#### Принцип работы

В режиме «Простой таймер» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. Таймер считает до значения, записанного в регистры периода счета таймера TMR\_PRDH, TMR\_PRDM и TMR\_PRDL. По достижению заданного значения таймер либо останавливается, либо начинает счет с нуля. Данная функция определяется битом CYCLES регистра TMR\_CTRL. Также во время работы допускается перезапись текущего периода счета таймера. В результате перезаписи текущее значение таймера будет сброшено в 0 и счет начнется заново. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM, TMR\_VALL.

Значение, до которого будет считать модуль, определяется как:

$$N_{clk} = PRD + 1,$$

где  $N_{clk}$  – количество тактов системной частоты.

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус окончания счета периода – бит END\_PRD. На основании данного статуса возможно возникновение прерывания. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Простой таймер»:

- в регистры TMR\_PRDH, TMR\_PRDM и TMR\_PRDL записать требуемое значение периода счета таймера;
- в регистре TMR\_MSK при необходимости разрешить прерывание;
- в регистре TMR\_CTRL:
  - установить требуемое значение бита CYCLES;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 00b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущее значение таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

### Работа в режиме «Таймер с внешней остановкой»

#### Принцип работы

В режиме «Таймер с внешней остановкой» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. При запуске таймер считает с нуля до момента возникновения заданного события на соответствующем выводе GPIO (альтернативная функция I\_TIMER\_EXT). Тип события остановки определяется битом STOP\_TYPE регистра TMR\_CFG. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM и TMR\_VALL.

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус переполнения таймера – бит OVW и статус остановки таймера по внешнему событию – бит STOP\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Таймер с внешней остановкой»:

- в регистре TMR\_CFG задать тип события остановки счета таймера путем записи требуемого значения в бит STOP\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 01b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущие значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Работа в режиме «Межсобытийный таймер»

### Принцип работы

В режиме «Межсобытийный таймер» модуль представляет собой 24-разрядный таймер с инкрементацией каждый такт системной частоты. После разрешения работы таймер ожидает событие старта на соответствующем выводе GPIO (альтернативная функция I\_TIMER\_EXT). Тип события старта определяется битом START\_TYPE регистра TMR\_CFG. Окончанием счета является событие остановки счета таймера на выводе GPIO. Тип события остановки определяется битом STOP\_TYPE регистра TMR\_CFG. Текущее значение таймера находится в регистрах TMR\_VALH, TMR\_VALM и TMR\_VALL.

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме вырабатывается статус переполнения таймера – бит OVW, статус запуска таймера по внешнему событию – бит START\_EVENT и статус остановки таймера по внешнему событию – бит STOP\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Межсобытийный таймер»:

- в регистре TMR\_CFG задать тип события запуска и остановки счета таймера путем записи требуемого значения в биты START\_TYPE и STOP\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 10b;
  - установить бит EN\_EXT в «0»;
  - установить бит EN в «1».
- текущее значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

## Работа в режиме «Таймер-счетчик»

**Примечание:** данный режим работы поддерживают модули TIMER0 и TIMER1. Модуль TIMER2 данный режим работы не поддерживает. При запуске TIMER2 в режиме «Таймер-счетчик» модуль уйдет в состояние ожидания, выход из которого возможен только по отключению модуля.

### Принцип работы

В режиме «Таймер-счетчик» модули взаимодействуют между собой. Один из модулей необходимо настроить в режим таймера, а второй в режим счетчика. На вход модуля, который работает в режиме счетчика может быть подана либо частота внутреннего RC-генератора, либо сигнал с соответствующего вывода GPIO (альтернативная функция I\_TIMER\_EXT). Источник сигнала, события на котором будут отслеживаться, определяется битом EVENT\_TYPE регистра TMR\_CFG модуля счетчика. Тип события старта задается в обоих модулях битом START\_TYPE регистра TMR\_CFG.

После настройки модуль в режиме таймера и модуль в режиме счетчика ожидают события старта от ранее выбранного источника. Затем модуль, работающий в режиме таймера, считает до момента остановки его модулем в режиме счетчика. Модуль в режиме счетчика считает до значения, записанного в регистры периода счета TMR\_PRDH, TMR\_PRDM и TMR\_PRDL. Счетчик увеличивается на 1 каждый раз, когда фиксирует заданное событие. Тип события определяется битом FIX\_TYPE регистра TMR\_CFG. По завершению счета модуль в режиме счетчика формирует событие окончания счета для модуля, работающего в режиме таймера. В результате оба модуля прекращают счет.

Перезаписывать значение TMR\_PRDH, TMR\_PRDM и TMR\_PRDL для модуля в режиме счетчика в ходе работы запрещено.

При START\_TYPE = FIX\_TYPE значение, до которого будет считать счетчик, определяется как:

$$N_{event} = PRD + 1,$$

где  $N_{event}$  – количество событий.

При START\_TYPE != FIX\_TYPE:

$$N_{event} = PRD.$$

### Статусы и прерывания

Статусы содержатся в регистре TMR\_ST. В данном режиме, для модуля, работающего в режиме таймера, вырабатывается статус переполнения таймера – бит OVW, статус запуска таймера по внешнему событию – бит START\_EVENT и статус остановки таймера по внешнему событию – бит STOP\_EVENT. Для модуля, работающего в режиме счетчика, вырабатывается статус запуска таймера по внешнему событию – бит START\_EVENT. На основании данных статусов возможно возникновение прерываний. Прерывание разрешается путем записи «1» в соответствующий бит регистра маски прерываний TMR\_MSK.

### Алгоритм работы

Процедура настройки режима «Таймер-счетчик» для модуля в режиме таймера:

- в регистре TMR\_CFG задать тип события запуска счета таймера путем записи требуемого значения в бит START\_TYPE;
- в регистре TMR\_MSK разрешить необходимые прерывания;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «0»;
  - в биты MODE необходимо записать значение 11b;
  - установить бит EN\_EXT в «1»;
  - установить бит EN в «1».
- текущее значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

Бит EN\_EXT разрешает соседнему модулю начать работу, таким образом синхронизируя запуск обоих модулей.

Процедура настройки режима «Таймер-счетчик» для модуля в режиме счетчика:

- в регистре TMR\_CFG:
  - задать источник сигнала битом EVENT\_TYPE;
  - задать тип события счетчика битом FIX\_TYPE;
  - задать тип события запуска счета битом START\_TYPE.
- в регистры TMR\_PRDH, TMR\_PRDM и TMR\_PRDL записать требуемое значение периода счета таймера;
- в регистре TMR\_MSK при необходимости разрешить прерывание;
- в регистре TMR\_CTRL:
  - установить бит CYCLES в «0»;
  - установить бит T/C в «1»;
  - в биты MODE необходимо записать значение 11b;
  - установить бит EN\_EXT в «1»;
  - установить бит EN в «1».
- текущее значения таймера можно вычитать из регистров TMR\_VALH, TMR\_VALM и TMR\_VALL.

Бит EN\_EXT разрешает соседнему модулю начать работу, таким образом синхронизируя запуск обоих модулей.

## Регистры таймеров

№	Аббревиатура	Доступ	Описание
<b>TIMER0</b>			
2700h	TMR0_CTRL	RW	Регистр управления
2701h	TMR0_CFG	RW	Регистр конфигурации
2702h	TMR0_PRDH	RW	Период счета таймера, старшая часть
2703h	TMR0_PRDM	RW	Период счета таймера, средняя часть
2704h	TMR0_PRDL	RW	Период счета таймера, младшая часть
2705h	TMR0_VALH	R	Текущее значение таймера, старшая часть
2706h	TMR0_VALM	R	Текущее значение таймера, средняя часть
2707h	TMR0_VALL	R	Текущее значение таймера, младшая часть
2708h	TMR0_MSK	RW	Регистр маски прерываний
2709h	TMR0_ST	R	Регистр статусов
<b>TIMER1</b>			
2720h	TMR1_CTRL	RW	Регистр управления
2721h	TMR1_CFG	RW	Регистр конфигурации
2722h	TMR1_PRDH	RW	Период счета таймера, старшая часть
2723h	TMR1_PRDM	RW	Период счета таймера, средняя часть
2724h	TMR1_PRDL	RW	Период счета таймера, младшая часть
2725h	TMR1_VALH	R	Текущее значение таймера, старшая часть
2726h	TMR1_VALM	R	Текущее значение таймера, средняя часть
2727h	TMR1_VALL	R	Текущее значение таймера, младшая часть
2728h	TMR1_MSK	RW	Регистр маски прерываний
2729h	TMR1_ST	R	Регистр статусов
<b>TIMER2</b>			
2740h	TMR2_CTRL	RW	Регистр управления
2741h	TMR2_CFG	RW	Регистр конфигурации
2742h	TMR2_PRDH	RW	Период счета таймера, старшая часть
2743h	TMR2_PRDM	RW	Период счета таймера, средняя часть
2744h	TMR2_PRDL	RW	Период счета таймера, младшая часть
2745h	TMR2_VALH	R	Текущее значение таймера, старшая часть
2746h	TMR2_VALM	R	Текущее значение таймера, средняя часть
2747h	TMR2_VALL	R	Текущее значение таймера, младшая часть
2748h	TMR2_MSK	RW	Регистр маски прерываний
2749h	TMR2_ST	R	Регистр статусов

## TMRx\_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв		CYCLES	T/C	MODE		EN_EXT	EN
Начальное значение	0							

**CYCLES** – работа таймера по достижению значения периода:

- 1 – повторный счет с нуля;
- 0 – остановка счета.

**T/C** – работа в режиме таймера или счетчика, имеет значение только при MODE «Таймер-счетчик»:

1 – режим счетчика;

0 – режим таймера.

**MODE** – режимы работы модуля:

11b – «Таймер-счетчик»;

10b – «Межсобытийный таймер»;

01b – «Таймер с внешней остановкой»;

00b – «Простой таймер».

**EN\_EXT** – разрешение работы соседнего модуля таймера, имеет значение только при MODE «Таймер-счетчик»:

1 – работа соседнего таймера разрешена;

0 – работа соседнего таймера запрещена.

При работе с регистрами модуля TIMER0 бит EN\_EXT управляет модулем TIMER1. При работе с регистрами модуля TIMER1 бит EN\_EXT управляет модулем TIMER0.

**EN** – разрешение работы таймера:

1 – таймер включен;

0 – таймер выключен.

#### TMRx\_CFG

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>				EVENT_TYPE	FIX_TYPE	STOP_TYPE	START_TYPE
Начальное значение	0							

**EVENT\_TYPE** – источник сигнала, на котором отслеживаются события в режиме «Таймер-счетчик»:

1 – вывод GPIO;

0 – внутренний RC-генератор.

**FIX\_TYPE** – тип события, которое считается счетчиком в режиме «Таймер-счетчик»:

1 – событие заднего фронта;

0 – событие переднего фронта.

**STOP\_TYPE** – тип события остановки счета таймера:

1 – остановка по заднему фронту;

0 – остановка по переднему фронту.

**START\_TYPE** – тип события старта счета таймера:

1 – старт по заднему фронту;

0 – старт по переднему фронту.

#### TMRx\_PRDH

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

**PRD** – период счета таймера, старшая часть.



**TMRx\_PRDM**

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

PRD – период счета таймера, средняя часть.

**TMRx\_PRDL**

Период счета таймера обновляется при записи в данный регистр.

Бит	7	6	5	4	3	2	1	0
Назначение	PRD							
Начальное значение	0							

PRD – период счета таймера, младшая часть.

**TMRx\_VALH**

Бит	7	6	5	4	3	2	1	0
Назначение	VAL							
Начальное значение	0							

VAL – текущее значение таймера, старшая часть.

**TMRx\_VALM**

Бит	7	6	5	4	3	2	1	0
Назначение	VAL							
Начальное значение	0							

VAL – текущее значение таймера, средняя часть.

**TMRx\_VALL**

Бит	7	6	5	4	3	2	1	0
Назначение	VAL							
Начальное значение	0							

VAL – текущее значение таймера, младшая часть.

**TMRx\_MSK**

Возможно формирование прерывания по любому биту статусного регистра TMR\_ST. Расположение битов в TMR\_ST и TMR\_MSK аналогично.

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>		STOP_EVENT		START_EVENT		OVW	END_PRD
Начальное значение	0							

Для каждого из битов справедливо:

- 1 – данное прерывание формируется;
- 0 – данное прерывание не формируется.

## TMRx\_ST

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>				STOP_EVENT	START_EVENT	OVW	END_PRD
Тип статуса	EVENT							
Начальное значение	0							

**STOP\_EVENT** – зафиксировано событие остановки таймера;

**START\_EVENT** – зафиксировано событие старта таймера;

**OVW** – зафиксировано переполнение таймера;

**END\_PRD** – зафиксирован конец счета периода.

## Контроллер прерываний (INT\_CTRL)

При выполнении функции, вызванной прерыванием, необходимо произвести сброс данного прерывания.

### Регистры «Контроллера прерываний»

№	Аббревиатура	Доступ	Описание
2800h	INT_FIX_CLR0	RW	Регистр зафиксированных прерываний, группа 0
2801h	INT_FIX_CLR1	RW	Регистр зафиксированных прерываний, группа 1
2802h	INT_FIX_CLR2	RW	Регистр зафиксированных прерываний, группа 2
2803h	INT_FIX_CLR3	RW	Регистр зафиксированных прерываний, группа 3

#### INT\_FIX\_CLR0

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв				I3_FIX	Резерв	I1_FIX	I0_FIX
Начальное значение	0							

**I3\_FIX** – прерывание CMM.

**I1\_FIX** – прерывание WDT.

**I0\_FIX** – прерывание WORK\_FSM.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR0:

1 – сбросить прерывание;

0 – не менять текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR0:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

#### INT\_FIX\_CLR1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв					I2_FIX	I1_FIX	I0_FIX
Начальное значение	0							

**I2\_FIX** – прерывание UART.

**I1\_FIX** – прерывание TIMER0.

**I0\_FIX** – прерывание GPIO.

Запись в **Ix\_FIX** регистра INT\_FIX\_CLR1:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix\_FIX** регистра INT\_FIX\_CLR1:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

## INT\_FIX\_CLR2

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>							IO_FIX
Начальное значение	0							

**IO\_FIX** – прерывание TIMER1.

Запись в **IO\_FIX** регистра INT\_FIX\_CLR2:

- 1 – сбросить прерывание;
- 0 – не меняет текущую настройку.

Чтение **IO\_FIX** регистра INT\_FIX\_CLR2:

- 1 – зафиксировано прерывание;
- 0 – прерывание отсутствует.

## INT\_FIX\_CLR3

Бит	7	6	5	4	3	2	1	0
Назначение	<i>Резерв</i>							IO_FIX
Начальное значение	0							

**IO\_FIX** – прерывание TIMER2.

Запись в **IO\_FIX** регистра INT\_FIX\_CLR3:

- 1 – сбросить прерывание;
- 0 – не меняет текущую настройку.

Чтение **IO\_FIX** регистра INT\_FIX\_CLR3:

- 1 – зафиксировано прерывание;
- 0 – прерывание отсутствует.

## Режим «SLEEP»

### Общая информация

Микроконтроллер имеет возможность переходить в режим пониженного энергопотребления. Существует два различных режима «SLEEP»:

- «Сон процессора»;
- «Глубокий сон».

### Сон процессора

Для перевода системы в режим «Сон процессора» необходимо записать бит CPU\_SLEEP\_EN в регистр PMM\_CTRL модуля PMM при отсутствии активных прерываний в регистре INT\_FIX модуля «Контроллер прерываний». При этом с процессора будет снята синхрочастота, до момента, пока контроллер прерываний не зафиксирует одно из прерываний системы. При возникновении прерывания подача синхросигнала на процессор 8051 будет возобновлена.

### Глубокий сон

Для перевода системы в режим «Глубокий сон» необходимо записать бит EN в регистр FSM\_CTRL модуля «Рабочий автомат». При этом произойдет запуск модуля «Рабочий автомат», и по началу счета система начнет переход в режим «Глубокий сон». Со всех модулей системы, кроме модулей «Рабочий автомат», СММ, РММ будет снята синхрочастота, ПЗУ перейдет в малопотребляющий режим. Выход из данного режима произойдет, когда таймер модуля «Рабочий автомат» окончит счет. При этом будет сформировано немаскируемое прерывание, и подача синхрочастоты возобновится на все модули системы. Также выход из данного режима может быть осуществлен по фронту сигнала на выводе заранее настроенного GPIO или по любому сигналу сброса.

## Режим «COLD\_SLEEP»

### Общая информация

Микроконтроллер имеет возможность переходить в режим пониженного энергопотребления с полностью отключенными генераторами «COLD\_SLEEP» (Холодный сон).

### Холодный сон

Для перевода системы в режим «Холодный сон» необходимо записать бит EN в соответствующий регистр РММ. При этом разрешающие входы генераторов будут переключены в неактивное состояние. Выход из режима «Холодный сон» производится за счет формирования перехода из лог. «0» в лог. «1» на входе любой площадки, а сброс флага пробуждения произойдет после обратного события, перехода из лог. «1» в лог. «0».

## JTAG

### Общее описание

Микроконтроллер содержит модуль JTAG, который поддерживает обязательные и ряд дополнительных пользовательских инструкции.

Инструкция	Код инструкции	Разрядность регистра данных
EXTEST	000b	–
TEST_MODE	001b	–
SAMPLE/PRELOAD	010b	–
JTAG_RAM_ACCESS	011b	25
JTAG_ROM_ACCESS	100b	25
IF_ACCESS	101b	19
DEBUGGER	110b	5
BYPASS	111b	1

### Структурная схема

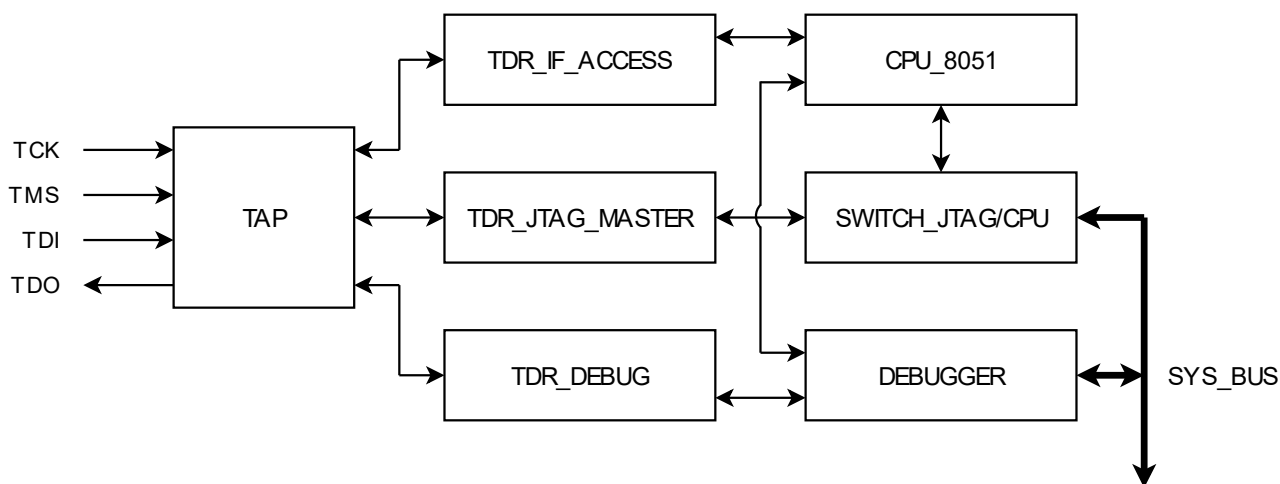


Рисунок 15. Структурная схема модуля JTAG с подключенными к нему блоками

На представленном выше рисунке изображены следующие блоки:

- TAP – управляющий рабочий автомат JTAG;
- TDR\_IF\_ACCESS – блок управления тестовым пользовательским регистром данных инструкции «IF\_ACCESS»;
- TDR\_JTAG\_MASTER – блок управления тестовым пользовательским регистром данных инструкций «JTAG\_RAM\_ACCESS» и «JTAG\_ROM\_ACCESS»;
- TDR\_DEBUG – блок управления тестовым пользовательским регистром данных инструкции «DEBUGGER»;
- CPU\_8051 – процессорное ядро;
- SWITCH\_CPU/JTAG – блок, дающий доступ к системной шине CPU или JTAG;
- DEBUGGER – отладчик.

**Инструкции «TEST\_MODE», «EXTEST», «BYPASS», «SAMPLE/PRELOAD»**

Инструкции «EXTEST» и «SAMPLE/PRELOAD» декодируются TAP-контроллером, но не имеют регистровых цепочек.

Инструкция «TEST\_MODE» имеет трехбитовый код 001b и не предполагает использование пользователем.

Инструкция «BYPASS» имеет трехбитовый код 111b и функционирует в соответствии со стандартом IEEE Std 1149.1-2001.

**Инструкция «JTAG\_RAM\_ACCESS»**

Инструкция «JTAG\_RAM\_ACCESS» представляет собой дополнительную пользовательскую инструкцию с кодом 011b, в результате выбора которой CPU 8051 будет отключен от системной шины, а в место него будет подключен тестовый пользовательский регистр данных разрядностью 25 бит:

<b>Бит</b>	–								<b>24</b>
<b>Назначение</b>	–								SYNCHR
<b>Начальное значение</b>	–								0
<b>Бит</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>	
<b>Назначение</b>	RW	CE_RAM	ADDR						
<b>Начальное значение</b>	0								
<b>Бит</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	
<b>Назначение</b>	ADDR								
<b>Начальное значение</b>	0								
<b>Бит</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	
<b>Назначение</b>	DATA								
<b>Начальное значение</b>	0								

**SYNCHR** – бит синхронизации данных (данные будут переданы, когда будет совершен переход из «0» в «1»):

- 1 – передать данные в системную шину;
- 0 – подготовка данных для передачи в системную шину.

**RW** – операция чтения или записи:

- 1 – прочитать данные;
- 0 – записать данные.

**CE\_RAM** – обращение в область памяти RAM:

- 1 – обращение в RAM-память или к ведомым устройствам системы;
- 0 – область RAM не задействуется.

**ADDR** – адрес обращения.

**DATA** – данные для записи. Младший бит данных находится в бите 7.

При чтении пользовательского тестового регистра данных, считанные с системной шины данные будут переданы со следующим за командой чтения пакетом данных в формате, представленном ниже:

Бит	–							24
Назначение	–							Резерв
Начальное значение	–							x
Бит	23	22	21	20	19	18	17	16
Назначение	Резерв							
Начальное значение	x							
Бит	15	14	13	12	11	10	9	8
Назначение	Резерв							
Начальное значение	x							
Бит	7	6	5	4	3	2	1	0
Назначение	DATA							
Начальное значение	0							

При этом во время чтения должна отправляться команда с CE\_RAM равным «0». Передача прочитанных данных осуществляется в соответствии со стандартом IEEE Std 1149.1-2001.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

#### Инструкция «JTAG\_ROM\_ACCESS»

Инструкция «JTAG\_ROM\_ACCESS» представляет собой дополнительную пользовательскую инструкцию с кодом 100b, в результате выбора которой CPU 8051 будет отключен от системной шины, а в место него будет подключен тестовый пользовательский регистр данных разрядностью 25 бит:

Бит	–							24
Назначение	–							SYNCHR
Начальное значение	–							x
Бит	23	22	21	20	19	18	17	16
Назначение	RW	CE_ROM	ADDR					
Начальное значение	0							
Бит	15	14	13	12	11	10	9	8
Назначение	ADDR							
Начальное значение	0							
Бит	7	6	5	4	3	2	1	0
Назначение	DATA							
Начальное значение	0							

**SYNCHR** – бит синхронизации данных (данные будут переданы, когда будет совершен переход из «0» в «1»):

- 1 – передать данные в системную шину;
- 0 – подготовка данных для передачи в системную шину.

**RW** – операция чтения или записи:

- 1 – прочитать данные;
- 0 – записать данные.



**CE\_ROM** – обращение в область памяти ROM:

1 – обращение в ROM-память;

0 – область ROM не задействуется.

**ADDR** – адрес обращения.

**DATA** – данные для записи. Младший бит данных находится в бите 7.

При чтении пользовательского тестового регистра данных, считанные с системной шины данные будут переданы со следующим за командой чтения пакетом данных в формате, представленном ниже:

<b>Бит</b>	–							<b>24</b>
<b>Назначение</b>	–							Резерв
<b>Начальное значение</b>	–							x
<b>Бит</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Назначение</b>	RW	CE_ROM	ADDR					
<b>Начальное значение</b>	0							
<b>Бит</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Назначение</b>	ADDR							
<b>Начальное значение</b>	0							
<b>Бит</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Назначение</b>	DATA							
<b>Начальное значение</b>	0							

При этом во время чтения должна отправляться команда с CE\_ROM равным «0». Передача прочитанных данных осуществляется в соответствии со стандартом IEEE Std 1149.1-2001.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

#### Инструкция «IF\_ACCESS»

Инструкция «IF\_ACCESS» («Интерфейс доступа») представляет собой дополнительную пользовательскую инструкцию с кодом 101b и пользовательским тестовым регистром данных разрядностью 19 бит.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

#### Инструкция «DEBUGGER»

Инструкция «DEBUGGER» («Отладчик») представляет собой дополнительную пользовательскую инструкцию с кодом 110b и пользовательским тестовым регистром данных разрядностью 5 бит.

Примечание: выход из инструкции не обнуляет пользовательский тестовый регистр данных.

