

Основные особенности

- Напряжение питания:
5,0 В \pm 10% – без использования
встроенного линейного регулятора;
7,0 В...15 В – при использовании
встроенного линейного регулятора;
- Ток потребления не более 20 мА;
- Микроконтроллерное ядро
архитектуры 8051:
– машинный цикл 1 такт.
- Встроенные аналоговые модули:
– 12 разрядный ЦАП;
– источник опорного напряжения
с масштабирующим ОУ;
– RC-генератор;
– супервизор питания;
– блок ФАПЧ;
– регуляторы напряжения.
- Режим энергосбережения («SLEEP»);
- Температурный диапазон
от -60°C до $+125^{\circ}\text{C}$;
- Технология изготовления КМОП КНИ;
- Стойкость к СВВФ.

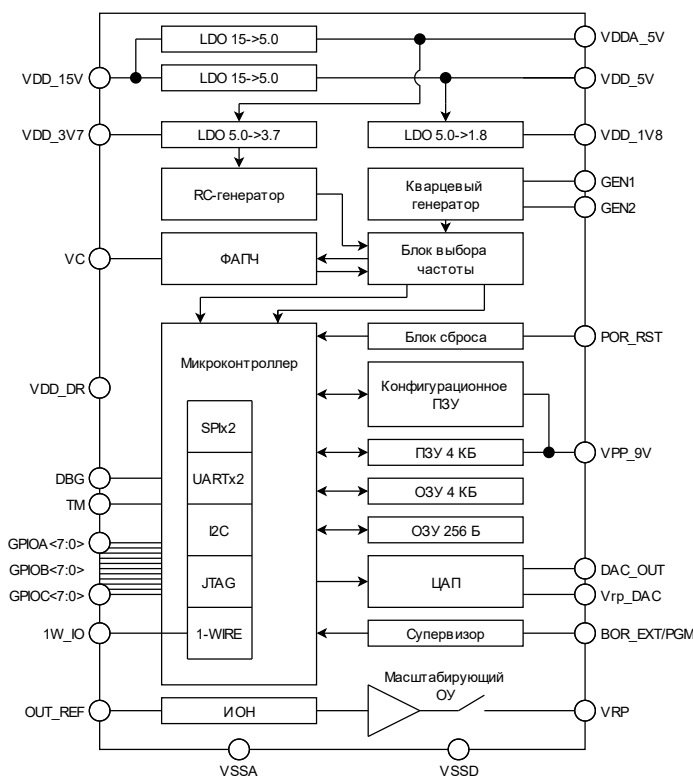


Рисунок 1. Структурная схема



ГГ – год выпуска
НН – неделя выпуска

Рисунок 2. Внешний вид
микросхемы 5400TP105-003

Общее описание

Микросхема 5400TP105-003 представляет собой малопотребляющий контроллер с архитектурой ядра 8051 для бортовых систем управления с рабочей частотой до 8,0 МГц. Микросхема выполнена на базе радиационно-стойкого аналого-цифрового БМК 5400TP10 по технологии КНИ.

Особенности микроконтроллера

- возможность выбора тактового сигнала (кварцевый генератор, RC-генератор, внешняя частота);
- возможность умножения тактовой частоты с помощью ФАПЧ с настраиваемым коэффициентом;
- возможность программирования выходной частоты RC-генератора;
- возможность программирования коэффициента усиления масштабирующего ОУ;
- 24 универсальных портов ввода/вывода (GPIO);
- опрос GPIO одной командой;
- настраиваемые прерывания по внешним событиям (с помощью GPIO).

Периферия

- 2 интерфейса SPI;
- 2 интерфейса UART;
- интерфейс 1-Wire;
- интерфейс I2C;
- интерфейс JTAG;
- 24 универсальные линии ввода/вывода с индивидуальной настройкой направления;
- три 24-разрядных таймера/счетчика;
- сторожевой таймер;
- модуль перевода системы в режим пониженного энергопотребления («SLEEP»).

Микросхема выполнена в 48-ми выводном металлокерамическом корпусе 5142.48-A.

Электрические параметры микросхемы

Таблица 1. Электрические характеристики (температурный диапазон от –60°C до +125°C)

Параметр, единица измерения	Норма параметра		
	не менее	типичное	не более
Частота тактирования ядра, МГц			8,0
Напряжение питания (VDD_15V), В при использовании встроенного регулятора для автоматического формирования напряжения питания ядра	7,0		15,75
Напряжение питания ядра, В	4,5	5,0	5,5
Напряжение питания интерфейсной части (VDD_DR), В	1,7	1,8 5,0	5,5
Выходное напряжение линейного регулятора, В			
15 В → 5,0 В (вывод VDD_5V, VDDA_5V)	4,5	5,0	5,5
5,0 В → 3,7 В (вывод VDD_3V7)	3,4	3,7	4,0
5,0 В → 1,8 В (вывод VDD_1V8)	1,62	1,8	1,98
Нижняя граница диапазона настройки частоты RC-генератора, кГц			90
Верхняя граница диапазона настройки частоты RC-генератора, кГц	400		
Ток потребления, мА			20
Ток в режиме низкого энергопотребления («SLEEP»), мА			1,0
Напряжение высокого уровня выходных цифровых сигналов, В при VDD_DR = 1,8 В при VDD_DR = 5,0 В	1,4 4,6	1,8 5,0	
Напряжение низкого уровня выходных цифровых сигналов, В при VDD_DR = 1,8 В при VDD_DR = 5,0 В		0 0	0,4 0,4
Дифференциальная нелинейность ЦАП, МЗР	–0,99		2,0
Интегральная нелинейность ЦАП, МЗР	–8,0		8,0
Выходное напряжение ИОН, В	0,92	1,0	1,08
Температурный дрейф ИОН, ppm/°C			100
Примечание: линейные регуляторы предназначены для формирования внутреннего питания, нагрузка выводов VDD_5V, VDD_3V7, VDD_1V8 недопустима.			

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) ⁽¹⁾ , В	7,0	15,75	−0,3	16,5
Аналоговое напряжение питания (VDDA_5V) ⁽²⁾ , В	4,5	5,5	−0,3	5,7
Цифровое напряжение питания (VDD_5V) ⁽²⁾ , В	4,5	5,5	−0,3	5,7
Напряжение питания интерфейсной части (VDD_DR), В	1,7	5,5	−0,3	5,7
Напряжение программирования ПЗУ (VPP_9V), В	8,5	9,0	−0,3	9,5
Напряжение внешнего опорного уровня ЦАП (V _{ref} _DAC), В	2,25	5,0	−0,3	5,7
Напряжение высокого уровня входных цифровых сигналов (GPIO в режиме входа, BOR_EXT/PGM, TM, DBG), В	VDD_DR −0,4	VDD_DR +0,3 ⁽³⁾	−0,3	VDD_DR +0,5 ⁽⁴⁾
Напряжение низкого уровня входных цифровых сигналов (GPIO в режиме входа, BOR_EXT/PGM, TM, DBG), В	0	0,4	−0,3	5,7
Температура эксплуатации, °C	−60	+125	−60	+150
Примечание: 1) При использовании линейного регулятора 15 В → 5,0 В и питание микросхемы обеспечивается подачей напряжения на вывод VDD_15V. 2) Если не используется линейный регулятор 15 В → 5,0 В и питание микросхемы обеспечивается подачей напряжения на вывод VDD_5V и VDDA_5V. 3) Не более 5,5 В. 4) Не более 5,7 В.				

Конфигурация и функциональное описание выводов

Таблица 3. Функциональное описание выводов

№ вывода	Наименование вывода	Назначение вывода
1	GPIOB<5>	Порт ввода-вывода микроконтроллера, разряд №5 группы В
2	GPIOB<6>	Порт ввода-вывода микроконтроллера, разряд №6 группы В
3	GPIOB<7>	Порт ввода-вывода микроконтроллера, разряд №7 группы В
4	VDD_5V	Вывод цифрового положительного напряжения питания (выходное напряжение LDO-регулятора 15 В → 5,0 В)
5	VC	Вывод блока ФАПЧ для подключения RC-фильтра
6	GEN1	Вход подключения кварцевого резонатора / вход для подачи внешней тактовой частоты
7	GEN2	Вход подключения кварцевого резонатора / выход контроля поданной внешней частоты
8	VSSD	Вывод цифрового отрицательного напряжения питания
9	GPIOC<0>/TCK	Порт ввода-вывода микроконтроллера, разряд №0 группы С / вход TCK интерфейса JTAG в тестовом режиме
10	GPIOC<1>/TMS	Порт ввода-вывода микроконтроллера, разряд №1 группы С / вход TMS интерфейса JTAG в тестовом режиме
11	GPIOC<2>/TDI	Порт ввода-вывода микроконтроллера, разряд №2 группы С / вход TDI интерфейса JTAG в тестовом режиме
12	GPIOC<3>/TDO	Порт ввода-вывода микроконтроллера, разряд №3 группы С / выход TDO интерфейса JTAG в тестовом режиме
13	GPIOC<4>	Порт ввода-вывода микроконтроллера, разряд №4 группы С
14	GPIOC<5>	Порт ввода-вывода микроконтроллера, разряд №5 группы С
15	GPIOC<6>	Порт ввода-вывода микроконтроллера, разряд №6 группы С
16	GPIOC<7>	Порт ввода-вывода микроконтроллера, разряд №7 группы С
17	VDD_1V8	Вывод напряжения питания ядра (выходное напряжение LDO-регулятора 5,0 В → 1,8 В)
18	VDD_DR	Вывод положительного напряжения питания универсальных портов ввода-вывода микроконтроллера 1,8 В – 5,0 В
19	1W_IO	Вывод интерфейса 1-Wire (тип вывода – открытый сток)
20	GPIOA<0>	Порт ввода-вывода микроконтроллера, разряд №0 группы А
21	GPIOA<1>	Порт ввода-вывода микроконтроллера, разряд №1 группы А
22	GPIOA<2>	Порт ввода-вывода микроконтроллера, разряд №2 группы А
23	GPIOA<3>	Порт ввода-вывода микроконтроллера, разряд №3 группы А
24	GPIOA<4>	Порт ввода-вывода микроконтроллера, разряд №4 группы А
25	GPIOA<5>	Порт ввода-вывода микроконтроллера, разряд №5 группы А
26	GPIOA<6>	Порт ввода-вывода микроконтроллера, разряд №6 группы А
27	GPIOA<7>	Порт ввода-вывода микроконтроллера, разряд №7 группы А
28	BOR_EXT / PGM	Вход внешнего супервизора питания / вход выбора режима программирования ПЗУ в тестовом режиме: лог. «1» – перевод в режим программирования, от шины отключается CPU и подключается JTAG; лог. «0» – стандартная работа.

№ вывода	Наименование вывода	Назначение вывода
29	TM	Вход для выбора режима работы микроконтроллера: лог. «1» – тестовый режим; лог. «0» – стандартная работа.
30	DBG	Вход для выбора блока памяти, выполняющего роль ПЗУ: лог. «1» – память ПЗУ (ROM_RAM_4KB) 4 КБ, ОЗУ 256 Б; лог. «0» – память ПЗУ (ROM_OTP_4KB) 4 КБ, ОЗУ 4 КБ + 256 Б.
31	POR_RST	Вход для подключения внешнего конденсатора блока сброса
32	Vrp_DAC	Вывод положительного опорного напряжения ЦАП
33	DAC_OUT	Выход ЦАП
34	VPP_9V	Вывод для программирования ПЗУ и конфигурационной памяти 9,0 В
35	VDDA_5V	Вывод аналогового положительного напряжения питания (выходное напряжение LDO-регулятора 15 В → 5,0 В)
36	VDD_3V7	Вывод положительного напряжения питания RC-генератора (выходное напряжение LDO-регулятора 5,0 В → 3,7 В)
37	VSSA	Вывод аналогового отрицательного напряжения питания
38	VRP	Выход масштабирующего ОУ (если ОУ включен в ходе настройки конфигурационной памяти)
39	OUT_REF	Выходное напряжение ИОН в тестовом режиме
40 – 42	Tech	Технологический вывод
43	VDD_15V	Вывод положительного напряжения питания 15 В
44	GPIOB<0>/H_S	Порт ввода-вывода микроконтроллера, разряд №0 группы В/ выбор режима конфигурационной памяти в тестовом режиме
45	GPIOB<1>/ RC_CLKOUT	Порт ввода-вывода микроконтроллера, разряд №1 группы В/ выходная частота RC-генератора в тестовом режиме
46	GPIOB<2>	Порт ввода-вывода микроконтроллера, разряд №2 группы В
47	GPIOB<3>	Порт ввода-вывода микроконтроллера, разряд №3 группы В
48	GPIOB<4>	Порт ввода-вывода микроконтроллера, разряд №4 группы В

Рекомендуемая схема применения

Таблица 4. Таблица внешних компонентов

Компонент	Номинал	Компонент	Номинал
R1	1 – 2 МОм	C3	10 нФ
R2	2 – 20 кОм	C4	1 нФ
R3	100 Ом	C5	20 нФ
R4	4,7 кОм	C6 – C13	100 – 200 нФ
C1, C2	16 – 64 пФ		
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.

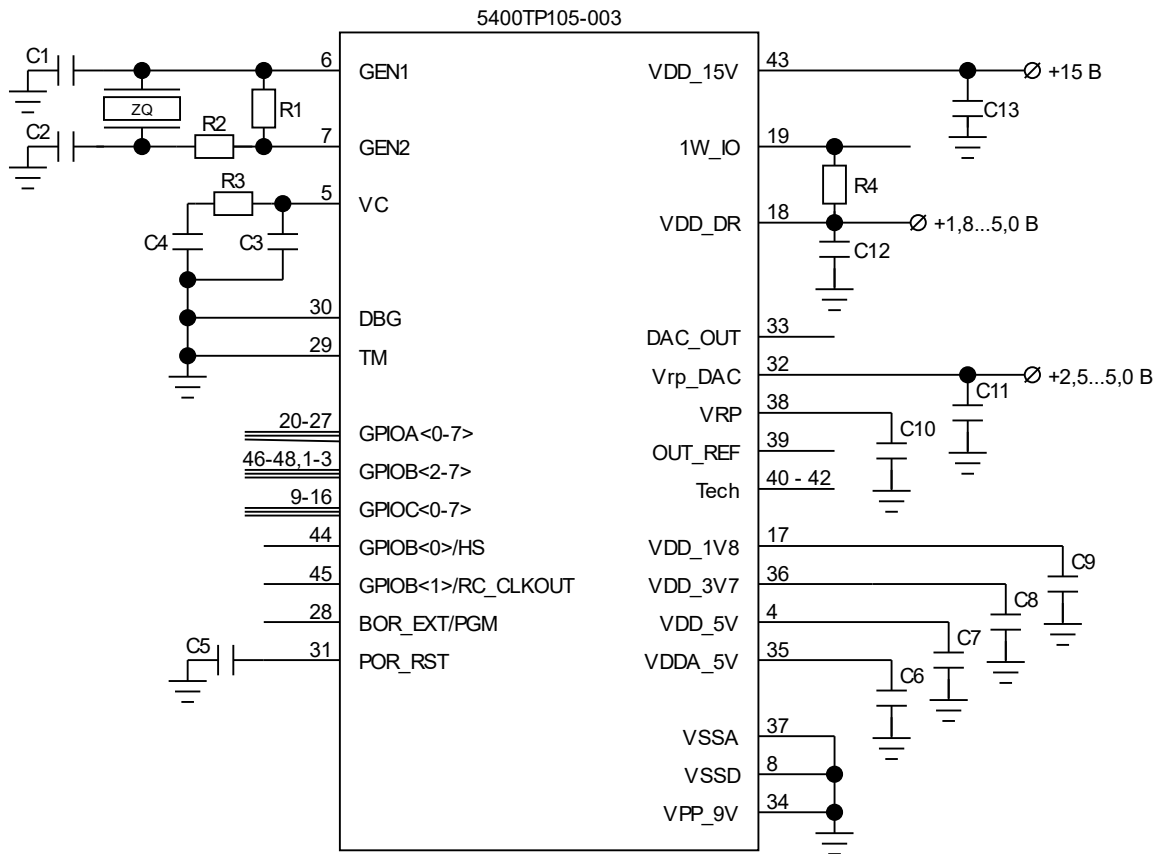


Рисунок 3. Рекомендуемая схема применения после программирования микросхемы в ПЗУ (при использовании встроенного линейного регулятора напряжения)

Примечание:

Если ЦАП не используется, то вывод Vrp_DAC (32) подключить к VSSA (37).

Если не используется 1-Wire интерфейс, то вывод 1W_IO (19) необходимо оставить в обрыве.

При использовании внешнего генератора, вывод GEN2 (7) необходимо оставить в обрыве.

При организации питания микроконтроллера от 5,0 В необходимо подавать напряжение 5,0 В на выводы VDD_15V (43), VDD_5V (4), VDDA_5V (35).

Если ФАПЧ не используется, то вывод VC (5) подключить к VSSA (37).

Technical drawing of a square frame assembly, labeled 'A'.

Top View (Left):

- Overall width: $12,7 \pm 0,127$
- Inner square frame dimensions: 41, 31, 29, 19, 17, 7, 5, 1, 48, 43.
- Label: 4 паза R0,3 (4 slots R0,3)
- Label: 44 паза R0,2 (44 slots R0,2)
- Label: Ключ (Key)

Side View (Right):

- Overall height: $2,124 \pm 0,18$
- Inner square frame height: $1,524 \pm 0,180$
- Label: 4,8 пазовок 0,4 (4,8 slots 0,4)
- Label: $\pm 0,38$
- Label: $12,7 \pm 0,127$
- Label: 1,016
- Label: 48 пазовок (48 slots)

Bottom View (Below):

- Overall width: $10 \times 1,016 = 10,16$
- Inner square frame width: $10 \times 1,016 = 10,16$
- Label: 1,016
- Label: 31
- Label: 41
- Label: 43
- Label: 29
- Label: 48
- Label: 1
- Label: 10,16
- Label: 1,016
- Label: 5
- Label: 7
- Label: 0,89 $\pm 0,20$
- Label: 0,508
- Label: 48 пазовок (48 slots)
- Label: 44 пазовок (44 slots)
- Label: 4,8 пазовок (4,8 slots)
- Label: 0,71
- Label: 19
- Label: 17

Информация для заказа

Обозначение	Маркировка	Корпус	Температурный диапазон
5400TP105-003 АЕНВ.431260.544ТУ карта заказа КФЦС.431260.012-003Д16	5400TP105-003	5142.48-A	-60°С...+125°С
К5400TP105-003 КФЦС.431000.001ТУ карта заказа КФЦС 431260.012-003.01Д16	К5400TP105-003	5142.48-A	-60°С...+125°С

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

Оглавление

Основные особенности.....	1
Общее описание	1
Особенности микроконтроллера	1
Периферия	2
Электрические параметры микросхемы.....	2
Электростатическая защита.....	3
Предельно-допустимые и предельные режимы эксплуатации	3
Конфигурация и функциональное описание выводов.....	4
Рекомендуемая схема применения.....	6
Габаритный чертеж.....	7
Информация для заказа	7
Техническое описание	12
Блок схема.....	12
Состав изделия.....	13
Конфигурация выводов, альтернативные функции.....	14
Карта памяти.....	17
Процессор 8051.....	18
Общие сведения	18
Поддержка IDE.....	18
Скорость выполнения команд	18
Прерывания.....	20
Подприоритеты прерываний.....	20
Регистры управления прерываниями	21
Ручной вызов прерываний	22
Таймер-счетчик.....	22
Блок управления внешними регистрами	23
Обращение к внешним регистрам.....	23
Типы статусов и прерывания	25
Система управления сбросом и питанием (PMM)	26
Структурная схема.....	26
Формирование сброса.....	26
Регистры модуля системы управления сбросом и питанием.....	27
Модуль управления источниками тактовых сигналов системы (CMM)	28
Общая информация	28
Структурная схема.....	28
Выбор источника тактирования системы	28
Деление частоты тактирования.....	29

Регистры модуля управления источниками тактовых сигналов системы	29
Сторожевой таймер (WDT).....	31
Общая информация	31
Регистры сторожевого таймера.....	31
GPIO.....	35
Общая информация	35
Структурная схема.....	35
Статусы и прерывания	36
Регистры GPIO	36
SPI.....	40
Общая информация	40
Структурная схема.....	40
Алгоритмы работы	41
Дуплексный и симплексный режимы	42
Передача данных.....	42
Прием данных	42
Работа с входом выбора микросхемы	43
Особенности работы в режиме «ведомый». Синхронизация и фильтрация	43
Регистры SPI	44
UART	49
Общая информация	49
Структурная схема.....	49
Делитель частоты	50
Высокоскоростной режим	50
Приемник	51
Буфер приемника.....	51
Передачик.....	52
Буфер передатчика	52
Прерывания.....	52
Таймер тайм-аута	52
Генерация и распознавание сигнала break	52
Особые режимы работы.....	53
Аппаратный контроль обмена	54
Регистры UART	54
I2C	63
Общая информация	63
Структурная схема.....	63
Алгоритмы работы	64

Статусы и прерывания	66
Регистры I2C	66
1-Wire	73
Общая информация	73
Структурная схема	73
Сигнализация шины 1-Wire	74
Предварительная настройка	76
Генерация цикла сброса/обнаружения присутствия	76
Передача байта	76
Прием байта	76
Битовый режим	77
Ускоренная передача	77
Режим ускоренного поиска ПЗУ (Search ROM Accelerator)	77
Пример использования режима ускоренного поиска ПЗУ	78
Статусы и прерывания	79
Регистры OWI	80
Рабочий автомат (WORK_FSM)	84
Общая информация	84
Особенности работы	84
Регистры «Рабочего автомата»	84
Таймер	86
Общая информация	86
Структурная схема	86
Работа в режиме «Простой таймер»	86
Принцип работы	86
Статусы и прерывания	87
Алгоритм работы	87
Работа в режиме «Таймер с внешней остановкой»	87
Принцип работы	87
Статусы и прерывания	87
Алгоритм работы	87
Работа в режиме «Межсобытийный таймер»	88
Принцип работы	88
Статусы и прерывания	88
Алгоритм работы	88
Работа в режиме «Таймер-счетчик»	88
Принцип работы	88
Статусы и прерывания	89

Алгоритм работы	89
Регистры таймеров	90
ЦАП	93
Общая информация	93
Алгоритм работы	93
Регистры модуля «ЦАП»	93
Контроллер прерываний (INT_CTRL)	94
Регистры «Контроллера прерываний»	94
Режим «SLEEP»	96
Общая информация	96
Сон процессора	96
Глубокий сон	96
JTAG	97
Общее описание	97
Структурная схема	97
Инструкции «TEST_MODE», «EXTEST», «BYPASS», «SAMPLE/PRELOAD»	98
Инструкция «JTAG_RAM_ACCESS»	98
Инструкция «JTAG_ROM_ACCESS»	99
Инструкция «IF_ACCESS»	100
Инструкция «DEBUGGER»	100
ANALOG_CFG	101
Регистры модуля ANALOG_CFG	101
Лист регистрации изменений	104

Блок схема

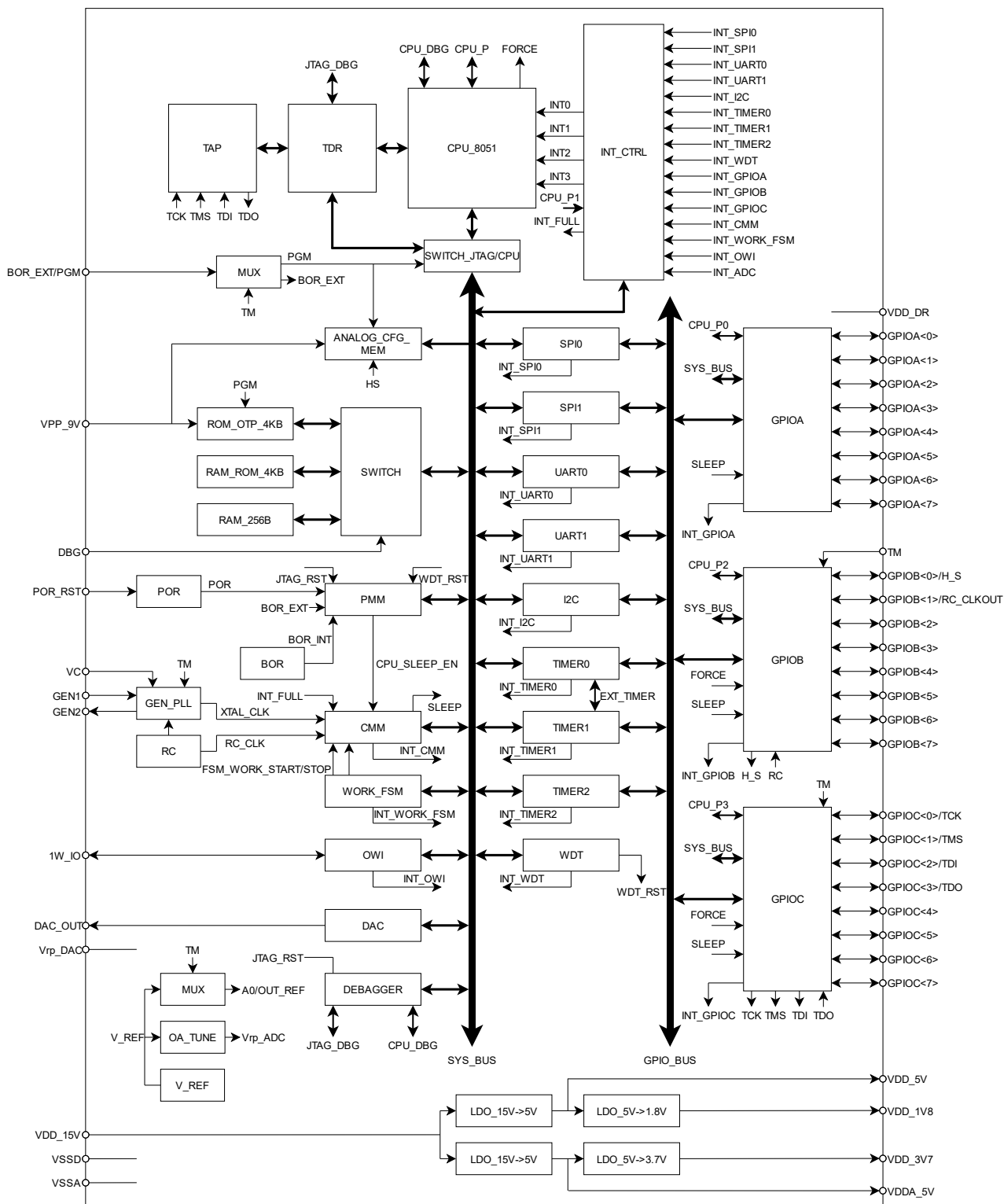


Рисунок 5. Блок схема 5400ТР105-003

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_4KB (One-Time Programmable) – однократно программируемое постоянное запоминающее устройство емкостью 4 КБ, память программ устройства;
- RAM_ROM_4KB – оперативное запоминающее устройство емкостью 4 КБ, микроконтроллер работает с данным модулем, как с памятью программ или памятью данных в зависимости от значения вывода DBG;
- RAM_256B – оперативное запоминающее устройство емкостью 256B, для хранения промежуточных данных, память данных устройства;
- SWITCH – модуль, разграничивающий доступ к RAM_OTP_4KB, RAM_ROM_4KB, RAM_256B;
- SWITCH JTAG/CPU – модуль, разграничивающий доступ к шине для JTAG и CPU;
- POR (Power-On Reset) – модуль сброса при включении;
- BOR (Brown-Out Reset) – модуль сброса при снижении напряжения питания;
- PMM (Power Management Module) – модуль системы управления сбросом и питанием;
- GEN_PLL – модуль ввода-вывода и фазовой автоподстройки частоты (ФАПЧ) тактирования;
- RC – RC-генератор;
- CMM (Clock Management Module) – модуль управления источниками тактовых сигналов системы;
- DEBUGGER – отладчик;
- OA_TUNE – масштабирующий операционный усилитель (МОУ);
- V_REF – источник опорного напряжения (ИОН);
- LDO_15V->5V, LDO_5V->3V7, LDO_5V->1V8 – линейные регуляторы напряжения.

Периферийные устройства:

- INT_CTRL – контроллер прерываний;
- SPI0, SPI1 – модули SPI;
- UART0, UART1 – модули UART;
- I2C – модуль I2C;
- GPIOA, GPIOB, GPIOC – контроллеры портов ввода-вывода;
- TIMER0, TIMER1, TIMER2 – универсальные таймеры;
- WDT – сторожевой таймер;
- WORK_FSM – рабочий автомат для режима «SLEEP»;
- OWI – модуль 1-Wire;
- DAC – цифро-аналоговый преобразователь.

Конфигурация выводов, альтернативные функции

Таблица 5. Альтернативные функции выводов GPIO

№ вывода	Наименование вывода	Альтернативная функция			Пояснение
		АФ0	АФ1	АФ2	
Порт А					
20	GPIOA<0>	SPI0_MOSI	SPI1_MOSI	I_TIMER0_EXT	Порт ввода-вывода микроконтроллера, разряд №0 группы А: <ul style="list-style-type: none">• SPI0/1 – MOSI (направление определяется режимом работы «ведущий»/«ведомый»);• TIMER0 – I_TIMER0_EXT (вход).
21	GPIOA<1>	SPI0_MISO	SPI1_MISO	I_TIMER1_EXT	Порт ввода-вывода микроконтроллера, разряд №1 группы А: <ul style="list-style-type: none">• SPI0/1 – SCK (направление определяется режимом работы «ведущий»/«ведомый»);• TIMER1 – I_TIMER1_EXT (вход).
22	GPIOA<2>	SPI0_SCK	SPI1_SCK	O_SLEEP	Порт ввода-вывода микроконтроллера, разряд №2 группы А: <ul style="list-style-type: none">• SPI0/1 – SCK (направление определяется режимом работы «ведущий»/«ведомый»);• режим «Глубокий сон» – O_SLEEP (выход).
23	GPIOA<3>	SPI0_I_CS	SPI1_I_CS	SPI0_O_CS	Порт ввода-вывода микроконтроллера, разряд №3 группы А: <ul style="list-style-type: none">• SPI0/1 – I_CS (вход);• SPI0 – O_CS (выход).
24	GPIOA<4>	UART0_TX	UART1_TX	SPI1_O_CS	Порт ввода-вывода микроконтроллера, разряд №4 группы А: <ul style="list-style-type: none">• UART0/1 – TX (выход);• SPI1 – O_CS (выход).
25	GPIOA<5>	UART0_RX	UART1_RX	«0»	Порт ввода-вывода микроконтроллера, разряд №5 группы А: <ul style="list-style-type: none">• UART0/1 – RX (вход);• лог. «0» (выход).
26	GPIOA<6>	I2C_SCL	UART0_CTS	UART1_CTS	Порт ввода-вывода микроконтроллера, разряд №6 группы А: <ul style="list-style-type: none">• I2C – SCL (направление определяется режимом работы «ведущий»/«ведомый»);• UART0/1 – CTS (вход).
27	GPIOA<7>	I2C_SDA	UART0_RTS	UART1_RTS	Порт ввода-вывода микроконтроллера, разряд №7 группы А: <ul style="list-style-type: none">• I2C – SDA (направление определяется протоколом);• UART0/1 – RTS (выход).
Порт В					
44	GPIOB<0>/H_S	SPI0_MOSI	SPI1_MOSI	I_TIMER0_EXT	Порт ввода-вывода микроконтроллера, разряд №0 группы В; при TM = 1 вывод принудительно работает как H/S (вход); во время обращения к внешним регистрам этот вывод работает как DATA<0> (направление определяется командой)
45	GPIOB<1>/RC_CLKOUT	SPI0_MISO	SPI1_MISO	I_TIMER2_EXT	Порт ввода-вывода микроконтроллера, разряд №1 группы В; при TM = 1 вывод принудительно работает как RC_CLKOUT (выход); во время обращения к внешним регистрам этот вывод работает как DATA<1> (направление определяется командой)
46	GPIOB<2>	SPI0_SCK	SPI1_SCK	O_SLEEP	Порт ввода-вывода микроконтроллера, разряд №2 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<2> (направление определяется командой)

№ вывода	Наименование вывода	Альтернативная функция			Пояснение
		АФ0	АФ1	АФ2	
47	GPIOB<3>	SPI0_I_CS	SPI1_I_CS	SPI0_O_CS	Порт ввода-вывода микроконтроллера, разряд №3 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<3> (направление определяется командой)
48	GPIOB<4>	UART0_TX	UART1_TX	SPI1_O_CS	Порт ввода-вывода микроконтроллера, разряд №4 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<4> (направление определяется командой)
1	GPIOB<5>	UART0_RX	UART1_RX	«0»	Порт ввода-вывода микроконтроллера, разряд №5 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<5> (направление определяется командой)
2	GPIOB<6>	I2C_SCL	UART0_CTS	UART1_CTS	Порт ввода-вывода микроконтроллера, разряд №6 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<6> (направление определяется командой)
3	GPIOB<7>	I2C_SDA	UART0_RTS	UART1_RTS	Порт ввода-вывода микроконтроллера, разряд №7 группы В; во время обращения к внешним регистрам этот вывод работает как DATA<7> (направление определяется командой)
Порт С					
9	GPIOC<0>/TCK	SPI0_MOSI	SPI1_MOSI	I_TIMER1_EXT	Порт ввода-вывода микроконтроллера, разряд №0 группы С; при ТМ = 1 вывод принудительно работает как TCK (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как WR/RD (выход)
10	GPIOC<1>/TMS	SPI0_MISO	SPI1_MISO	I_TIMER2_EXT	Порт ввода-вывода микроконтроллера, разряд №1 группы С; при ТМ = 1 вывод принудительно работает как TMS (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как EN (выход)
11	GPIOC<2>/TDI	SPI0_SCK	SPI1_SCK	O_SLEEP	Порт ввода-вывода микроконтроллера, разряд №2 группы С; при ТМ = 1 вывод принудительно работает как TDI (вход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как SEL<0> (выход)
12	GPIOC<3>/TDO	SPI0_I_CS	SPI1_I_CS	SPI0_O_CS	Порт ввода-вывода микроконтроллера, разряд №3 группы С; при ТМ = 1 вывод принудительно работает как TDO (выход) интерфейса JTAG; во время обращения к внешним регистрам этот вывод работает как SEL<1> (выход)
13	GPIOC<4>	UART0_TX	UART1_TX	SPI1_O_CS	Порт ввода-вывода микроконтроллера, разряд №4 группы С
14	GPIOC<5>	UART0_RX	UART1_RX	«0»	Порт ввода-вывода микроконтроллера, разряд №5 группы С
15	GPIOC<6>	I2C_SCL	UART0_CTS	UART1_CTS	Порт ввода-вывода микроконтроллера, разряд №6 группы С; во время обращения к внешним регистрам этот вывод работает как SEL<2> (выход)

№ вывода	Наименование вывода	Альтернативная функция			Пояснение
		АФ0	АФ1	АФ2	
16	GPIOC<7>	I2C_SDA	UART0_RTS	UART1_RTS	Порт ввода-вывода микроконтроллера, разряд №7 группы C; во время обращения к внешним регистрам этот вывод работает как SEL<3> (выход)

Порты JTAG (TCK, TDI, TMS, TDO) мультиплексированы с портами GPIOC. Выбор назначения выводов (GPIOC или JTAG) осуществляется выводом TM.

Порты интерфейсов (SPI, UART и т.д.) также мультиплексированы с портами GPIO. Выбор назначения выводов осуществляется с помощью альтернативных функций во время работы микроконтроллера.

Карта памяти

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

Таблица 6. Адресное пространство памяти программ и данных

№	Выделенное пространство	Зона	Аббревиатура	Описание
0000h	4 КБ	ПЗУ	ROM	ПЗУ
0000h	4352 Б	ОЗУ	RAM	ОЗУ
2000h	256 Б		PMM	Подсистема управления электропитанием
2100h	256 Б		CMM	Подсистема управления тактированием
2200h	256 Б		WDT	Сторожевой таймер
2300h	256 Б		GPIOA	Универсальные порты ввода-вывода
2400h	256 Б		GPIOB	
2500h	256 Б		GPIOC	
2600h	256 Б		SPI0	Интерфейсы SPI
2700h	256 Б		SPI1	
2800h	256 Б		UART0	Интерфейсы UART
2900h	256 Б		UART1	
2A00h	256 Б		I2C	Интерфейс I2C
2B00h	256 Б		OWI	Интерфейс 1-WIRE
2C00h	256 Б		WORK_FSM	Рабочий автомат
2D00h	32 Б		TIMER0	Таймеры
2D20h	32 Б		TIMER1	
2D40h	32 Б		TIMER2	
3000h	256 Б		DAC	Цифро-аналоговый преобразователь
3100h	256 Б		INT_CTRL	Контроллер прерываний
3200h	64 Б		DEBUGGER	Отладчик
3300h	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 7. 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 8. Описание прерываний и адрес перехода

Номер прерывания	Описание	Адрес перехода	Внутреннее название прерываний
0	WDT	0003h	IE0
	WORK_FSM		
	CMM		
1	Внутренний таймер/счетчик T0	000Bh	TF0
2	I2C	0013h	IE1
	UART0		
	SPI0		
	TIMER0		
	GPIOA		
3	Внутренний таймер/счетчик T1	001Bh	TF1
4	OWI	0023h	IS0
	UART1		
	SPI1		
	TIMER1		
	GPIOB		
5	GPIOC	002Bh	IS1
	TIMER2		

Прерывания IS0 и IS1 всегда работают по уровню.

Подприоритеты прерываний

При одинаковом приоритете обработка прерываний выполняется в следующей последовательности

Таблица 9. 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	Резерв			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.

Блок управления внешними регистрами

Блок управления внешними регистрами предназначен для быстрого доступа к 16 внешним 8-разрядным регистрам. Для доступа к этим регистрам в процессорное ядро 8051 модифицировано поведение команды MOV.

При обращении командой MOV в диапазоны памяти E8h-EFh и D8h-DFh резидентного ОЗУ выполняется остановка большей части блоков процессора и задействуется блок управления внешними регистрами. Этот блок выполняет запись/чтение внешнего регистра, разворачивая диаграмму записи/чтения на линиях ввода-вывода. Блок управления внешними регистрами управляет линиями ввода-вывода напрямую, минуя системную шину, что позволяет прочитать или записать данные в регистр за минимальное количество тактов.

Во время остановки процессора продолжает работать таймер и блок обращения к памяти программ, осуществляющий предварительную вычитку данных в FIFO.

Обращение к внешним регистрам

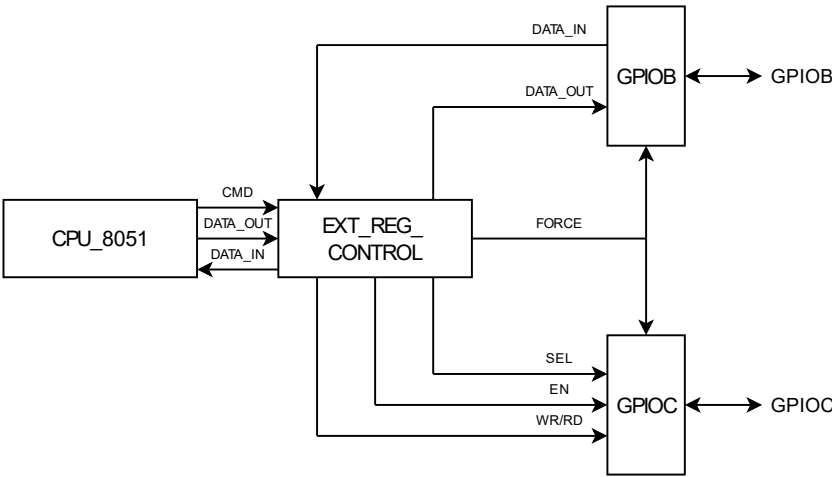


Рисунок 6. Принцип работы блока управления внешними регистрами

Когда ядро 8051 декодирует одну из команд обращения к внешним регистрам, оно передает эту команду (и, в случае записи, данные для записи) в блок управления внешними регистрами (EXT_REG_CONTROL). Блок, в свою очередь, выдает сигнал FORCE для GPIOB и GPIOC. Поэтому сигналу на выходы GPIOC начинают выдаваться сигналы SEL, EN и WR/RD для внешних регистров, а на выходы GPIOB выдаются данные для записи в регистр (при записи), либо же эти выходы принимают данные из внешнего регистра (при чтении). Включение/выключение выходного буфера GPIO происходит автоматически, вне зависимости от текущего значения регистра GPIO_DIR. Вывод GPIO должен быть настроен как вывод общего назначения (соединен с портом P процессора) в регистрах GPIO_ALTF0 и GPIO_ALTF1.

Таблица 10. Описание сигналов внешних регистров

Сигнал	Вывод GPIO	Направление	Описание
DATA<7:0>	GPIOB<7:0>	Выход при записи, вход при чтении	8-битная шина данных
SELL<3:0>	GPIOC<7:6; 3:2>	Выход	Выбор внешнего регистра
EN	GPIOC<1>	Выход	Сигнал разрешения доступа к внешнему регистру
WR/RD	GPIOC<0>	Выход	Сигнал записи/чтения: «1» – запись, «0» – чтение

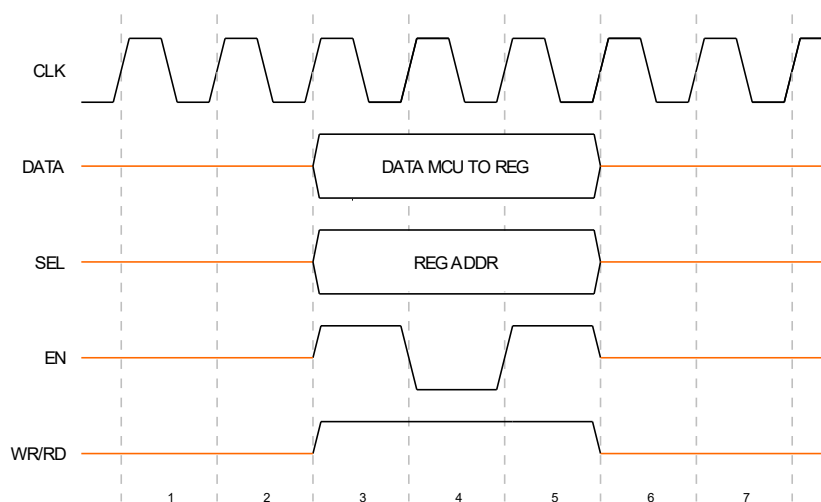


Рисунок 7. Временная диаграмма записи во внешний регистр

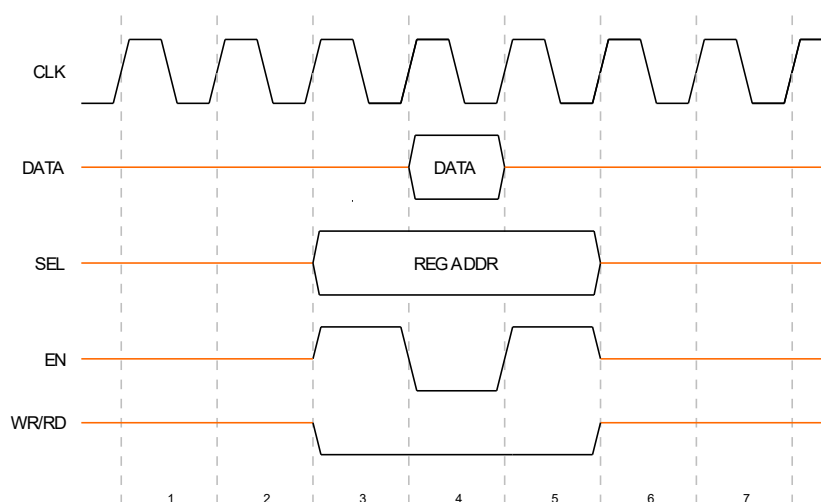


Рисунок 8. Временная диаграмма чтения из внешнего регистра

До начала диаграммы GPIOB и GPIOC были настроены на вход, поэтому все сигналы (DATA, SEL, EN, WR/RD) находятся в состоянии высокого импеданса.

Сигнал CLK, показанный на диаграммах – это тактовый сигнал процессора 8051. Это внутренний сигнал системы, он не выдается на внешние регистры, однако именно к нему привязаны обе формируемые диаграммы.

Диаграмма начинается в момент, когда процессор начинает выполнение команды MOV. На такте 1 CPU декодирует полученную из памяти программ команду. На такте 2 он передает команду в блок управления регистрами. В этот момент блок управления внешними регистрами захватывает контроль над GPIOB и GPIOC. Следующие 3 такта (такты 3, 4, 5) блок выдает диаграмму записи или чтения. После этого все линии GPIOB и GPIOC возвращаются в состояние высокого импеданса.

На тактах 6 и 7 несмотря на то, что чтение или запись закончены, CPU удерживается в состоянии остановки. Это сделано для того, чтобы если следующей командой CPU считал значение со своих портов P, он не считал неверные данные. Данные от GPIO к портам P попадают через синхронизатор, который задерживает данные на 2 такта. Соответственно, эти 2 такта, CPU должен удерживаться, чтобы данные успели пройти через синхронизатор. Таким образом, после окончания команды MOV CPU увидит на своих портах точно те же данные, что и до выполнения команды MOV.

На 8 такте CPU начинает выполнение следующей команды.

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

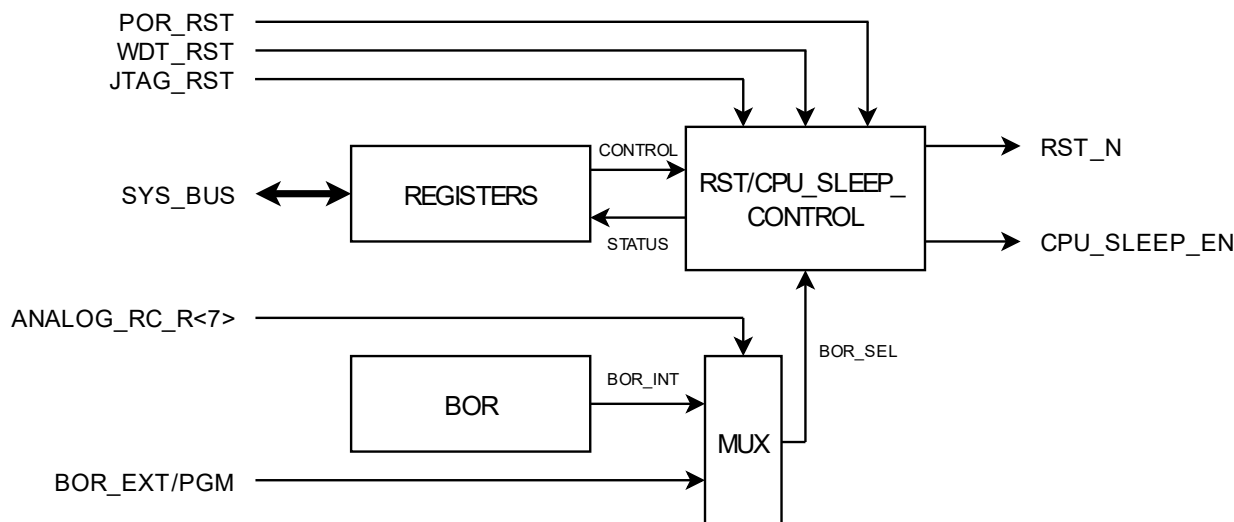


Рисунок 9. Структурная схема системы управления сбросом и питанием

Система управления сбросом и питанием состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- BOR – супервизор питания, формирует сигнал BOR при падении напряжения питания ниже $3,8 \pm 0,2$ В;
- MUX – мультиплексор, позволяет выбрать источник BOR битом ANALOG_RC_R<7> конфигурационной памяти;
- 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_RST и BOR_EXT/PGM имеют активный уровень «0». Вывод BOR_EXT/PGM работает, только когда вывод TM находится в «0». После того как на выводе POR_RST и внутреннем сигнале BOR_SEL сформируется неактивный уровень, еще в течение 1000 тактов системной частоты блок RST/CPU_SLEEP_CONTROL подает сигнал сброса системы RST_N. Микроконтроллер может быть сброшен программными методами от сторожевого таймера (сигнал WDT_RST) или через JTAG (сигнал JTAG_RST). Следует иметь в виду, что JTAG сбрасывается только по сигналу POR_RST, а модуль DEBUGGER по POR_RST и BOR_SEL. В модуле CMM регистры, связанные с выбором и настройкой текущей частоты сбрасываются только по POR_RST и BOR_SEL, 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	RC	Регистр статусов

PMM_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							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.

PMM_ST

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв						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 источниками тактовых сигналов системы предназначен для переключения между источниками тактирования, изменения периода синхросигнала и для перехода в режим «SLEEP».

Микроконтроллер в процессе работы может переключаться между двумя источниками тактового сигнала:

- XTAL_CLK;
- RC_CLK.

В свою очередь, источником синхросигнала XTAL_CLK в зависимости от настроек конфигурационной памяти (модуль ANALOG_CFG) может быть как внешний кварцевый генератор, так и встроенный RC-генератор, а источником RC_CLK всегда является встроенный RC-генератор.

Структурная схема

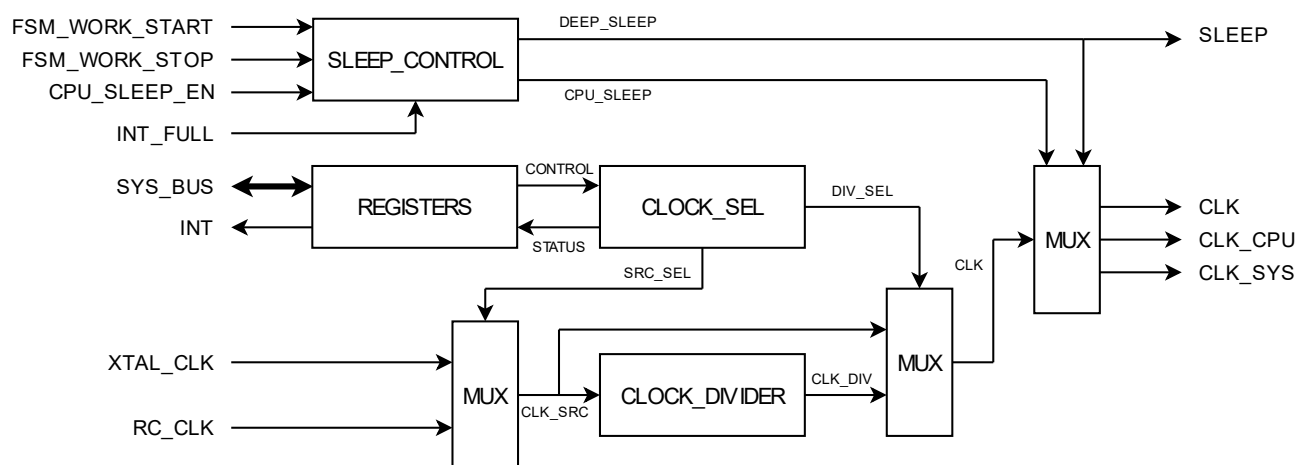


Рисунок 10. Структурная схема модуля управления источниками тактовых сигналов системы

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 от XTAL_CLK. Пользователь может изменить источник тактирования путем записи необходимого значения в бит RC_XTAL регистра CMM_CTRL. 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», а NSWITCH «1». При большой разнице частот, процесс переключения займет порядка 12 тактов частоты, на которую происходит переключение, и 3 тактов текущей частоты.

Пользователь имеет возможность не переключаться между источниками XTAL_CLK и RC_CLK при переходе в режим «Глубокий сон», если источником XTAL_CLK уже является RC-генератор

без ФАПЧ. Переход системы в режим «Глубокий сон» при тактировании от источника XTAL_CLK на какой-либо другой частоте приведет к полной остановке системы без возможности программного восстановления работоспособности.

Деление частоты тактирования

Текущая частота тактирования системы может быть поделена. Для этого необходимо записать коэффициент в регистр CMM_DIV.

Регистры модуля управления источниками тактовых сигналов системы

№	Аббревиатура	Доступ	Описание
2100h	CMM_CTRL	RW	Регистр управления
2101h	CMM_DIV	RW	Регистр 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*					Резерв*		RC_XTAL
Начальное значение	0					1		0

* – Не изменять начальное значение.

RC_XTAL – бит управления источником тактирования системы f_{src} :

1 – тактирование от RC_CLK;

0 – тактирование от XTAL_CLK.

CMM_DIV

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

COEF – коэффициент деления исходной частоты тактирования.

При COEF>0 частота тактирования системы рассчитывается по формуле:

$$f_{clk} = \frac{f_{src}}{2 * COEF},$$

где f_{src} – исходная частота от источника XTAL_CLK или RC_CLK.

При COEF=0:

$$f_{clk} = f_{src}$$

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.

СММ_ST

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

SWITCH – переключение между частотой XTAL_CLK и RC_CLK:

1 – идет переключение на выбранную частоту;

0 – система работает на выбранной частоте.

NSWITCH – процесс переключения частоты тактирования окончен.

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 окончанию 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 регистры GPIO_ALTF0 и GPIO_ALTF1.

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. Порты процессора соединены с блоками GPIO следующим образом:

- Порт P0 соединен с GPIOA;
- Порт P2 соединен с GPIOB;
- Порт P3 соединен с GPIOC.

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».

Структурная схема

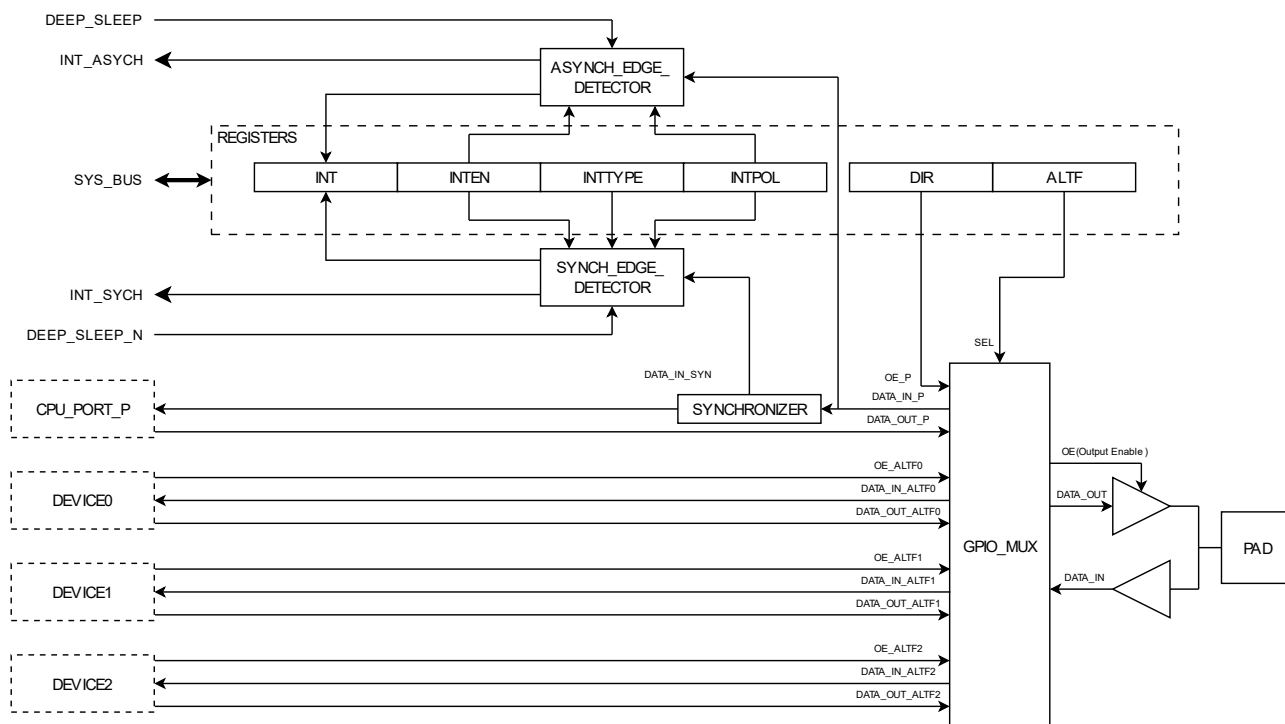


Рисунок 11. Структурная схема GPIO для одного из выводов микроконтроллера

GPIO_MUX соединяет PAD (вывод микроконтроллера) либо с портом P процессора (CPU_PORT_P), либо с одной из альтернативных функций этого вывода. GPIO_MUX управляется регистрами ALTF. Если вывод соединен с портом процессора, то направление (вход/выход) определяется регистром DIR.

Если вывод соединен с альтернативной функцией, то направление (вход/выход) определяется этой альтернативной функцией (периферийным устройством DEVICEx).

В блоке GPIO присутствуют два детектора, способных формировать прерывания - синхронный (SYNCH_EDGE_DETECTOR) и асинхронный (ASYNCH_EDGE_DETECTOR). Работа детекторов управляется регистрами INTEN, INTPOL и INTTYPE. Статус прерываний сохраняется в регистре 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
GPIOA			
2300h	GPIOA_DIR_SET	RW	Установка режима работы выходного буфера
2301h	GPIOA_DIR_CLR	RW	
2304h	GPIOA_ALTF0	RW	Выбор альтернативной функции
2305h	GPIOA_ALTF1	RW	
2306h	GPIOA_INTEN_SET	RW	Разрешение прерываний
2307h	GPIOA_INTEN_CLR	RW	
2308h	GPIOA_INTTYPE_SET	RW	Выбор типа прерывания (фронт/уровень)
2309h	GPIOA_INTTYPE_CLR	RW	
230Ah	GPIOA_INTPOL_SET	RW	Выбор полярности входного сигнала, при которой формируются прерывания
230Bh	GPIOA_INTPOL_CLR	RW	
230Ch	GPIOA_INT	R	Статус прерываний

№	Аббревиатура	Доступ	Описание
GPIOB			
2400h	GPIOB_DIR_SET	RW	Установка режима работы выходного буфера
2401h	GPIOB_DIR_CLR	RW	
2404h	GPIOB_ALTF0	RW	Выбор альтернативной функции
2405h	GPIOB_ALTF1	RW	
2406h	GPIOB_INTEN_SET	RW	Разрешение прерываний
2407h	GPIOB_INTEN_CLR	RW	
2408h	GPIOB_INTTYPE_SET	RW	Выбор типа прерывания (фронт/уровень)
2409h	GPIOB_INTTYPE_CLR	RW	
240Ah	GPIOB_INTPOL_SET	RW	Выбор полярности входного сигнала, при которой формируются прерывания
240Bh	GPIOB_INTPOL_CLR	RW	
240Ch	GPIOB_INT	R	Статус прерываний
GPIOC			
2500h	GPIOC_DIR_SET	RW	Установка режима работы выходного буфера
2501h	GPIOC_DIR_CLR	RW	
2504h	GPIOC_ALTF0	RW	Выбор альтернативной функции
2505h	GPIOC_ALTF1	RW	
2506h	GPIOC_INTEN_SET	RW	Разрешение прерываний
2507h	GPIOC_INTEN_CLR	RW	
2508h	GPIOC_INTTYPE_SET	RW	Выбор типа прерывания (фронт/уровень)
2509h	GPIOC_INTTYPE_CLR	RW	
250Ah	GPIOC_INTPOL_SET	RW	Выбор полярности входного сигнала, при которой формируются прерывания
250Bh	GPIOC_INTPOL_CLR	RW	
250Ch	GPIOC_INT	R	Статус прерываний

GPIOx_DIR_SET/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.

GPIOx_ALTF0

Бит	7	6	5	4	3	2	1	0
Назначение	IO3_ALTF		IO2_ALTF		IO1_ALTF		IO0_ALTF	
Начальное значение	0							

IOx_ALTF – альтернативная функция:

11b – включена альтернативная функция ALTF2;

10b – включена альтернативная функция ALTF1;

01b – включена альтернативная функция ALTF0;

00b – альтернативные функции выключены, выходным буфером управляет GPIO.

GPIOx_ALTF1

Бит	7	6	5	4	3	2	1	0
Назначение	IO7_ALTF		IO6_ALTF		IO5_ALTF		IO4_ALTF	
Начальное значение	0							

IOx_ALTF – альтернативная функция:

11b – включена альтернативная функция ALTF2;

10b – включена альтернативная функция ALTF1;

01b – включена альтернативная функция ALTF0;

00b – альтернативные функции выключены, выходным буфером управляет GPIO.

GPIOx_INTEN_SET/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.

GPIOx_INTTYPE_SET/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_INTTYPE** регистра GPIO_INTTYPE_CLR:

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

Чтение **IOx_INTTYPE** регистров GPIO_INTTYPE_SET/GPIO_INTTYPE_CLR:

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

GPIOx_INTPOL_SET/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.

GPIOx_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/CLR и GPIO_INTTYPE_SET/CLR условиям) на данном выводе;
- 0 – фронт или уровень не был зафиксирован.

SPI

Общая информация

SPI – последовательный синхронный стандарт передачи данных в режиме дуплекса, предназначенный для обеспечения простого высокоскоростного сопряжения микроконтроллера и периферии.

Основные характеристики:

- работа в режиме «ведомого» и «ведущего»;
- работа в дуплексном и симплексном режимах;
- формат кадра 8 или 16 бит;
- максимальная скорость передачи соответствует системной частоте;
- отдельные буферы на прием и передачу глубиной 8 слов;
- фильтр по линии синхросигнала.

Структурная схема

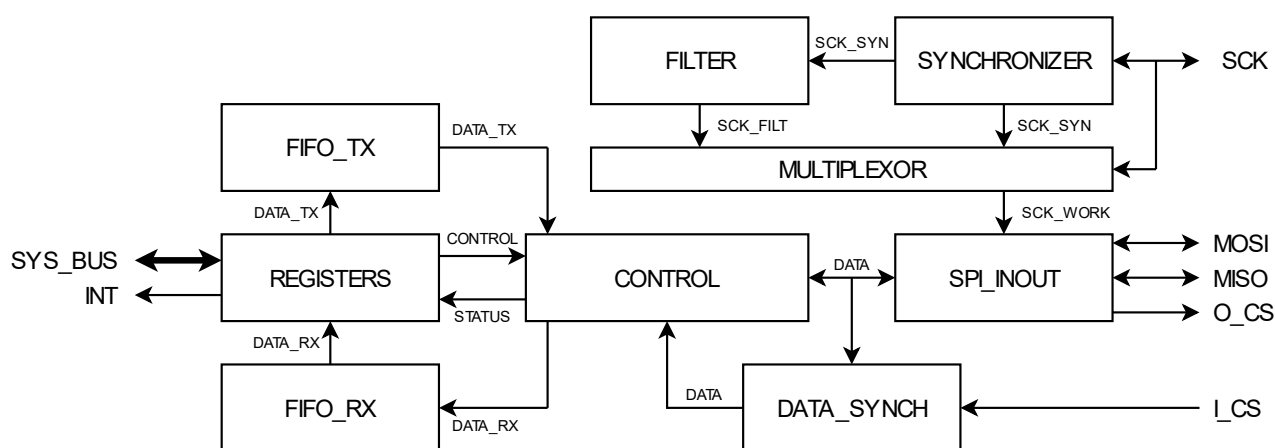


Рисунок 12. Структурная схема SPI интерфейса

SPI состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO_TX – буфер передатчика;
- FIFO_RX – буфер приемника;
- CONTROL – управляющий автомат модуля SPI;
- SPI_INOUT – блок управления линиями интерфейса;
- DATA_SYNCH – синхронизатор данных, производящий синхронизацию данных с системной частоты на частоту SCK и в обратном направлении;
- SYNCHRONIZER – синхронизатор линии SCK, работающий по требованию пользователя;
- FILTER – фильтр линии SCK, избавляющий от помех;
- MULTIPLEXOR – мультиплексор, выбирающий частоту тактирования блока SPI_INOUT в зависимости от входных настроек.

Алгоритмы работы

В режиме «ведущий» модуль автоматически генерирует синхросигнал для ведомого устройства по линии SCK.

Процедура работы с ведущим:

- задействовать один из свободных выводов GPIO как CS, настроить его на выход и перевести в «1»;
- в регистре SPI_MSK разрешить необходимые прерывания;
- в регистре SPI_CFG2 установить биты BR для определения скорости обмена;
- в регистре SPI_CFG0:
 - определить формат данных с помощью бита DWW;
 - настроить формат кадра при помощи бита BO;
 - установить биты POL и PHA для определения соответствия между данными и синхросигналом (настройка должна быть одинакова для ведущего и для ведомого);
 - установить бит MOD для перехода в режим «ведущий».
- для режимов, связанных с передачей данных необходимо записать отправляемую информацию в регистры SPI_TX1 и SPI_TX0 (необходимо записать столько слов данных, сколько необходимо отправить за данную передачу; имеется возможность отправить более 8 слов в режимах с PHA в «1», дописывая слова по опустошению буфера передатчика);
- перевести CS в «0», используя ранее настроенный GPIO, таким образом разрешив работу ведомому устройству;
- в регистре SPI_CTRL разрешить работу модуля битом EN;
- на основании данных регистра SPI_ST установить момент завершения передачи, перевести CS в «1» и выключить модуль битом EN в регистре SPI_CTRL;
- принимаемые данные можно вычитать из регистров SPI_RX1 и SPI_RX0;
- при необходимости продолжить работу с процедуры записи отправляемой информации или следующей за ней процедуры.

В данном формате MOSI работает как выход данных, а MISO как вход данных. Так как CS задается через вывод GPIO, пользователю необходимо самостоятельно выдерживать задержки между фронтами SCK и CS в начале и конце передачи данных. Минимальная задержка перед первым фронтом SCK определяется временем записи бита EN в регистр SPI_CTRL.

В режиме «ведомый» синхросигнал поступает от ведущего устройства. Значение бит BR не влияет на работу модуля. Рекомендуется включить модуль до того, как начнет передаваться синхросигнал, иначе будут приняты сдвинутые данные. Буфер передатчика должен содержать данные **до первого изменения синхросигнала**, иначе отправляемые данные могут быть искажены при перезаписи в сдвиговый регистр.

Процедура работы с ведомым:

- в регистре SPI_MSK разрешить необходимые прерывания;
- в регистре SPI_CFG0:
 - определить формат данных с помощью бита DWW;
 - настроить формат кадра при помощи бита BO;
 - установить биты POL и PHA для определения соответствия между данными и синхросигналом (настройка должна быть одинакова для ведущего и для ведомого);
 - установить бит MOD для перехода в режим «ведомый».
- в регистре SPI_CTRL разрешить работу модуля битом EN;
- для режимов, связанных с передачей данных необходимо записать отправляемую информацию в регистры SPI_TX1 и SPI_TX0;
- на основании данных регистра SPI_ST установить момент завершения передачи;
- принимаемые данные можно вычитать из регистров SPI_RX1 и SPI_RX0;
- при необходимости продолжить передачу данных с процедуры записи отправляемой информации или следующей за ней процедуры.

В данном формате MOSI работает как вход данных, MISO как выход данных.

Дуплексный и симплексный режимы

SPI может работать в двух конфигурациях:

- две однонаправленные линии данных;
- одна двунаправленная линия данных (режим «только прием» или «только передача»).

Две однонаправленные линии данных (стандартный режим). Активируется, когда бит TR_MOD регистра SPI_CFG0 равен «0». Бит RXO регистра SPI_CFG1 задает режим передачи данных. Если RXO неактивен, то модуль работает как на прием, так и на передачу. Если RXO активен, то модуль работает только на прием. Ведущий постоянно генерирует синхросигнал для ведомого. Линия передачи остается незадействованной. У ведущего это линия MOSI, у ведомого – MISO.

Одна двунаправленная линия данных. Режим активируется, когда бит TR_MOD регистра SPI_CFG0 равен «1». В данном режиме модуль SPI работает только на прием, либо только на передачу, что определяется битом EN_TX. У ведущего для приема и передачи используется линия MOSI, у ведомого MISO. MISO у ведущего или MOSI у ведомого не используется. В режиме «только прием» ведущий постоянно генерирует синхросигнал для ведомого. В режиме «только передача» необходимо игнорировать состояние принимающего буфера.

Передача данных

В соответствии с требованиями протокола SPI, в зависимости от настройки модуля, данные передаются на линию либо по фронту, либо по спаду синхросигнала. Передача начинается при наличии данных в буфере передатчика. Данные поступают в сдвиговый регистр и далее биты слова передаются последовательно старшим или младшим вперед в зависимости от настройки BO. Установка бита TXB_E, сообщает о том, что буфер передатчика опустел, но данные еще могут находиться в сдвиговом регистре, следовательно, будет продолжаться их передача. Об окончании передачи последнего слова данных сообщает бит NULL_BUF. С этого момента генерация синхросигнала будет остановлена, до новой записи в буфер передатчика.

Буфер рассчитан на 8 слов. При переполнении буфера передатчика в регистре SPI_ST устанавливается бит TXB_OV. Записываемое слово будет утеряно. Пользователь имеет возможность очистить буфер передатчика путем записи в бит TXBS_RST единицы.

По битам TXB_E, NULL_BUF, TXB_OV может быть настроено прерывание в регистре SPI_MSK.

Прием данных

В соответствии с требованиями протокола SPI, в зависимости от настройки модуля, данные фиксируются по фронту, либо по спаду синхросигнала. Биты последовательно заполняют сдвиговый регистр. Затем из сдвигового регистра принятые данные поступают в буфер приемника. Когда в буфер передано хотя бы одно слово данных, в регистре SPI_ST выставляется бит RX_NE.

Буфер рассчитан на 8 слов данных. Пользователь может последовательно вычитать принятые данные или очистить буфер путем записи бита RXBS_RST в регистр SPI_CTRL во избежание переполнения. В случае, когда буфер заполнен, приход нового слова данных вызовет установку бита RXB_OV регистра SPI_ST. Само слово записано не будет.

В режиме «только передача» данные в буфере приемника не фиксируются.

По битам RX_NE, RXB_OV может быть настроено прерывание в регистре SPI_MSK.

Работа с входом выбора микросхемы

Модуль SPI имеет в своем составе вход выбора микросхемы – I_CS.

В режиме «ведомый» необходимо держать низкий логический уровень на входе I_CS на протяжении всей передачи. При появлении высокого логического уровня на данном входе, регистр приемника будет очищен, а работа модуля остановлена до появления требуемого значения на I_CS.

Пользователь может перейти в программный режим управлением входа I_CS с помощью бита SS_CTRL регистра SPI_CFG0. Далее пользователю достаточно имитировать необходимый логический уровень с помощью бита SSS регистра SPI_CTRL. В остальном принцип работы модуля полностью соответствует работе в аппаратном режиме.

Особенности работы в режиме «ведомый». Синхронизация и фильтрация

Модуль SPI может вести прием и передачу данных на системной частоте и близкой к ней. Однако это накладывает ограничение на работу с буфером передатчика в режиме «ведомый». Данные в буфер должны быть гарантированно записаны до выдачи последнего бита данных из сдвигового регистра передатчика в режимах с PHA «0» или до первого изменения SCK перед началом передачи нового слова в режимах с PHA «1», либо же после полного окончания передачи (актуально если отправляется несколько слов в одной посылке, часть которых дописывается в процессе отправки данных). В противном случае, слово, которое было записано в нарушении этих требований, может быть искажено. Данное ограничение можно избежать, если разрешить работу синхронизатора. В таком случае в сдвиговый регистр будут записаны новые данные или нули.

Модуль SPI оснащен синхронизатором синхросигнала. Для включения синхронизатора необходимо записать «1» в бит SYN регистра SPI_CFG1. Синхронизатор вносит задержку в два такта системной частоты. Значит для корректного приема и передачи данных синхрочастота должна быть уменьшена и соотносится с системной как $f_{clk}/7$.

Модуль SPI оснащен фильтром синхрочастоты. Для включения фильтра необходимо записать «1» в бит FIL регистра SPI_CFG1. Не рекомендуется включать фильтр независимо от синхронизатора. Фильтр может сгладить один сбой на промежутке времени, соответствующем трем тактам системной частоты. При этом частота синхросигнала также должна быть понижена.

С учетом включенного синхронизатора и фильтра частота синхросигнала должна быть меньше системной частоты минимум в 13 раз.

Также при работе на системной частоте или близкой к ней необходимо учитывать, что синхронизация CS занимает 1-2 такта частоты микроконтроллера. Поэтому после перехода CS в «0» необходимо сделать паузу 3-4 такта частоты микроконтроллера перед началом подачи SCK. В случае PHA равной «0» первый бит данных поступит на MISO по синхронизированному CS с буфера передатчика. Захват данных с буфера в сдвиговый регистр происходит на первом такте SCK по фронту или срезу в зависимости от POL (аналогично в режимах с PHA равной «1»). После приема последнего бита сообщения данные поступят в буфер приемника через 2-3 такта частоты микроконтроллера. Останется ли CS в «0» или перейдет в «1» не имеет значения. MISO перейдет в Z-состояния через 1-2 такта после перехода CS в «0». Ниже на рисунке представлен лучший вариант передачи в режиме «00», так как не потребовалось дополнительных тактов на синхронизацию.

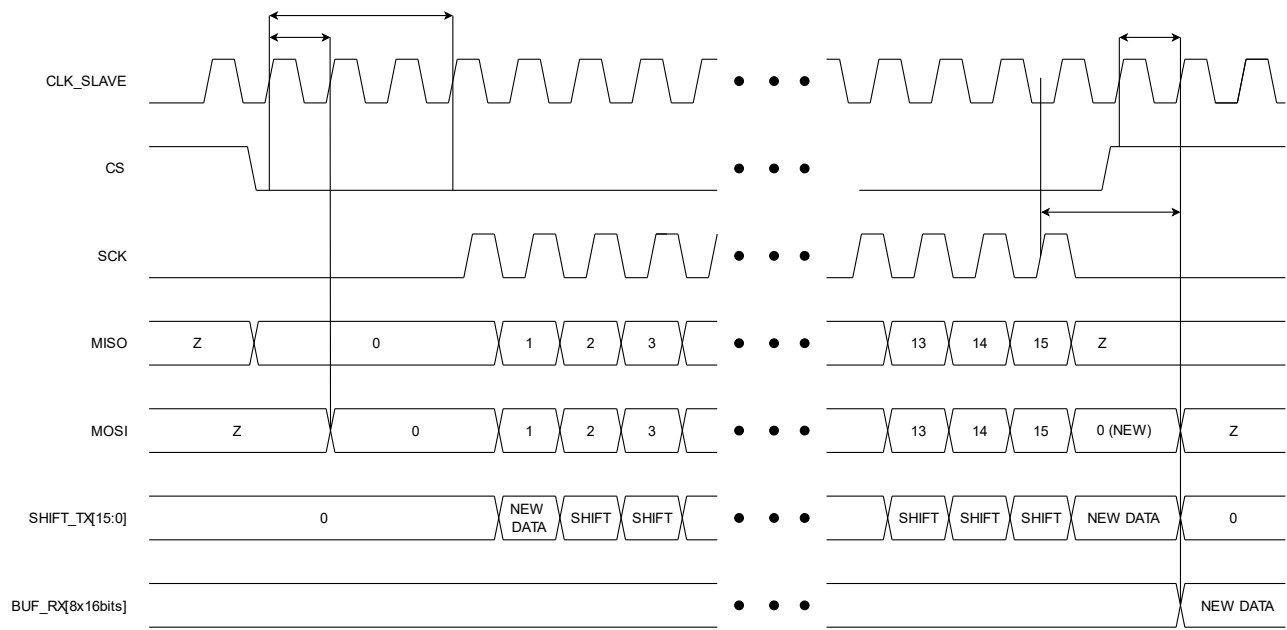


Рисунок 13. Передача данных в режиме «ведомый» на частоте микроконтроллера

Как видно на рисунке новые данные поступают в сдвиговый регистр по последнему срезу (или фронту в зависимости от POL) SCK в режимах с PHA «0». Если буфер пуст, то это будут нули. В режимах с PHA «1» новые данные поступают в сдвиговый регистр по первому фронту (или срезу) первого бита нового слова. Из этого следует, что в режимах с PHA «1» пользователь может делать паузы между передачами слов данных в рамках одной передачи SPI, во время которых ведомый микроконтроллер может обработать только что полученные данные и записать результат в буфер передатчика.

Таким образом, данные в буфере передатчика не должны меняться рядом с моментом захвата в сдвиговый регистр передатчика или в сдвиговый регистр приемника ведущего.

Регистры SPI

№	Аббревиатура	Доступ	Описание
SPI0			
2600h	SPI0_CTRL	RW	Регистр управления
2604h	SPI0_CFG0	RW	Регистр конфигурации 0
2605h	SPI0_CFG1	RW	Регистр конфигурации 1
2606h	SPI0_CFG2	RW	Регистр конфигурации 2
2607h	SPI0_CFG3	RW	Регистр конфигурации 3
2608h	SPI0_MSK	RW	Регистр маски
260Ch	SPI0_ST	R	Регистр статуса
2610h	SPI0_TX0	W	Регистр для записи данных в буфер передатчика 0
2611h	SPI0_TX1	W	Регистр для записи данных в буфер передатчика 1
2610h	SPI0_RX0	R	Регистр для чтения принятых данных 0
2611h	SPI0_RX1	R	Регистр для чтения принятых данных 1
SPI1			
2700h	SPI1_CTRL	RW	Регистр управления
2704h	SPI1_CFG0	RW	Регистр конфигурации 0
2705h	SPI1_CFG1	RW	Регистр конфигурации 1
2706h	SPI1_CFG2	RW	Регистр конфигурации 2

2707h	SPI1_CFG3	RW	Регистр конфигурации 3
2708h	SPI01_MSK	RW	Регистр маски
270Ch	SPI1_ST	R	Регистр статуса
2710h	SPI1_TX0	W	Регистр для записи данных в буфер передатчика 0
2711h	SPI1_TX1	W	Регистр для записи данных в буфер передатчика 1
2710h	SPI1_RX0	R	Регистр для чтения принятых данных 0
2711h	SPI1_RX1	R	Регистр для чтения принятых данных 1

SPIx_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв				SSS	TXBS_RST	RXBS_RST	EN
Начальное значение	0							

SSS (Slave Select Signal) – состояние этого бита воздействует на устройство только при установленном бите SS_CTRL (при программном управлении I_CS):

1 – имитировать высокий логический уровень;

0 – имитировать низкий логический уровень.

Значение этого бита принудительно заменяет состояние входа I_CS, которое при этом игнорируется.

TXBS_RST (Transmitter Buffer Set Reset) – сброс состояния буфера передатчика:

1 – сброс состояния буфера;

0 – сброс неактивен.

Сброс записанной «1» происходит аппаратно через один такт.

RXBS_RST (Receiver Buffer Set Reset) – сброс состояния буфера приемника:

1 – сброс состояния буфера;

0 – сброс неактивен.

Сброс записанной «1» происходит аппаратно через один такт.

EN – включение модуля SPI:

1 – модуль включен;

0 – модуль выключен.

SPIx_CFG0

Бит	7	6	5	4	3	2	1	0
Назначение	EN_TX	TR_MOD	SS_CTRL	DWW	BO	POL	PHA	MOD
Начальное значение	0							

EN_TX– в режиме двунаправленного обмена данными по одной линии, разрешает или запрещает передачу данных:

1 – выход активен (только передача данных);

0 – выход неактивен (только прием данных).

TR_MOD– разрешает или запрещает использование двунаправленного режима обмена данными по одной линии:

1 – режим 1-ой двунаправленной линии данных;

0 – режим 2-х однонаправленных линий данных.

SS_CTRL (Slave Signal Control) – программное управление выбором устройства:

1 – программный контроль включен;

0 – программный контроль выключен.

Когда этот бит установлен, вместо уровня на входе I_CS учитывается состояние бита SSS.

DWW (Data Word Width) – формат данных:

1 – 16 бит;

0 – 8 бит.

BO (Bit Order) – порядок передачи данных:

1 – первым передается младший значащий разряд – LSB;

0 – первым передается старший значащий разряд – MSB.

POL – полярность тактового сигнала:

1 – высокий уровень в режиме ожидания на выводе SCK;

0 – низкий уровень в режиме ожидания на выводе SCK.

PHA – фаза тактового сигнала:

1 – строб данных происходит по второму перепаду тактового сигнала;

0 – строб данных происходит по первому перепаду тактового сигнала.

MOD – выбор режима:

1 – режим «ведущий»;

0 – режим «ведомый».

SPIx_CFG1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв					FIL	SYN	R XO
Начальное значение	0							

FIL – разрешение работы фильтра I_SCK:

1 – I_SCK передается через фильтр;

0 – I_SCK передается без фильтрации.

Фильтр необходимо включать вместе с синхронизатором. Скорость сигнала I_SCK при включенном синхронизаторе и фильтре должна быть не более $f_{clk}/13$.

SYN – разрешение работы синхронизаторов I_SCK:

1 – I_SCK передается через два триггера;

0 – I_SCK передается напрямую.

Скорость сигнала I_SCK при включенном синхронизаторе должна быть не более $f_{clk}/7$:

R XO (Receive Only) – только прием:

1 – выход отключен (только прием данных);

0 – полнодуплексный режим (передача и прием данных).

В комбинации с битом TR_MOD задает направление передачи данных.

SPIx_CFG2

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

BR (Bit Rate) – выбор скорости обмена:

255 – $f_{clk}/510$;

...

3 – $f_{clk}/6$;

2 – $f_{clk}/4$;

1 – $f_{clk}/2$;

0 – f_{clk} .

SPIx_CFG3

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

SOCS (Set Output Chip Select) – выбор ведомого устройства:

1 – ведомое устройство выбрано (сигнал O_CS переходит в «0»);

0 – ведомое устройство не выбрано (сигнал O_CS в «1»).

SPIx_MSK

Прерывания формируется, только если модуль включен.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв			NULL_BUF	RX_NE	MOD_FAIL	TXB_E	RXB_OV
Начальное значение	0							

Для каждого из битов справедливо:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

SPIx_ST

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв	NULL_BUF	BUSY	TXB_OV	RX_NE	MOD_FAIL	TXB_E	RXB_OV
Тип статуса		EVENT	FLAG	EVENT	FLAG	EVENT	FLAG	EVENT
Начальное значение	0						1	0

NULL_BUF – зафиксирован пустой буфер передатчика на момент окончания выдачи слова.

BUSY – флаг занятости. В режиме «ведомый» данный статус выставляется по первому принятому биту и сбрасывается только при выключении модуля. В режиме «ведущий» статус переходит в активный уровень только при наличии данных в сдвиговом регистре передатчика или при обмене данными. Прерывание по данному биту отсутствует.

TXB_OV (Transmitter Buffer Overwrite) – переполнение буфера передатчика. Сообщает о том, что пользователь пытается записать слово данных в уже заполненный буфер передатчика. Записываемое слово будет утеряно. Прерывание по данному биту отсутствует.

RX_NE (Receiver Not Empty) – в буфере приемника содержится хотя бы одно слово данных. Если пользователь сбросит буфер или вычитает все слова, то бит перейдет в неактивное состояние.

MOD_FAIL – вход I_CS был переведен в низкий логический уровень в режиме «ведущий». Данная ситуация возможна при включенном модуле только, если выбрана соответствующая альтернативная функция GPIO или выбран программный контроль линии I_CS битом SS_CTRL регистра SPI_CFG0 и бит SSS регистра SPI_CTRL в «0».

TXB_E (Transmitter Buffer Empty) – буфер передатчика пуст. Статус переходит в неактивный уровень, когда в буфер будет записано хотя бы одно слово данных. Стоит учитывать, что записанное слово переместится в сдвиговый регистр, как только тот опустеет и буфер опять окажется пустым.

RXB_OV (Receiver Buffer Overwrite) – буфер приемника переполнен, а значит он уже потерял одно из принимаемых слов данных. Для того, чтобы такой ситуации не происходило необходимо вычитывать принимаемые данные из буфера через регистр SPI_RX.

SPIx_TX0

При записи в данный регистр в буфер передатчика записываются старшая и младшая часть данных.

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

DATA_TX – данные на передачу, младшая часть.

SPIx_TX1

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

DATA_TX – данные на передачу, старшая часть.

SPIx_RX0

Данный регистр предназначен для чтения принятых данных из буфера приемника. Первое принятое слово данных сразу же выставляется на выходе данного регистра и меняется на следующее при чтении из данного регистра. При чтении из пустого буфера будут вычитаны нули.

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

DATA_RX – принятые данные, младшая часть.

SPIx_RX1

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

DATA_RX – принятые данные, старшая часть.

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

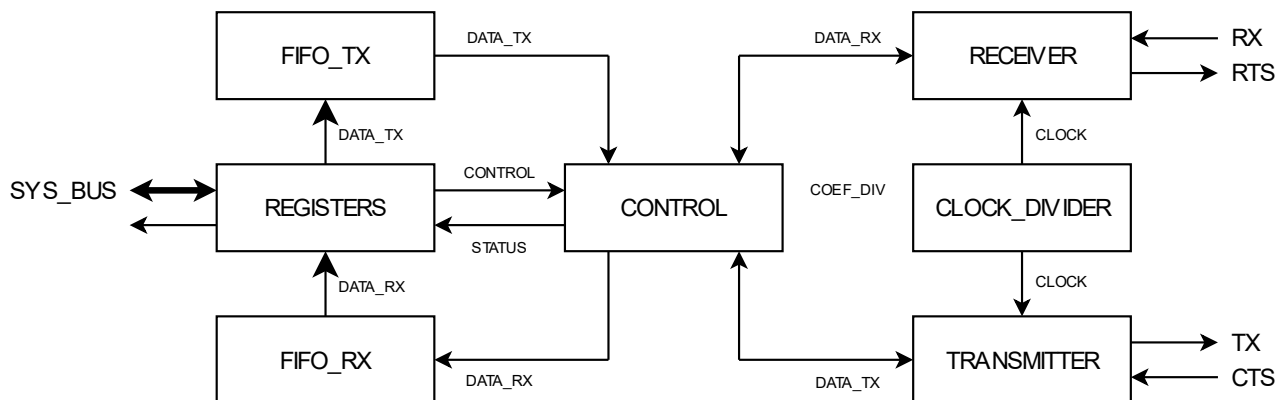


Рисунок 14. Структурная схема UART

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

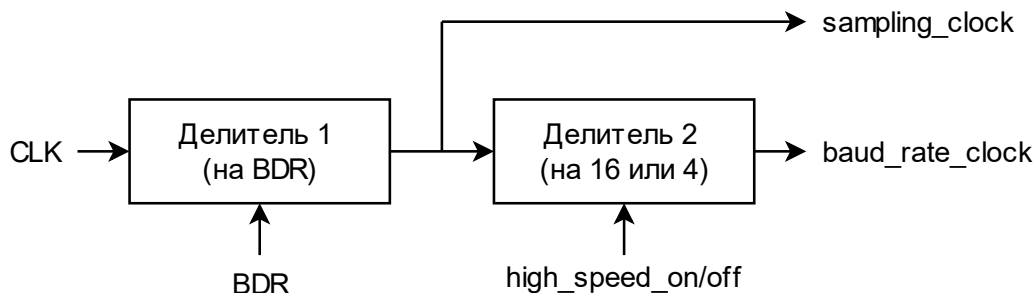


Рисунок 15. Структурная схема делителя частоты

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.

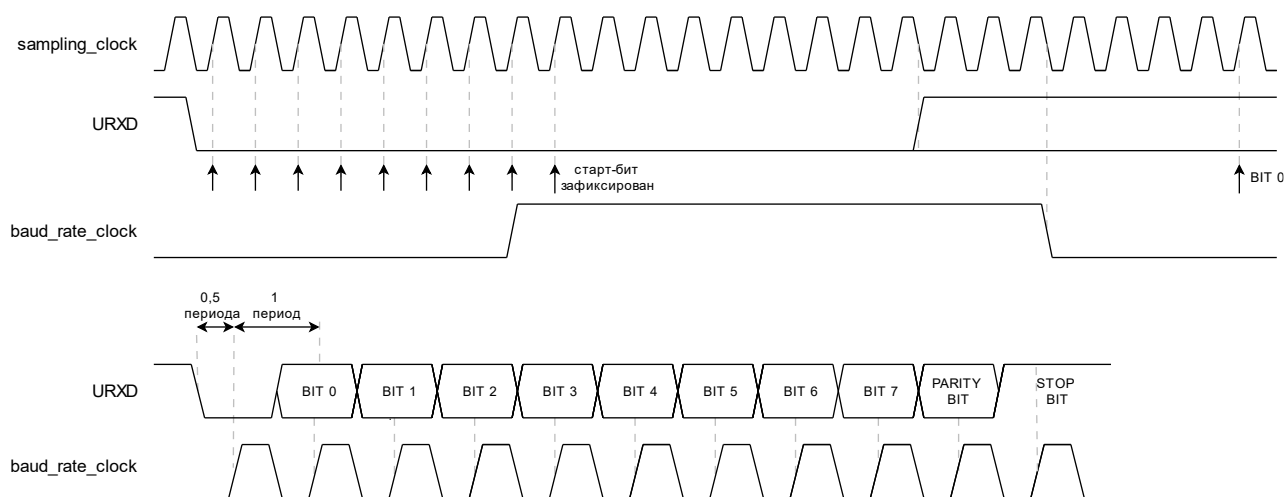


Рисунок 16. Временная диаграмма работы приемника

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
UART0			
2800h	UART0_CFG0	RW*	Регистр конфигурации 0
2801h	UART0_CFG1	RW*	Регистр конфигурации 1
2804h	UART0_BDR0	RW*	Регистр настройки скорости обмена 0
2805h	UART0_BDR1	RW*	Регистр настройки скорости обмена 1
2808h	UART0_TXFIFOLVLO	RW	Регистр контроля уровня заполнения буфера передатчика 0
2809h	UART0_TXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера передатчика 1
280Ch	UART0_RXFIFOLVLO	RW	Регистр контроля уровня заполнения буфера приемника 0
280Dh	UART0_RXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера приемника 1
2810h	UART0_NBMSK	RW	Регистр маски адреса 9-битного режима
2814h	UART0_NBADDR0	RW	Регистр фильтра адреса в 9-битном режиме 0
2815h	UART0_NBADDR1	RW	Регистр фильтра адреса в 9-битном режиме 1
2818h	UART0_MSK0	RW	Регистр маски прерываний 0
2819h	UART0_MSK1	RW	Регистр маски прерываний 1
281Ch	UART0_CTRL	RW	Регистр управления
2820h	UART0_TX	W	Регистр буфера передатчика
2824h	UART0_RX0	R	Регистр буфера приемника 0
2825h	UART0_RX1	R	Регистр буфера приемника 1
2828h	UART0_ST0	R	Регистр статусов 0
2829h	UART0_ST1	R	Регистр статусов 1
UART1			
2900h	UART1_CFG0	RW*	Регистр конфигурации 0
2901h	UART1_CFG1	RW*	Регистр конфигурации 1
2904h	UART1_BDR0	RW*	Регистр настройки скорости обмена 0
2905h	UART1_BDR1	RW*	Регистр настройки скорости обмена 1
2908h	UART1_TXFIFOLVLO	RW	Регистр контроля уровня заполнения буфера передатчика 0
2909h	UART1_TXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера передатчика 1
290Ch	UART1_RXFIFOLVLO	RW	Регистр контроля уровня заполнения буфера приемника 0

№	Аббревиатура	Доступ	Описание
290Dh	UART1_RXFIFOLVL1	RW	Регистр контроля уровня заполнения буфера приемника 1
2910h	UART1_NBMSK	RW	Регистр маски адреса 9-битного режима
2914h	UART1_NBADDR0	RW	Регистр фильтра адреса в 9-битном режиме 0
2915h	UART1_NBADDR1	RW	Регистр фильтра адреса в 9-битном режиме 1
2918h	UART1_MSK0	RW	Регистр маски прерываний 0
2919h	UART1_MSK1	RW	Регистр маски прерываний 1
291Ch	UART1_CTRL	RW	Регистр управления
2920h	UART1_TX	W	Регистр буфера передатчика
2924h	UART1_RX0	R	Регистр буфера приемника 0
2925h	UART1_RX1	R	Регистр буфера приемника 1
2928h	UART1_ST0	R	Регистр статусов 0
2929h	UART1_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.

UARTx_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 послылке:

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 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.

UARTx_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.

UARTx_BDR0

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

BDR – коэффициент деления, младшая часть.

UARTx_BDR1

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

BDR – коэффициент деления, старшая часть.

UARTx_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».

UARTx_TXFIFOLVL1

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

* – Не изменять начальное значение.

UARTx_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».

UARTx_RXFIFOLVL1

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

* – Не изменять начальное значение.

UARTx_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.

UARTx_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.

UARTx_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.

UARTx_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.

UARTx_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.

UARTx_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).

UARTx_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).

UARTx_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.

UARTx_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».

UARTx_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.

UARTx_ST1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв	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 непрочитанное слово данных (и приемник включен).

I2C

Общая информация

Интерфейс I2C применяется для связи между собой однокристальных микроконтроллеров, ЖКИ-индикаторов, портов ввода-вывода, микросхем памяти, аналого-цифровых и цифро-аналоговых преобразователей и т.д.

Основные характеристики:

- только две линии – последовательная линия данных (SDA) и последовательная линия синхронизации (SCL);
- возможностью работы в multi-master среде;
- последовательная передача данных по 8 бит;
- скорости передачи данных: 100 кбит/с, 400 кбит/с;
- фильтрация сигналов на линиях передачи данных (SDA, SCL) от помех;
- глубина буфера приемника и передатчика 8 слов.

Все операции на шине I2C осуществляются при помощи двух проводов SDA и SCL. Как SDA, так и SCL являются двунаправленными линиями, которые необходимо подсоединить к положительному источнику питания через подтягивающий резистор. Когда шина свободна, обе линии за счет подтягивающих резисторов принимают высокий логический уровень. Выходные каскады устройств, подключенных к шине, должны иметь открытый сток или открытый коллектор.

Структурная схема

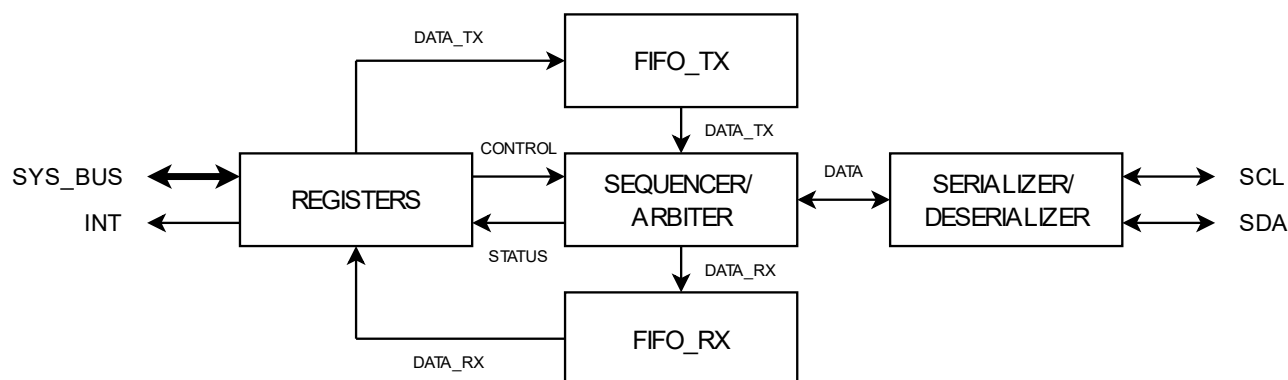


Рисунок 17. Структурная схема I2C

I2C состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- FIFO_TX – буфер передатчика;
- FIFO_RX – буфер приемника;
- SEQUENCER/ARBITER – управляющий автомат интерфейса;
- SERIALIZER/DESERIALIZER – блок предназначен для преобразования параллельного потока данных от управляющего автомата в последовательный, а также для преобразования последовательно потока с внешней шины, в параллельный – для управляющего автомата.

Алгоритмы работы

Процедура передачи данных ведущим:

- в регистрах I2C_MSK0, I2C_MSK1, I2C_MSK2 разрешить необходимые прерывания;
- в регистре I2C_CFG установить биты FILT_DEPTH, определяющие глубину фильтрации;
- в регистре I2C_PRSC0 определить длительность высокого и низкого уровня SCL, установив младшие 8 бит PRSC;
- в регистре I2C_PRSC1:
 - определить режим работы битом F/S, если выбран Fast-mode, то задать коэффициент заполнения битом DUTY;
 - определить длительность высокого и низкого уровня SCL, установив старшие 4 бита PRSC.
- в регистре I2C_PRSC3 установить максимальную длительность петли обратной связи линии SCL битами TRISE;
- записать адрес с «0» в младшем бите и отправляемые данные в регистр I2C_TXFIFO (данный этап можно провести после включения или подачи START последовательности; также данные можно дописывать как во время передачи, контролируя состояние буфера посредством регистра I2C_TXWORDS, статуса BTF регистра I2C_ST0 или статуса FIFO_EMPTY_TX регистра I2C_ST1 и соответствующего прерывания, так и после когда бит статуса TX_END_EMPTY_FIFO регистра I2C_ST2 уже выставлен);
- в регистре I2C_CTRL:
 - определить тип адресации битом ADDR_MOD;
 - установить бит START;
 - включить модуль битом EN.
- по статусному биту FIFO_EMPTY_TX регистра I2C_ST1 или TX_END_EMPTY_FIFO регистра I2C_ST2 установить бит STOP регистра I2C_CTRL (если бит STOP устанавливается по флагу FIFO_EMPTY_TX, т.е. во время передачи последнего слова, то STOP последовательность будет развернута незамедлительно после отправки последнего байта данных и считывания его бита подтверждения, в противном случае между последним $T_{low}/2$ и началом STOP последовательности возникает задержка, связанная с уходом на обработчик прерываний, чтением и записью регистров);
- на основании статусного бита STOP регистра I2C_ST0 установить момент завершения подачи STOP последовательности, выключить модуль или продолжить работу.

При потере арбитража во время отправки данных в случае, если адрес записан в первом бите FIFO_TX, пользователь имеет возможность предпринять вторую попытку после освобождения шины без повторного заполнения буфера, сбросив указатель чтения битом RST_TX_RDP регистра I2C_CTRL.

Процедура приема данных ведущим:

- в регистрах I2C_MSK0, I2C_MSK1, I2C_MSK2 разрешить необходимые прерывания;
- в регистре I2C_CFG:
 - установить биты FILT_DEPTH, определяющие глубину фильтрации;
 - определить режим работы FIFO_RX битом EN_OV.
- в регистре I2C_PRSC0 определить длительность высокого и низкого уровня SCL, установив младшие 8 бит PRSC;
- в регистре I2C_PRSC1:
 - определить режим работы битом F/S, если выбран Fast-mode, то задать коэффициент заполнения битом DUTY;
 - определить длительность высокого и низкого уровня SCL, установив старшие 4 бита PRSC.
- в регистре I2C_PRSC3 установить максимальную длительность петли обратной связи линии SCL битами TRISE;
- записать адрес с «1» в младшем бите в регистр I2C_TXFIFO;

- если необходимо принять более одного байта, то необходимо установить количество слов, при котором формируется соответствующий статус в регистре I2C_RXTHRESHOLD (для завершения приема необходимо отправить NACK во время передачи последнего байта, поэтому установленное значение должно быть как минимум на одно меньше, чем принимаемых байт);
- в регистре I2C_CTRL:
 - определить тип адресации битом ADDR_MOD;
 - если необходимо принять более одного байта установить бит ACK;
 - установить бит START;
 - включить модуль битом EN.
- в регистре I2C_CTRL по статусному биту RX_THRESHOLD_PASS регистра I2C_ST1 или FIFO_RX_NOT_EMPTY, или TX_END_EMPTY_FIFO регистра I2C_ST2 (в зависимости от количества принимаемых байт; если бит STOP устанавливается по FIFO_RX_NOT_EMPTY, то между последним $T_{low}/2$ и началом STOP последовательности возникает задержка, связанная с уходом на обработчик прерываний, чтением и записью регистров):
 - если принимается более одного байта записать «0» в бит ACK;
 - установить бит STOP.
- на основании статусного бита STOP регистра I2C_ST0 установить момент завершения подачи STOP последовательности, считать полученные данные из регистра I2C_RXFIFO, выключить модуль или продолжить работу.

Пользователь имеет возможность считывать буфер принятых данных и изменять регистр I2C_RXTHRESHOLD во время передачи данных ведомым, что позволяет принимать более 8 байт.

Процедура передачи данных ведомым:

- в регистрах I2C_MSK0, I2C_MSK1, I2C_MSK2 разрешить необходимые прерывания;
- в регистре I2C_CFG:
 - установить биты FILT_DEPTH, определяющие глубину фильтрации;
 - определить режим работы FIFO_RX битом EN_OV.
- задать адрес устройства в регистрах I2C_ADDR1 и I2C_ADDR0;
- в регистре I2C_PRSC2 определить длительность удержания уровня SDA после заднего фронта SCL;
- записать отправляемые данные в регистр I2C_TXFIFO;
- в регистре I2C_CTRL:
 - определить тип адресации битом ADDR_MOD;
 - установить бит ACK;
 - включить модуль битом EN.
- дописывать данные в регистр I2C_TXFIFO по мере опустошения выходного буфера, отслеживая его состояние с помощью бит BTF, FIFO_EMPTY_TX или TX_END_EMPTY_FIFO статусных регистров, а также регистра I2C_TXWORDS.

Если на момент чтения нового слова из выходного буфера он пуст, то будут отправлены нули. Мастер должен отправить NACK после приема последнего байта данных, иначе ведомый предпримет попытку продолжить передачу.

Процедура приема данных ведомым:

- в регистрах I2C_MSK0, I2C_MSK1, I2C_MSK2 разрешить необходимые прерывания;
- в регистре I2C_CFG:
 - установить биты FILT_DEPTH, определяющие глубину фильтрации;
 - определить режим работы FIFO_RX битом EN_OV.
- задать адрес устройства в регистрах I2C_ADDR1 и I2C_ADDR0;
- в регистре I2C_PRSC2 определить длительность удержания уровня SDA после заднего фронта SCL;

-
- в регистре I2C_RXTHRESHOLD установить количество слов, при котором формируется соответствующий статус;
- в регистре I2C_CTRL:
 - определить тип адресации битом ADDR_MOD;
 - установить бит ACK;
 - включить модуль битом EN.
- считывать данные и управлять битом ACK регистра I2C_CTRL по мере формирования соответствующих статусов.

Статусы и прерывания

В отличие от остальных модулей в I2C:

- прерывание по биту-флагу не формируется, если бит уже находится в «1», и в этот момент разрешается прерывание;
- прерывание по биту-событию не формируется от последующих событий после перехода бита в «1», пока он не будет сброшен чтением обратно в «0».

Остальные принципы работы, описанные в разделе «Типы статусов и прерывания», остаются неизменными.

Регистры I2C

№	Аббревиатура	Доступ	Описание
2A00h	I2C_CFG	RW	Регистр конфигурации
2A04h	I2C_CTRL	RW	Регистр управления
2A08h	I2C_ST0	R	Регистр статуса 0
2A09h	I2C_ST1	R	Регистр статуса 1
2A0Ah	I2C_ST2	R	Регистр статуса 2
2A0Ch	I2C_ADDR0	RW	Регистр адреса приемника 0
2A0Dh	I2C_ADDR1	RW	Регистр адреса приемника 1
2A10h	I2C_PRSC0	RW	Предделитель 0
2A11h	I2C_PRSC1	RW	Предделитель 1
2A12h	I2C_PRSC2	RW	Предделитель 2
2A13h	I2C_PRSC3	RW	Предделитель 3
2A14h	I2C_MSK0	RW	Регистр маски прерываний 0
2A15h	I2C_MSK1	RW	Регистр маски прерываний 1
2A16h	I2C_MSK2	RW	Регистр маски прерываний 2
2A18h	I2C_TXFIFO	RW	Буфер данных на передачу
2A1Ch	I2C_RXFIFO	R	Буфер принятых данных
2A20h	I2C_TXWORDS	R	Количество непрочитанных слов в буфере передачи
2A24h	I2C_RXTHRESHOLD	RW	Указывает количество непрочитанных слов в буфере приема, при котором формируется соответствующий признак в регистре ST

I2C_CFG

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

FILT_DEPTH – настройка глубины фильтров для входных сигналов SDA, SCL. Выбирается следующим образом:

$$FILT_DEPTH > \frac{T_{noise}}{T_{clk}},$$

где T_{noise} – максимальная длительность помехи, которую необходимо отфильтровать, T_{clk} – период частоты тактирования системы. Соответственно при максимальном значении битов FILT_DEPTH может быть отфильтрована помеха длительностью менее 31 T_{clk} . При значении «0» фильтр выключен.

EN_OV – при переполнении буфера FIFO приемника разрешает или запрещает перезаписывать данные:

- 1 – перезапись данных разрешена;
- 0 – перезапись данных запрещена.

При приеме в режиме «ведущий», когда переполнение разрешено, контроллер может запрашивать от ведомого новые слова и записывать их поверх старых. При возникновении переполнения поднимается признак FIFO_RX_OV. Если переполнение запрещено ведущий остановит прием после появления признака FIFO_RX_FULL.

При приеме в режиме «ведомый», когда переполнение разрешено, можно записывать новые слова поверх старых. При возникновении переполнения установится признак FIFO_RX_OV. Если переполнение запрещено, ведомый не будет записывать новые слова поверх старых, однако при приеме нового слова и наличии признака FIFO_RX_FULL установится признак FIFO_RX_OV.

I2C_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	RST_RX_PNTRS	RST_TX_RDP	RST_TX_PNTRS	ADDR_MOD	ACK	STOP	START	EN
Начальное значение	0							

RST_RX_PNTRS – запись единицы приводит к сбросу указателей FIFO_RX.

RST_TX_RDP (Reset Transmitter Read Pointer) – запись единицы приводит к сбросу указателя чтения FIFO_TX.

RST_TX_PNTRS – запись единицы приводит к сбросу указателей FIFO_TX.

ADDR_MOD – тип адресации:

- 1 – 10-ти битная адресация;
- 0 – 7-ми битная адресация.

ACK – разрешение подтверждение после приема байта (адреса или данных):

- 1 – отправлять подтверждение;
- 0 – не отправлять подтверждение.

При переводе бита в «0» модуль запишет последний байт в буфер приема и подаст на линию NACK.

STOP – сформировать STOP последовательность:

- 1 – в режиме «ведущий» формируется STOP;
- 0 – STOP не формируется.

Бит переводится в «0» устройством после STOP.

START – сформировать START последовательность.:

1 – в режиме «ведущий» формируется START;

0 – START не формируется.

Бит переводится в «0» устройством после START.

EN – разрешение работы модуля:

1 – модуль включен;

0 – модуль выключен.

I2C_ST0

Бит	7	6	5	4	3	2	1	0
Назначение	FIFO_RX_OV	FIFO_RX_FULL	ARB_LOST	BUS_ERROR	ACK_FAILURE	BTF	STOP	START
Тип статуса	FLAG		EVENT	EVENT	EVENT	EVENT	EVENT	EVENT
Начальное значение	0							

FIFO_RX_OV – FIFO_RX переполнен.

FIFO_RX_FULL – FIFO_RX полностью заполнен. При наличии FIFO_RX_OV обнуляется, контролироваться не должен.

ARB_LOST – признак потери арбитража (только для режима «ведущий»). При потере арбитража в процессе передачи слова данных ведущий продолжает формировать импульсы SCL для всех 8 бит данных, затем освобождает линию SCL, и после этого формирует статус ARB_LOST.

BUS_ERROR – признак несвоевременного события START или STOP. Признак появляется, если зафиксирован перепад линии SDA на фоне единичного уровня SCL во время передачи байта данных или бита подтверждения.

ACK_FAILURE – не было подтверждения после передачи байта данных/адреса.

BTF (Byte Transfer Completed) – обмен байтом завершен:

1 – обмен байтом завершился успешно;

0 – обмена байтом не было/завершился не успешно.

Переводится в единицу, если:

- при приеме, когда принят новый байт и отправлен ACK/NACK;
- при передаче, если байт был передан и получен ACK.

Примечание: если при передаче байта принят NACK, BTF не переводится в единицу.

STOP – в режиме «ведущий» сформирован STOP, в режиме «ведомый» обнаружен STOP.

START – в режиме «ведущий» сформирован START, в режиме «ведомый» обнаружен START.

I2C_ST1

Бит	7	6	5	4	3	2	1	0
Назначение	RX_THRESHOLD_PASS	MODE	BUS_CLEAR	SLV_RX	SLV_TX	FIFO_EMPTY_TX	FIFO_FULL_TX	FIFO_RX_NOT_EMPTY
Тип статуса	FLAG							
Начальное значение	1	0	1	0	0	1	0	0

RX_THRESHOLD_PASS – количество непрочитанных слов в буфере приема больше либо равно значению регистра I2C_RX_THRESHOLD.

MODE – режим:

1 – блок находится в режиме «ведущий»;

0 – блок находится в режиме «ведомый».

BUS_CLEAR – признак того, что на линии не идет обмен. Линия будет считаться занятой, если зафиксирован низкий уровень на линиях SDA или SCL, и свободной – если произошло событие STOP.

SLV_RX – признак, что в режиме «ведомый» устройство адресовано на запись.

SLV_TX – признак, что в режиме «ведомый» устройство адресовано на чтение.

FIFO_EMPTY_TX – FIFO_TX пуст. Возникает перед передачей последнего слова данных в момент чтения из буфера (за такт системной частоты до выдачи первого бита на линии SDA в момент времени $T_{low}/2$).

FIFO_FULL_TX – FIFO_TX полностью заполнен.

FIFO_RX_NOT_EMPTY – FIFO_RX не пуст. При наличии FIFO_RX_OV обнуляется, контролироваться не должен.

I2C_ST2

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв							TX_END_EMPTY_FIFO
Тип статуса								EVENT
Начальное значение	0							

TX_END_EMPTY_FIFO – признак того, что в момент завершения передачи слова буфер передачи был пуст. Возникает в момент завершения приема ACK либо NACK после передачи слова (за такт системной частоты до $T_{low}/2$).

I2C_ADDR0

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

ADDR – адрес в режиме «ведомый», младшая часть.

I2C_ADDR1

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

ADDR – адрес в режиме «ведомый», старшая часть.

I2C_PRSC0

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

PRSC – предделитель, младшая часть.

I2C_PRSC1

Бит	7	6	5	4	3	2	1	0
Назначение	F/S	DUTY	Резерв		PRSC			
Начальное значение	0							

F/S – режим:

1 – Fast-mode;

0 – Standard-mode.

DUTY – коэффициент заполнения для Fast-mode:

$$1 - \frac{T_{high}}{T_{low}} = \frac{9}{16};$$

$$0 - \frac{T_{high}}{T_{low}} = \frac{1}{2}.$$

PRSC – предделитель, определяет длительность высокого (T_{high}) и низкого (T_{low}) уровня SCL в режиме «ведущий», старшая часть.

При Fast-mode и DUTY = 1:

$$T_{high} = 9 * PRSC * T_{clk};$$

$$T_{low} = 16 * PRSC * T_{clk}.$$

При Fast-mode и DUTY = 0:

$$T_{high} = PRSC * T_{clk};$$

$$T_{low} = 2 * PRSC * T_{clk}.$$

При Standard-mode:

$$T_{high} = 2 * PRSC * T_{clk};$$

$$T_{low} = 2 * PRSC * T_{clk}.$$

При установке битов PRSC необходимо, чтобы общее время, затраченное на обратную связь при спаде SCL, не превышало $T_{low}/2$. Для этого достаточно выполнения следующего неравенства:

$$\frac{T_{low}}{2 * T_{clk}} > \left\lceil \frac{T_{fall}}{T_{clk}} \right\rceil + FILT_DEPTH + 2,$$

где T_{fall} – максимальная длительность заднего фронта SCL на входе устройства. Из этого следует, что при Standard-mode или Fast-mode с DUTY в «0» и $T_{fall} \ll T_{clk}$ минимальное значение битов PRSC равно 3 при выключенном фильтре (биты FILT_DEPTH в «0»), однако рекомендуется округлять отношение длительности заднего фронта и периода синхросигнала в большую сторону.

Смена значения на линии SDA в режиме «ведущий», если он передает, происходит в момент времени $T_{low}/2$.

I2C_PRSC2

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

SLV_HLD – длительность удержания уровня SDA, младшая часть.

I2C_PRSC3

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

TRISE – максимальная длительность петли обратной связи линии SCL в режиме «ведущий». Используется с целью установить корректную скважность на линии SCL, которая не будет зависеть от длительности переднего фронта SCL на входе устройства и глубины фильтрации. Значение должно быть установлено исходя из максимального времени нарастания переднего фронта SCL, выраженного в периодах частоты тактирования системы:

$$TRISE = \left\lceil \frac{T_{rise}}{T_{clk}} \right\rceil + FILT_DEPTH + 3,$$

где T_{rise} – максимальная длительность переднего фронта SCL на входе устройства. Также следует учитывать:

$$TRISE \leq \frac{T_{high}}{T_{clk}}.$$

SLV_HLD – длительность удержания уровня SDA, старшая часть. В режиме «ведомый» через поле SLV_HLD определяется длительность удержания уровня SDA от заднего фронта SCL до выдачи бита ведомым I2C:

$$T_{hold} = T_{fall} + (FILT_DEPTH + SLV_HLD + 6) * T_{clk}.$$

I2C_MSK0

Возможно формирование прерывания по любому биту статусного регистра I2C_ST0. Расположение битов в I2C_ST0 и I2C_MSK0 аналогично.

Бит	7	6	5	4	3	2	1	0
Назначение	FIFO_RX_OV	FIFO_RX_FULL	ARB_LOST	BUS_ERROR	ACK_FAILURE	BTF	STOP	START
Начальное значение	0							

Для каждого из битов справедливо:

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

I2C_MSK1

Возможно формирование прерывания по любому биту статусного регистра I2C_ST1. Расположение битов в I2C_ST1 и I2C_MSK1 аналогично.

Бит	7	6	5	4	3	2	1	0
Назначение	RX_THRESHOLD_PASS	MODE	BUS_CLEAR	SLV_RX	SLV_TX	FIFO_EMPTY_TX	FIFO_FULL_TX	FIFO_RX_NOT_EMPTY
Начальное значение	0							

Для каждого из битов справедливо:

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

I2C_MSK2

Расположение статусного бита аналогично I2C_ST2.

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

TX_END_EMPTY_FIFO - разрешить прерывание по статусу TX_END_EMPTY_FIFO:

1 – данное прерывание формируется;

0 – данное прерывание не формируется.

I2C_TXFIFO

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

TX_FIFO – буфер передаваемых данных. При передаче адреса в режиме «ведущий» пользователь должен самостоятельно задать R/W битом 0.

I2C_RXFIFO

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

RX_FIFO – буфер принятых данных.

I2C_TXWORDS

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

TX_WORDS – количество непрочитанных слов в буфере передачи (от 0 до 8).

I2C_RXTHRESHOLD

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

RX_THRES – количество непрочитанных слов в буфере приема (от 0 до 8), при котором формируется соответствующий признак в регистре статуса.

1-Wire

Общая информация

1-Wire или OWI – это интерфейс, который представляет собой двунаправленную шину связи для устройств с низкоскоростной передачей данных, в которой данные передаются по цепи питания. Используется всего два провода, один общий, а второй для питания и данных.

Основные характеристики модуля:

- работа в режиме «ведущего»;
- минимальная частота тактирования 1 МГц;
- поддержка стандартной и повышенной скоростей обмена;
- поддержка набора стандартных команд;
- расширенный набор статусов и прерываний.

Структурная схема

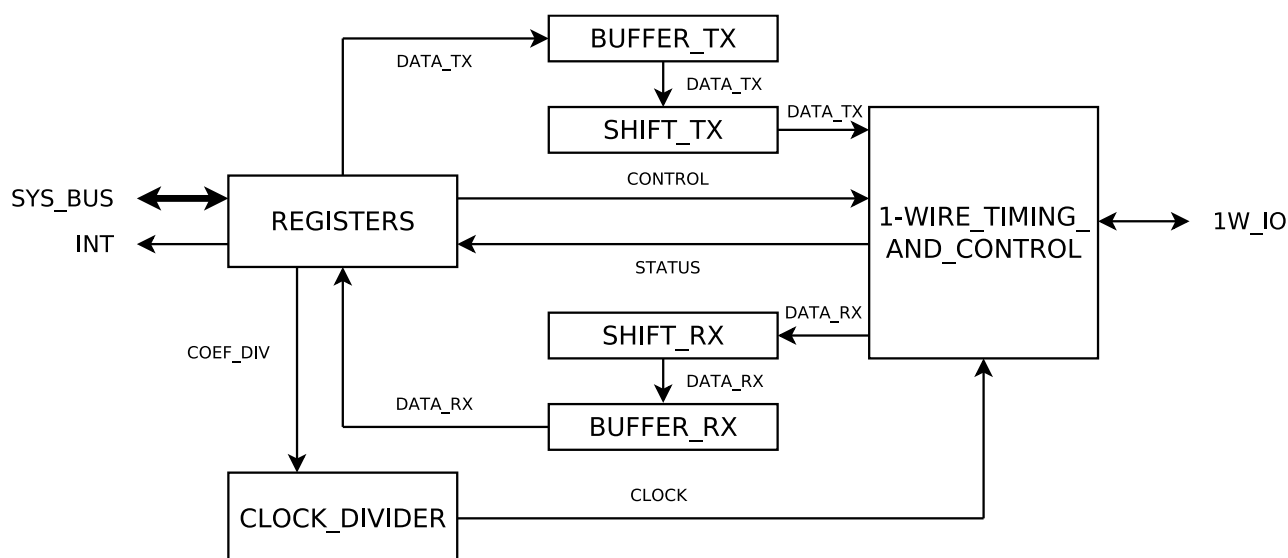


Рисунок 18. Структурная схема 1-Wire

OWI состоит из следующих блоков:

- REGISTERS – блок для хранения управляющих данных и статусов;
- BUFFER_TX – буфер передатчика;
- SHIFT_TX – сдвиговый регистр передатчика;
- BUFFER_RX – буфер приемника;
- SHIFT_RX – сдвиговый регистр приемника;
- CLOCK_DIVIDER – делитель частоты;
- 1-WIRE_TIMING_AND_CONTROL – основной блок интерфейса, содержащий управляющие автоматы и отвечающий за корректное функционирование.

Сигнализация шины 1-Wire

Модуль OWI поддерживает следующие формы сигналов: цикл сброса/обнаружения присутствия, тайм-слот записи нуля, тайм-слот записи единицы и тайм-слот чтения.

Цикл сброса/обнаружения присутствия состоит из двух этапов: в течение времени t_{RSTL} ведущий подтягивает линию к низкому уровню, далее начинается окно t_{RSTH} , в течение которого после задержки t_{PDH} ведомое устройство переводит линию к низкому уровню на промежуток времени t_{PDL} , чтобы сообщить о своем присутствии. Для проверки импульса присутствия модуль OWI сначала ожидает t_{PDW} , а затем в течение t_{PDSW} производит выборку напряжения вывода 1W_IO.

Модуль OWI позволяет включить маскирование импульса присутствия, генерируемого ведомым устройством, установив бит PPM регистра OWI_CTRL. При установленном бите PPM модуль OWI генерирует импульс присутствия (пунктирная линия), который начинается с конца t_{PPMS} и длится до конца t_{PPME} .

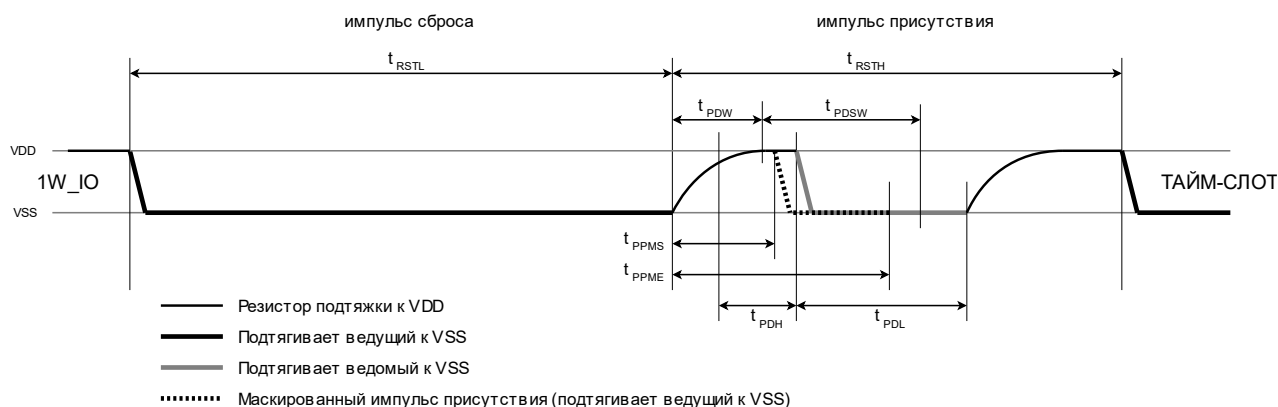


Рисунок 19. Цикл сброса/обнаружения присутствия

Тайм-слот записи нуля состоит из двух этапов: ведущий подтягивает линию в течение t_{W0L} , далее следует время восстановления t_{REC0} . Сумма времени t_{W0L} и t_{REC0} равна длительности тайм-слота t_{SLOT} .

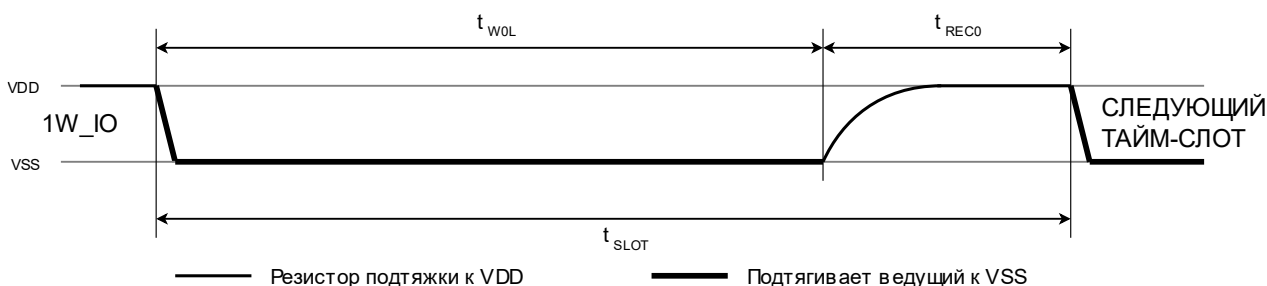


Рисунок 20. Тайм-слот записи нуля

Тайм-слот записи единицы также состоит из двух этапов: ведущий подтягивает линию в течение t_{W1L} , затем ожидается завершение тайм-слота.

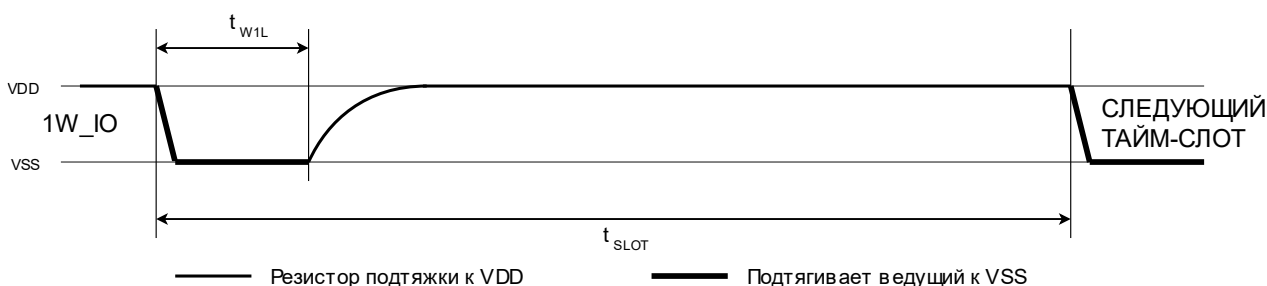


Рисунок 21. Тайм-слот записи единицы

Тайм-слот чтения состоит из трех этапов: ведущий подтягивает линию к низкому уровню в течение t_{W1L} , ведомый устанавливает уровень на линии, после чего ожидается завершение тайм-слота. В течение t_{MRS} модуль OWI осуществляет выборку напряжения вывода 1W_IO. Следует иметь ввиду, что ведомое устройство, отвечая нулем, начинает устанавливать низкий уровень на линии 1W_IO до того, как истечет время t_{W1L} .

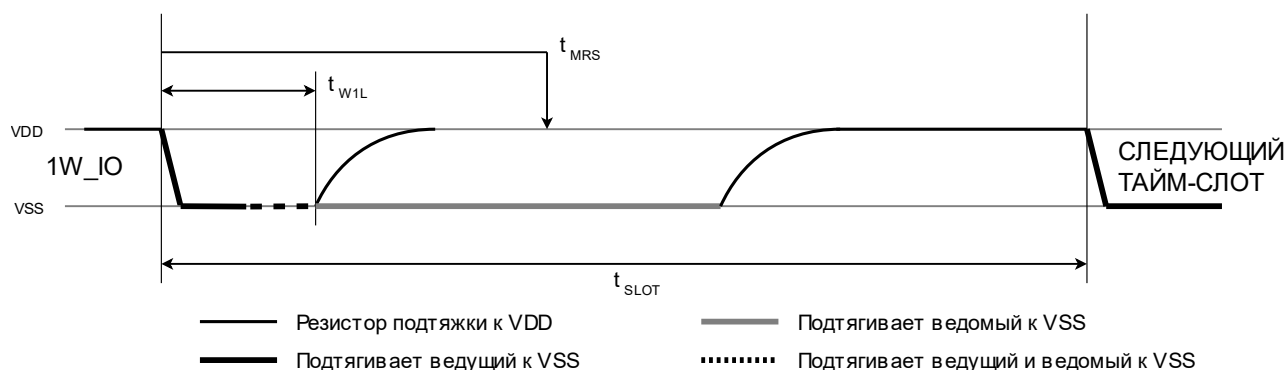


Рисунок 22. Тайм-слот чтения

Таблица 9. Временные характеристики вывода 1W_IO*

Обозначение	Параметр, единица измерения	Условия	Мин.	Макс.
T_{OWI}	Внутренний период синхросигнала модуля, мкс	**	0,8	1
t_{SLOT}	Длительность тайм-слота, мкс	Стандартная скорость передачи, LLM = 0	56	70
		Стандартная скорость передачи, LLM = 1	64	80
		Ускоренная передача	8	10
t_{W0L}	Время низкого уровня тайм-слота записи нуля (Write-zero Low time), мкс	Стандартная скорость передачи	48	60
		Ускоренная передача	6,4	8
t_{W1L}	Время низкого уровня тайм-слота записи единицы/чтения (Write-one/read Low time), мкс	Стандартная скорость передачи, LLM = 0	4,8	6
		Стандартная скорость передачи, LLM = 1	6,4	8
		Ускоренная передача	0,8	1
t_{MRS}	Время выборки тайм-слота чтения (Master Read Sample time), мкс	Стандартная скорость передачи, LLM = 0	12	15
		Стандартная скорость передачи, LLM = 1	19,2	24
		Ускоренная передача	1,6	2
t_{RECO}	Время восстановления уровня тайм-слота записи нуля (Write-zero Recovery time), мкс	Стандартная скорость передачи, LLM = 0	8	10
		Стандартная скорость передачи, LLM = 1	16	20
		Ускоренная передача	1,6	2
t_{RSTL}	Время низкого уровня цикла сброса (Reset Low time), мкс	Стандартная скорость передачи	480	600
		Ускоренная передача	56	70
t_{RSTH}	Время высокого уровня цикла сброса (Reset High time), мкс	Стандартная скорость передачи	384	480
		Ускоренная передача	46,4	58
t_{PDW}		Стандартная скорость передачи	8	10

Обозначение	Параметр, единица измерения	Условия	Мин.	Макс.
	Время ожидания цикла присутствия (Presence-Detect Wait time), мкс	Ускоренная передача	1,6	2
t_{PDSW}	Время выборки цикла присутствия (Presence-Detect Sample Window), мкс	Стандартная скорость передачи, LLM = 0	48,8	61
		Стандартная скорость передачи, LLM = 1	60,8	76
		Ускоренная передача	6,4	8
t_{PPMS}	Время начала маскированного цикла присутствия (Presence-Pulse Mask Start), мкс	Стандартная скорость передачи	16	20
t_{PPME}	Время завершения маскированного цикла присутствия (Presence-Pulse Mask End), мкс	Стандартная скорость передачи	72	90

* – Временные характеристики зависят от задержек на универсальных портах ввода-вывода. Если порты работают медленно, то эти характеристики изменятся соответствующим образом.

** – Для достижения данного периода делитель входной системной частоты настроить в соответствии с таблицей 11.

Предварительная настройка

Перед началом работы с модулем OWI настроить предделитель и делитель частоты и включить систему тактирования в регистре OWI_PRCR. При необходимости использовать порывания задать соответствующие значение в регистр маски OWI_MSK. После этого модуль готов к работе на стандартной скорости и в байтовом режиме.

Генерация цикла сброса/обнаружения присутствия

Для генерации цикла сброса/обнаружения присутствия записать код 01h в регистр OWI_CFG. Если включено прерывание по биту PD в регистре OWI_MSK, то микроконтроллер может продолжить работу до завершения цикла. После завершения прочитать бит PDR регистра OWI_ST, чтобы установить, был ли обнаружен импульс присутствия.

Передача байта

Чтобы передать байт по шине 1-Wire необходимо записать его по адресу OWI_BUF в буфер передатчика. Затем данные переходят в сдвиговый регистр передатчика, где они последовательно отправляются на шину младшим битом вперед (LSB). Соответственно новый байт данных может быть записан в буфер передатчика. Как только сдвиговый регистр передатчика станет пустым, данные будут снова переданы из буфера, и процесс повторится. Каждый из этих регистров имеет флаг (регистр OWI_ST), на основе которого можно вызвать прерывание (регистр OWI_MSK). Бит TBE статусного регистра устанавливается, когда буфер передатчика пуст и готов принять новые данные. Как только происходит запись байта в буфер передатчика, TBE очищается. Бит TЕМT статусного регистра устанавливается, когда сдвиговый регистр пуст и готов принять новые данные. Как только байт передается из буфера в сдвиговый регистр, бит TЕМT очищается, а TBE устанавливается.

Прием байта

Для считывания данных с шины 1-Wire необходимо подготовить ведомое устройство (команды, ранее полученные от ведущего). Процесс приема данных аналогичен передаче. Ведущий начинает прием, записывая FFh по адресу OWI_BUF в буфер передатчика. Данные, которые затем передаются в сдвиговый регистр приемника – это побитовое «И» переданных и принятых данных. После того,

как сдвиговый регистр приемника заполняется, данные передаются в буфер приемника, откуда они могут быть прочитаны по адресу OWI_BUF. Следующие байты от ведомого устройства могут быть прочитаны путем повторной записи FFh в буфер приемника. Если ведомый не был готов передать данные, то полученные данные будут идентичны переданным. На основе состояния буфера приемника и сдвигового регистра приемника также могут генерироваться прерывания. Бит RBF статусного регистра устанавливается, когда данные из сдвигового регистра переданы в буфер, а очищается после чтения буфера. Пока флаг RBF установлен, дальнейшие передачи по шине 1-Wire выполняться не должны, иначе данные могут быть потеряны, так как произойдет перезапись буфера приемника следующим принятым байтом.

Битовый режим

Чтобы активировать битовый режим необходимо записать в бит BIT_CTL регистра OWI_CTRL «1». Последующая работа такая же, как и в байтовом режиме, однако важен только младший бит в буфере приемника/передатчика.

Ускоренная передача

Переключение между скоростями происходит посредством бита OD регистра OWI_CTRL. Любые операции на шине 1-Wire после обновления бита OD происходят на новой скорости. Режим длинной линии относится только к стандартной скорости.

Режим ускоренного поиска ПЗУ (Search ROM Accelerator)

Прежде чем перейти в режим ускоренного поиска ПЗУ необходимо выполнить цикл сброса/обнаружения присутствия и отправить команду поиска ПЗУ (Search ROM – F0h) в байтовом режиме. Буфер приемника не должен содержать данных. Далее можно переходить в режим ускоренного поиска ПЗУ, переведя бит SRA регистра OWI_CFG в «1».

Для завершения одной процедуры поиска ведущий должен передать 16 байт, построенных согласно таблице 10. Каждый байт, записанный в регистр передатчика, содержит полубайт данных решений несоответствий. В конце каждой процедуры ведущий узнает идентификатор одного ведомого устройства.

Таблица 11. Последовательность переданных/принятых байт в режиме ускоренного поиска ПЗУ

№ байта	Буфер	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
1	Передатчик	r ₃	x	r ₂	x	r ₁	x	r ₀	x
	Приемник	ID ₃	d ₃	ID ₂	d ₂	ID ₁	d ₁	ID ₀	d ₀
2	Передатчик	r ₇	x	r ₆	x	r ₅	x	r ₄	x
	Приемник	ID ₇	d ₇	ID ₆	d ₆	ID ₅	d ₅	ID ₄	d ₄
...									
16	Передатчик	r ₆₃	x	r ₆₂	x	r ₆₁	x	r ₆₀	x
	Приемник	ID ₆₃	d ₆₃	ID ₆₂	d ₆₂	ID ₆₁	d ₆₁	ID ₆₀	d ₆₀

x – биты, которые не имеют значения;

r_n – биты, которые выдаст ведущий при несоответствии идентификаторов;

d_n – флаги несоответствия (равен «1», если есть несоответствие или нет ответа);

ID_n – биты фактически выбранного идентификатора.

Для каждого бита (с 0 по 63) ведущий генерирует следующие три тайм-слота:

b1 – прием данных (истинный бит);

b2 – прием данных (инверсный бит);

b3 – передача данных.

В третьем временном интервале b3 ведущий передает «1» и «0» следующим образом:

b3 = r_n , если у двух и более устройств разные значения n-ого бита;

b3 = b1, если у всех устройств одинаковый n-ый бит;

b3 = «1», в случае если есть ошибка (нет ответа).

Бит, переданный в тайм-слоте b3, идентичен ID_n.

Для начала выполнения поиска ПЗУ во все r_n биты пишутся нули. В случае возникновения ошибки все последующие биты ID_n будут равны «1» до тех пор, пока бит SRA не будет деактивирован. Таким образом, если биты ID₆₃ и d₆₃ равны «1», то произошла ошибка во время процедуры поиска и необходимо повторить последнюю последовательность. В противном случае, все ID_n (с 0 до 63) – это идентификатор ПЗУ устройства, которое было найдено и адресовано. После завершения поиска ПЗУ нужно очистить бит SRA для вывода ведущего из режима Search ROM Accelerator. Для поиска следующего устройства повторно используется предыдущий набор r_n ($n = 0 \dots 63$), но r_m устанавливается в «1», где m – наивысший порядок флага несоответствия. Все r_i , где $i > m$, устанавливаются в «0». Этот процесс повторяется до тех пор, пока не произойдет расхождение на одной и той же битовой позиции в течение двух проходов.

Пример использования режима ускоренного поиска ПЗУ

В данном примере рассматривается процесс поиска четырех различных устройств на шине 1-Wire. Они имеют следующие идентификаторы (представлены только младшие значащие биты от 0 до 8):

ROM1 = 00110101...

ROM2 = 10101010...

ROM3 = 11110101...

ROM4 = 00010001...

1) Микроконтроллер инициализирует импульс сброса, записывая 01h в регистр OWI_CFG. Все подчиненные устройства отвечают импульсом обнаружением присутствия.

2) Микроконтроллер отправляет команду поиска ПЗУ (Search ROM), записывая F0h в буфер передатчика. Необходимо дождаться установления статуса RBF и считать (освободить) буфер приемника.

3) Микроконтроллер переводит модуль в режим Search ROM Accelerator, записывая 02h в регистр OWI_CFG.

4) Микроконтроллер записывает 00h в буфер передатчика и считывает возвращаемые данные из буфера приемника. Этот процесс повторяется для всех 16 байт. Считанные данные будут содержать в ID_n ROM4, а «1» установлена в d₀ (r_n содержит «0», следовательно, ROM2 и ROM3 переходят в состояние ожидания) и d₂ (аналогично первому случаю ROM1 переходит в состояние ожидания). Принятые данные выглядят следующим образом (ID_n подчеркнуты, наиболее значащий флаг несоответствия **выделен**):

полученные данные (1) = 100010010000001...

5) Микроконтроллер преобразует полученные данные в идентификатор устройства путем устранения лишних битов.

6) Микроконтроллер записывает 00h в регистр OWI_CFG для выхода из режима ускоренного поиска ПЗУ. После этого ведущий может продолжить работу с адресованным ведомым.

7) Для нахождения следующего устройства шаги с 1-ого по 6-ой повторяются, только на этот раз в передаваемой последовательности наиболее значащий бит решения несоответствия (r_2 в данном случае) инвертируется, и все данные, следующие за ним, устанавливаются в ноль. Полученные данные содержат идентификатор ROM1, а d₀ и d₂ снова установлены в «1»:

отправленные данные (2) = 0000010000000000...

полученные данные (2) = 1000110100010001...

8) Так как наиболее значащий флаг несоответствия остался прежним (d_2), то нужно использовать следующий наиболее значащий (d_0). Процесс повторяется. Следующие итерации выглядят следующим образом:

отправленные данные (3) = 0100000000000000...

полученные данные (3) = 1110010001000100...

отправленные данные (4) = 0101000000000000...

полученные данные (4) = 1111010100010001...

9) Так как наиболее значащий флаг несоответствия (d_2) не изменился, то переходим к следующему (d_0). Однако, d_0 обработан дважды, а менее значимых флагов несоответствия больше нет. Следовательно поиск завершен, найдены идентификаторы всех четырех устройств.

Статусы и прерывания

В отличие от остальных модулей в OWI прерывание по биту-событию не формируется от последующих событий после перехода бита в «1», пока он не будет сброшен чтением обратно в «0». Остальные принципы работы, описанные в разделе «Типы статусов и прерывания», остаются неизменными.

Регистры OWI

№	Аббревиатура	Доступ	Описание
2B00h	OWI_CFG	RW*	Регистр конфигурации
2B04h	OWI_BUF	RW	Регистр буфера приемника/передатчика
2B08h	OWI_ST	R	Регистр статуса
2B0Ch	OWI_MSK	RW	Регистр маски
2B10h	OWI_PRCR	RW	Регистр делителя частоты
2B14h	OWI_CTRL	RW	Регистр управления

* – Не все регистры доступны для записи.

OWI_CFG

Данный регистр используется для генерации цикла сброса/обнаружения присутствия 1-Wire, а также для активации или деактивации режима ускоренного поиска ПЗУ (Search ROM Accelerator), который позволяет избежать использования одноканальной операции с шиной (бит FOW) во время процедуры поиска. В дополнение к этим двум функциям регистр содержит 2 бита для непосредственного управления выводом 1W_IO.

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв				OW_IN	FOW	SRA	1WR
Начальное значение	0				1W_IO	0		

OW_IN (1-Wire Input) – текущее значение на линии 1-Wire (только чтение):

1 – значение на линии соответствует логической единице;

0 – значение на линии соответствует логическому нулю.

Используется в сочетании с битом FOW для выполнения пользовательских функций. Например, для чтения необходимо изменить FOW на «0», затем подождать необходимое количество времени и установить состояние OW_IN.

FOW (Force 1-Wire) – прямое управление линией 1-Wire:

1 – переводит линию в состояние логического нуля;

0 – оставляет линию в состоянии высокого импеданса.

Данную функцию необходимо разрешить путем записи бита EN_FOW регистра OWI_CTRL. В противном случае изменение FOW не влияет на состояние шины. Если EN_FOW и FOW равны «1», то генерация импульса сброса и функция передачи/приема (регистр OWI_BUF) не работают. Изменение EN_FOW на «0» имеет тот же эффект, что и запись в FOW «0».

SRA (Search ROM Accelerator) – включение режима ускоренного поиска ПЗУ:

1 – модуль работает в режиме ускоренного поиска ПЗУ;

0 – модуль работает в нормальном режиме.

1WR (1-Wire Reset) – управление генерацией последовательности сброса/обнаружения присутствия на линии 1-Wire:

1 – инициализировать генерацию цикла сброса/обнаружения присутствия;

0 – цикл сброса/обнаружения присутствия не генерируется.

Бит переводится в «0» после завершения генерации цикла сброса/обнаружения присутствия. Цикл можно прервать, записав «0».

OWI_BUF

Все передаваемые и принимаемые данные проходят соответственно через буфер передатчика и приемника. Запись по данному адресу подключает к шине данных буфер передатчика, а считывание подключает к шине данных буфер приемника.

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

DATA – данные на передачу или принятые данные. Данные могут передаваться в байтовом или битовом режиме в зависимости от бита BIT_CTL регистра OWI_CTRL. В битовом режиме имеет значение только младший бит.

OWI_ST

Бит	7	6	5	4	3	2	1	0
Назначение	OW_LOW	OW_SHORT	RSRF	RBF	TEMT	TBE	PDR	PD
Тип статуса	EVENT	EVENT	FLAG					EVENT
Начальное значение	0				1	1	1	0

OW_LOW – принят импульс присутствия от ведомого устройства. Ведомое устройство при подключении к незанятой шине генерирует спонтанный импульс присутствия, который устанавливает этот бит. Любой переход в «0» на незанятой шине также устанавливает этот бит.

OW_SHORT – короткое замыкание на шине. Перед началом импульса сброса или тайм-слота ведущий проверяет, находится ли шина в режиме ожидания на высоком уровне. Если в этот момент уровень шины низкий, то связь невозможна.

RSRF (Receive Shift Register Full) – сдвиговый регистр приемника заполнен. Когда этот бит равен «0», то это указывает на то, что сдвиговый регистр приемника не содержит данных или осуществляет их прием. После передачи данных в буфер приемника этот бит сбрасывается аппаратно.

RBF (Receive Buffer Full) – буфер приемника заполнен. Флаг переходит в «1», когда есть новые данные, полученные из сдвигового регистра приемника. Если бит равен «0», то это указывает, что буфер приемника не содержит новых данных. Этот бит сбрасывается аппаратно после чтения буфера приемника.

TEMT (Transmit Shift Register Empty) – сдвиговый регистр передатчика пуст и готов принять новые данные. Флаг устанавливается в «1», когда регистр сдвига передатчика не содержит данных. После передачи данных из буфера передатчика в сдвиговый регистр передатчика этот бит переходит в «0». Если бит равен «0», то это указывает на то, что регистр сдвига передатчика занят передачей данных.

TBE (Transmit Buffer Empty) – буфер передатчика пуст и готов принять новые данные. Флаг устанавливается в «1», когда буфер передатчика не содержит данных. После записи данных в буфер передатчика этот бит переходит в «0». Если бит равен «0», то это указывает на то, что буфер передатчика ждет окончания сдвига данных из сдвигового регистра передатчика.

PDR (Presence Detect Result) – результат цикла обнаружения присутствия:

1 – импульс присутствия не обнаружен;

0 – обнаружен импульс присутствия.

PD (Presence Detect) – цикл сброса/обнаружения присутствия завершен, а результат находится в бите PDR. При использовании прерывания по данному биту оно должно быть включено в OWI_MSK до инициализации цикла сброса.

OWI_MSK

Бит	7	6	5	4	3	2	1	0
Назначение	OW_LOW	OW_SHORT	RSRF	RBF	TEMT	TBE	Резерв	PD
Начальное значение	0							

Для каждого из битов справедливо:

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

OWI_PRCR

Модуль OWI требует внутренней тактовой частоты 1 МГц. Эта частота генерируется от тактовой частоты системы f_{clk} . Регистр OWI_PRCR позволяет управлять делителем и передделителем тактовой частоты с целью максимально приблизить f_{clk} к 1 МГц. Систему тактирования модуля необходимо настроить и включить до начала передачи по шине 1-Wire.

Бит	7	6	5	4	3	2	1	0
Назначение	CLK_EN	Резерв			DIV		PRE	
Начальное значение	0							

CLK_EN – включение системы тактирования OWI:

- 1 – система тактирования включена;
- 0 – система тактирования выключена.

DIV – выбор коэффициента делителя частоты:

- 111b – $f_{clk}/128$;
- 110b – $f_{clk}/64$;
- 101b – $f_{clk}/32$;
- 100b – $f_{clk}/16$;
- 011b – $f_{clk}/8$;
- 010b – $f_{clk}/4$;
- 001b – $f_{clk}/2$;
- 000b – f_{clk} .

PRE – выбор коэффициента предделителя частоты:

- 11b – $f_{clk}/7$;
- 10b – $f_{clk}/5$;
- 01b – $f_{clk}/3$;
- 00b – f_{clk} .

Делитель и предделитель работают совместно.

Таблица 12. Настройка коэффициентов деления частоты

Мин. f_{clk} , МГц	Макс. f_{clk} , МГц	Макс. погрешность f_{clk} , %	Коэф. деления	DIV			PRE	
				2	1	0	1	0
1	1,25	25	1	0	0	0	0	0
2	2,5	25	2	0	0	1	0	0
3	3,75	25	3	0	0	0	0	1
4	< 5	25	4	0	1	0	0	0
5	< 6	20	5	0	0	0	1	0
6	< 7	17	6	0	0	1	0	1
7	< 8	14	7	0	0	0	1	1
8	< 10	25	8	0	1	1	0	0

OWI_CTRL

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв	OD	BIT_CTL	Резерв*		EN_FOW	PPM	LLM
Начальное значение	0							

* – Не изменять начальное значение.

OD (Overdrive) – режимом ускоренной передачи:

1 – режим ускоренной передачи включен;

0 – режим ускоренной передачи выключен.

BIT_CTL – выбор битового или байтового режима:

1 – режим битовой передачи включен;

0 – режим байтовой передачи выключен.

Следовательно, в битовом режиме соответствующие биты в статусном регистре устанавливаются после 1 тайм-слота, а не после 8 тайм-слотов, как в байтовом режиме.

EN_FOW (Enable Force 1-Wire) – разрешение прямого управления линией 1-Wire:

1 – разрешено управление линией 1-Wire посредством бита FOW регистра OWI_CFG;

0 – прямое управление линией 1-Wire запрещено.

PPM (Presence Pulse Masking) – режим маскирования импульса присутствия:

1 – маскирование импульса присутствия включено;

0 – маскирование импульса присутствия выключено.

В условиях длинной линии импульсы присутствия, генерируемые ведомыми устройствами, могут вызывать помехи на линии. Маскирование импульса присутствия может предотвратить неблагоприятные эффекты, вызванные импульсом присутствия. Если PPM равен «1», то результат цикла обнаружения присутствия (бит PDR регистра OWI_ST) всегда равен «0», даже если на шине нет ведомого устройства.

LLM (Long Line Mode) – режим длинной линии:

1 – режим длинной линии включен;

0 – режим длинной линии выключен.

Включение режима длинной линии изменяет стандартные временные характеристики, чтобы они лучше подходили для связи по длинной линии.

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
2C00h	FSM_CTRL	W	Регистр управления
2C01h	FSM_PRD2	RW	Регистр периода счета 2
2C02h	FSM_PRD1	RW	Регистр периода счета 1
2C03h	FSM_PRD0	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_PRD2

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

PRD – период счета таймера, старшая часть.

FSM_PRD1

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

PRD – период счета таймера, средняя часть.

FSM_PRD0

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.

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».
- текущие 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».
- текущие 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».
- текущие 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
TIMER0			
2D00h	TMR0_CTRL	RW	Регистр управления
2D01h	TMR0_CFG	RW	Регистр конфигурации
2D02h	TMR0_PRDH	RW	Период счета таймера, старшая часть
2D03h	TMR0_PRDM	RW	Период счета таймера, средняя часть
2D04h	TMR0_PRDL	RW	Период счета таймера, младшая часть
2D05h	TMR0_VALH	R	Текущее значение таймера, старшая часть
2D06h	TMR0_VALM	R	Текущее значение таймера, средняя часть
2D07h	TMR0_VALL	R	Текущее значение таймера, младшая часть
2D08h	TMR0_MSK	RW	Регистр маски прерываний
2D09h	TMR0_ST	R	Регистр статусов
TIMER1			
2D20h	TMR1_CTRL	RW	Регистр управления
2D21h	TMR1_CFG	RW	Регистр конфигурации
2D22h	TMR1_PRDH	RW	Период счета таймера, старшая часть
2D23h	TMR1_PRDM	RW	Период счета таймера, средняя часть
2D24h	TMR1_PRDL	RW	Период счета таймера, младшая часть
2D25h	TMR1_VALH	R	Текущее значение таймера, старшая часть
2D26h	TMR1_VALM	R	Текущее значение таймера, средняя часть
2D27h	TMR1_VALL	R	Текущее значение таймера, младшая часть
2D28h	TMR1_MSK	RW	Регистр маски прерываний
2D29h	TMR1_ST	R	Регистр статусов
TIMER2			
2D40h	TMR2_CTRL	RW	Регистр управления
2D41h	TMR2_CFG	RW	Регистр конфигурации
2D42h	TMR2_PRDH	RW	Период счета таймера, старшая часть
2D43h	TMR2_PRDM	RW	Период счета таймера, средняя часть
2D44h	TMR2_PRDL	RW	Период счета таймера, младшая часть
2D45h	TMR2_VALH	R	Текущее значение таймера, старшая часть
2D46h	TMR2_VALM	R	Текущее значение таймера, средняя часть
2D47h	TMR2_VALL	R	Текущее значение таймера, младшая часть
2D48h	TMR2_MSK	RW	Регистр маски прерываний
2D49h	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	Резерв				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	Резерв				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	Резерв				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.

ЦАП**Общая информация**

12-разрядный R-2R ЦАП.

Алгоритм работы

Процедура работы с ЦАП:

- в регистре DAC_CTRL разрешить подачу синхросигнала на ЦАП битом EN;
- отправлять данные на преобразование в регистры DAC_VALUE1 и DAC_VALUE0;
- выключить ЦАП битом EN или продолжить работу записав новые данные.

Регистры модуля «ЦАП»

№	Аббревиатура	Доступ	Описание
3000h	DAC_CTRL	RW	Регистр управления
3004h	DAC_VALUE0	RW	Данные на преобразование 0
3005h	DAC_VALUE1	RW	Данные на преобразование 1

DAC_CTRL

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

EN – включение ЦАП:

1 – ЦАП включен;

0 – ЦАП выключен.

DAC_VALUE0

Данные обновятся на входе ЦАП после записи в данный регистр.

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

VALUE – данные на преобразование, младшая часть.

DAC_VALUE1

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

VALUE – данные на преобразование, старшая часть.

Контроллер прерываний (INT_CTRL)

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

Регистры «Контроллера прерываний»

№	Аббревиатура	Доступ	Описание
3100h	INT_FIX_CLR0	RW	Регистр зафиксированных прерываний, группа 0
3101h	INT_FIX_CLR1	RW	Регистр зафиксированных прерываний, группа 1
3102h	INT_FIX_CLR2	RW	Регистр зафиксированных прерываний, группа 2
3103h	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	I2_FIX	Резерв	I0_FIX
Начальное значение	0							

I3_FIX – прерывание CMM.

I2_FIX – прерывание WDT.

I0_FIX – прерывание WORK_FSM.

Запись в **Ix_FIX** регистра INT_FIX_CLR:

1 – сбросить прерывание;

0 – не менять текущую настройку.

Чтение **Ix_FIX** регистра INT_FIX_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

INT_FIX_CLR1

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв			I4_FIX	I3_FIX	I2_FIX	I1_FIX	I0_FIX
Начальное значение	0							

I4_FIX – прерывание I2C.

I3_FIX – прерывание SPI0.

I2_FIX – прерывание UART0.

I1_FIX – прерывание TIMER0.

I0_FIX – прерывание GPIOA.

Запись в **Ix_FIX** регистра INT_FIX_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix_FIX** регистра INT_FIX_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

INT_FIX_CLR2

Бит	7	6	5	4	3	2	1	0
Назначение	Резерв			I4_FIX	I3_FIX	I2_FIX	I1_FIX	I0_FIX
Начальное значение	0							

I4_FIX – прерывание OWI.

I3_FIX – прерывание SPI1.

I2_FIX – прерывание UART1.

I1_FIX – прерывание TIMER1.

I0_FIX – прерывание GPIOB.

Запись в **Ix_FIX** регистра INT_FIX_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix_FIX** регистра INT_FIX_CLR:

1 – зафиксировано прерывание;

0 – прерывание отсутствует.

INT_FIX_CLR3

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

I1_FIX – прерывание TIMER2.

I0_FIX – прерывание GPIOC.

Запись в **Ix_FIX** регистра INT_FIX_CLR:

1 – сбросить прерывание;

0 – не меняет текущую настройку.

Чтение **Ix_FIX** регистра INT_FIX_CLR:

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, а также ПЗУ, ФАПЧ и МОУ перейдут 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.

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	24
JTAG_ROM_ACCESS	100b	24
IF_ACCESS	101b	18
DEBUGGER	110b	5
BYPASS	111b	1

Структурная схема

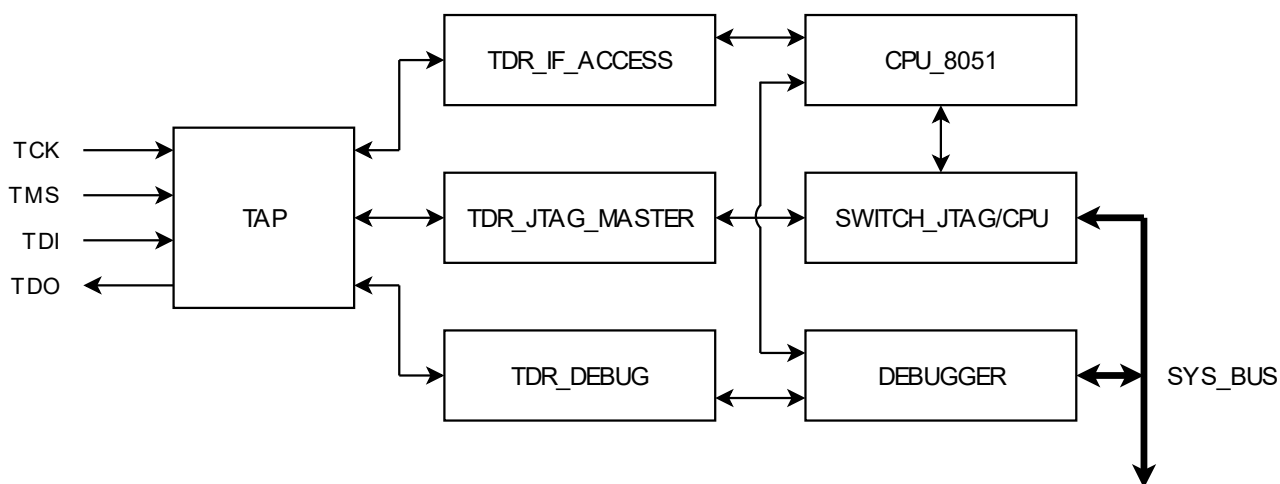


Рисунок 24. Структурная схема модуля JTAG с подключенными к нему блоками

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_DEBUGGER – 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:

Бит	—							24
Назначение	—							SYNCHR
Начальное значение	—							0
Бит	23	22	21	20	19	18	17	16
Назначение	RW	CE_RAM	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_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.

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.

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	0							
Бит	15	14	13	12	11	10	9	8
Назначение	Резерв							
Начальное значение	0							
Бит	7	6	5	4	3	2	1	0
Назначение	DATA							
Начальное значение	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.

ANALOG_CFG

Регистры модуля ANALOG_CFG подключены к конфигурационному однократно программируемому ПЗУ. В тестовом режиме, когда $TM = 1$ (Test Mode), в зависимости от вывода $GPIOB<0>/H_S$ источником конфигурационных данных могут быть, либо регистры, либо ПЗУ («1» – «SOFT» и «0» – «HARD» режим соответственно). В «SOFT» режиме при подаче 9 В на вывод VPP_9V конфигурационное ПЗУ прожигается. При $TM = 0$ конфигурационная память будет работать в режиме «HARD», а вывод $GPIOB<0>/H_S$ как порт ввода-вывода.

Регистры модуля ANALOG_CFG

№	Аббревиатура	Доступ	Описание
3300h	ANALOG_BUF	W	Регистр управления выходным буфером ЦАП, управления выходным масштабирующим операционным усилителем (МОУ) после ИОН (вывод VRP), настройки коэффициента усиления МОУ
3301h	ANALOG_PLL	W	Регистр настройки источника тактовой частоты микроконтроллера
3302h	ANALOG_RC	W	Регистр настройки емкости конденсатора RC-генератора
3303h	ANALOG_REF	W	Регистр управления конденсаторами частотной коррекции МОУ и настройки выходного напряжения ИОН (вывод OUT_REF)
3305h	ANALOG_IROM	W	Регистр настройки тока считывания ПЗУ
3306h	ANALOG_RC_R	W	Регистр настройки сопротивления резистора RC-генератора, выбора источника BOR, управления выводами $GPIOB<1>$, OUT_REF и источниками тактовой частоты в тестовом режиме

ANALOG_BUF

Бит	7	6	5	4	3	2	1	0
Назначение	DAC_BUF_EN	OA_EN	OA_GAIN_N			OA_GAIN_M		
Начальное значение	0							

DAC_BUF_EN – включение выходного буфера ЦАП:

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

OA_EN – включение выходного МОУ после ИОН (вывод VRP):

- 1 – МОУ включен;
- 0 – МОУ выключен (Z-состояние).

OA_GAIN_N – коэффициент масштабирования N для настройки коэффициента усиления МОУ.

OA_GAIN_M – коэффициент масштабирования M для настройки коэффициента усиления МОУ.

Блок МОУ состоит из ОУ общего назначения и потенциометра, используемого для подбора коэффициентов масштабирования. На вход блока подается выходное напряжение ИОН, которое усиливается в зависимости от масштабирующих коэффициентов.

Если блок выключен, то он находится в режиме ожидания и не потребляет ток. Имеется возможность отключения частотной коррекции.

Таблица 13. Настройка коэффициентов масштабирования

Коэффициент усиления МОУ								
$\begin{matrix} M \\ N \end{matrix}$	000b	001b	010b	011b	100b	101b	110b	111b
000b	2	1,5	1,333333	1,25	1,2	1,166667	1,142857	1,125
001b	3	2	1,666667	1,5	1,4	1,333333	1,285714	1,25
010b	4	2,5	2	1,75	1,6	1,5	1,428571	1,375
011b	5	3	2,333333	2	1,8	1,666667	1,571429	1,5
100b	6	3,5	2,666667	2,25	2	1,833333	1,714286	1,625
101b	7	4	3	2,5	2,2	2	1,857143	1,75
110b	8	4,5	3,333333	2,75	2,4	2,166667	2	1,875
111b	9	5	3,666667	3	2,6	2,333333	2,142857	2

ANALOG_PLL

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

PLL – совместно с ТМ значения в данном регистре определяют и настраивают источник тактового сигнала микроконтроллера XTAL_CLK.

Таблица 14. Определение и настройка источника тактовой частоты микроконтроллера

ТМ	PLL								RC_OUT	RC_CLK	XTAL_CLK
	7	6	5	4	3	2	1	0			
1	0	X	X	X	X	X	X	X	RC	RC	GEN1_EXT
1	1	X	X	X	0	X	X	X	–	RC	GEN1_EXT
1	1	X	X	0	1	K2	K1	K0	GEN1ext+ PLL	RC	GEN1_EXT
1	1	X	X	1	1	K2	K1	K0	RC+ PLL	RC	GEN1_EXT
0	X	0	0	X	0	X	X	X	–	RC	GEN1_EXT
0	X	0	1	X	0	X	X	X	–	RC	GEN1_QV
0	X	1	X	X	0	X	X	X	–	RC	RC
0	X	X	0	0	1	K2	K1	K0	–	RC	GEN1_EXT+PLL
0	X	X	1	0	1	K2	K1	K0	–	RC	GEN1_QV+PLL
0	X	X	X	1	1	K2	K1	K0	–	RC	RC+PLL

Используемые в таблице 13 обозначения:

x – биты, которые не имеют значения;

GEN1_EXT – внешняя тактовая частота с вывода GEN1 (источник – внешний генератор);

GEN1_QV – тактовая частота с вывода GEN1 (источник – встроенный генератор на основе внешнего кварцевого резонатора);

GEN1_EXT+PLL – внешняя тактовая частота, пропущенная через ФАПЧ;

GEN1_QV+PLL – частота с кварцевого генератора, пропущенная через ФАПЧ;

RC+PLL – частота с RC-генератора, пропущенная через ФАПЧ;

RC – тактовая частота с выхода встроенного RC-генератора;

K2, K1, K0 – коэффициент умножения ФАПЧ.

При записи в биты RC_C регистра ANALOG_RC максимального значения 7Fh, а в биты RC_R регистра ANALOG_RC_R минимального значения 0 – частота RC-генератора будет не более 90 кГц.

При записи в биты RC_C регистра ANALOG_RC минимального значения 0, а в биты RC_R регистра ANALOG_RC_R максимального значения 7 – частота RC-генератора будет не менее 400 кГц.

ANALOG_RC

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

RC_C – настройка емкости конденсатора RC-генератора.

ANALOG_REF

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

OA_C_EN – включение конденсаторов частотной коррекции МОУ:

1 – конденсаторы частотной коррекции включены;

0 – конденсаторы частотной коррекции выключены.

V_REF – настройка выходного напряжения ИОН (вывод OUT_REF).

ANALOG_IROM

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

IROM – настройка тока считывания ПЗУ.

ANALOG_RC_R

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

BOR – супервизор питания:

1 – внешний (вывод BOR_EXT/PGM при TM = 0);

0 – внутренний.

RC_R – настройка сопротивления резистора RC-генератора.

[illegible]