

## Приложение 4

### Микроконтроллер AT90S2313 фирмы Atmel

---

---

Материал этого приложения служит дополнением к главе 3 и представляет собой перевод с английского официальной статьи о микроконтроллере AT90S2313. Перевод выполнил Ю. Андриенко (г. Харьков). При подготовке к публикации материал несколько сокращен и переработан. Исходный текст можно найти в Internet по адресу: [www.kirov.ru/~ua4nx/fail/](http://www.kirov.ru/~ua4nx/fail/).

---

AT90S2313 — экономичный 8-битовый КМОП микроконтроллер, построенный с использованием расширенной RISC архитектуры AVR. Исполняя по одной команде за период тактовой частоты, AT90S2313 имеет производительность около 1MIPS на МГц, что позволяет разработчикам создавать системы оптимальные по скорости и потребляемой мощности.

В основе ядра AVR лежит расширенная RISC архитектура, объединяющая развитый набор команд и 32 регистра общего назначения. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), что дает доступ к любым двум регистрам за один машинный цикл.

Подобная архитектура обеспечивает десятикратный выигрыш в эффективности кода по сравнению с традиционными CISC микроконтроллерами.

AT90S2313 предлагает следующие возможности: 2кБ загружаемой флэш-памяти; 128 байт EEPROM; 15 линий ввода/вывода общего назначения; 32 рабочих регистра; настраиваемые таймеры/счетчики с режимом совпадения; внешние и внутренние прерывания; программируемый универсальный последовательный порт; программируемый сторожевой таймер со встроенным генератором; SPI последовательный порт для загрузки программ; два выбираемых программно режима низкого энергопотребления.

Холостой режим (Idle Mode) отключает ЦПУ, оставляя в рабочем состоянии регистры, таймеры/счетчики, SPI порт и систему прерываний. Экономичный режим (Power Down Mode) сохраняет содержимое регистров, но отключает генератор, запрещая функционирование всех встроенных устройств до внешнего прерывания или аппаратного сброса.

Микросхемы производятся с использованием технологии энергонезависимой памяти высокой плотности фирмы Atmel. Загружаемая флэш-память на кристалле может быть перепрограммирована прямо в системе через последовательный интерфейс SPI или доступным программатором энергонезависимой памяти. Объединяя на одном кристалле усовершенствованный 8-битовый RISC процессор с загружаемой флэш-памятью, AT90S2313 является мощным микроконтроллером, который позволяет создавать достаточно гибкие и эффективные по стоимости устройства.

AT90S2313 поддерживается полной системой разработки включающей в себя макроассемблер, программный отладчик/симулятор, внутрисхемный эмулятор и отладочный комплект.

## Описание выводов

**VCC** — вывод источника питания.

**GND** — земля.

**Port B (PB7–PB0)** — порт **B** является 8-битовым двунаправленным портом ввода/вывода. Для выводов порта предусмотрены внутренние подтягивающие резисторы (выбираются для каждого бита). Выводы PB0 и PB1 также являются положительным (AIN0) и отрицательным (AIN1) входами встроенного аналогового компаратора. Выходные буферы порта **B** могут поглощать ток до 20 мА и непосредственно управлять светодиодными индикаторами. Если выводы PB0..PB7 используются как входы и извне устанавливаются в низкое состояние, они являются источниками тока, если включены внутренние подтягивающие резисторы. Кроме того, Порт **B** обслуживает некоторые специальные функции, которые будут описаны ниже.

На рис. П4.1 показаны расположения выводов микроконтроллера AT90S2313.

**Port D (PD6..PD0)** — порт **D** является 7-битовым двунаправленным портом с внутренними подтягивающими резисторами. Выходные буферы порта **D** могут поглощать ток до 20мА. Как входы установленные в низкое состояние, выводы порта **D** являются источниками тока, если задействованы подтягивающие резисторы. Кроме того Порт **D** обслуживает некоторые специальные функции, которые будут описаны ниже.

**RESET** — вход сброса. Удержание на входе низкого уровня в течение двух машинных циклов (если работает тактовый генератор), сбрасывает устройство.

**XTAL1** — вход инвертирующего усилителя генератора и вход внешнего тактового сигнала.

**XTAL2** — выход инвертирующего усилителя генератора.

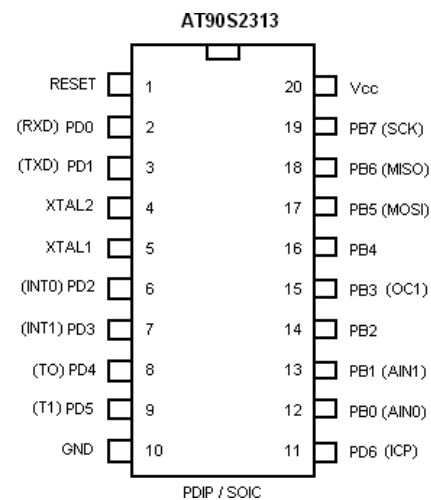


Рис. П4.1. Расположение выводов МК AT90S2313

## Кварцевый генератор

XTAL1 и XTAL2 являются входом и выходом инвертирующего усилителя, на котором можно собрать генератор тактовых импульсов. Можно использовать как кварцевые, так и керамические резонаторы. При подключении внешнего тактового сигнала вывод XTAL2 остается неподключенным, а XTAL1 подключается в выходу внешнего генератора. На рис. П4.2 показаны варианты подключения кварцевого резонатора и внешнего генератора.

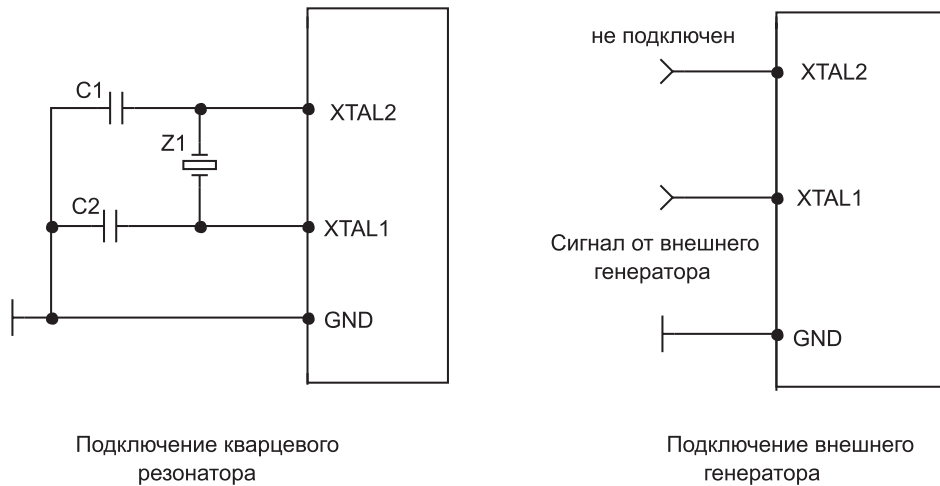


Рис. П4.2. Схемы подключения кварца и генератора

## Обзор архитектуры AT90S2313

Регистровый файл быстрого доступа содержит 32 8-разрядных регистра общего назначения, доступ к которым осуществляется за один машинный цикл. Поэтому за один машинный цикл выполняется одна операция АЛУ. Два операнда выбираются из регистрового файла, выполняется операция, результат ее записывается в регистровый файл — все за один машинный цикл.

Шесть из 32 регистров можно использовать как три 16-разрядных указателя в адресном пространстве данных, что дает возможность использовать высокоэффективную адресную арифметику (16-разрядные регистры X, Y и Z). Один из трех адресных указателей (регистр Z) можно использовать для адресации таблиц в памяти программ. Это X-, Y- и Z-регистры.

АЛУ поддерживает арифметические и логические операции с регистрами, с константами и регистрами. Операции над отдельными регистрами также выполняются в АЛУ.

Кроме регистровых операций, для работы с регистровым файлом могут использоваться доступные режимы адресации, поскольку регистровый файл занимает адреса \$00-\$1F в области данных, обращаться к ним можно как к ячейкам памяти.

Пространство ввода состоит из 64 адресов для периферийных функций процессора, таких как управляющие регистры, таймеры/счетчики и другие. Доступ к пространству ввода/вывода может осуществляться непосредственно, как к ячейкам памяти расположенным после регистрового файла (\$20-\$5F).

Процессоры AVR построены по гарвардской архитектуре с отдельными областями памяти программ и данных. Доступ к памяти программ осуществляется при помощи одноуровневого буфера. Во время выполнения команды, следующая выбирается из памяти программ. Подобная концепция дает возможность выполнять по одной команде за каждый машинный цикл. Память программ — это внутрисистемная загружаемая флэш-память.

При помощи команд относительных переходов и вызова подпрограмм осуществляется доступ ко всему адресному пространству. Большая часть команд AVR имеет размер 16-разрядов, одно слово. Каждый адрес в памяти программ содержит одну 16- или 32-разрядную команду.

При обработке прерываний и вызове подпрограмм адрес возврата запоминается в стеке. Стек размещается в памяти данных общего назначения, соответственно размер стека ограничен только размером доступной памяти данных и ее использованием в программе. Все программы пользователя должны инициализировать указатель стека (SP) в программе выполняемой после сброса (до того как вызываются подпрограммы и разрешаются прерывания). 8-разрядный указатель стека доступен для чтения/записи в области ввода/вывода.

Доступ к 128 байтам статического ОЗУ, регистровому файлу и регистрам ввода/вывода осуществляется при помощи пяти доступных режимов адресации поддерживаемых архитектурой AVR.

Все пространство памяти AVR является линейным и непрерывным.

Гибкий модуль прерываний имеет собственный управляющий регистр в пространстве ввода/вывода, и флаг глобального разрешения прерываний в регистре состояния. Каждому прерыванию назначен свой вектор в начальной области памяти программ. Различные прерывания имеют приоритет в соответствии с расположением их векторов. По младшим адресам расположены векторы с большим приоритетом.

## Файл регистров общего назначения

Все команды оперирующие регистрами прямо адресуются к любому из регистров за один машинный цикл. Единственное исключение — пять команд оперирующих с константами SBCI, SUBI, CPI, ANDI, ORI и команда LDI, загружающая регистр константой. Эти команды работают только со второй половиной регистрового файла — R16-R31. Команды SBC, SUB, CP, AND и OR, также как и все остальные, применимы ко всему регистровому файлу.

Каждому регистру присвоен адрес в пространстве данных, они отображаются на первые 32 ячейки ОЗУ. Хотя регистровый файл физически размещен вне ОЗУ, подобная организация памяти дает гибкий доступ к регистрам. Регистры X, Y и Z могут использоваться для индексации любого регистра.

Кроме обычных функций, регистры R26-R31 имеют дополнительные функции, эти регистры можно использовать как адресные указатели в области памяти данных. Эти регистры обозначаются как X, Y, Z и определены как «Регистр X», «Регистр Y» и «Регистр Z». Схематическое изображение этих регистров показано на рис. П4.3.

При различных режимах адресации эти регистры могут использо-

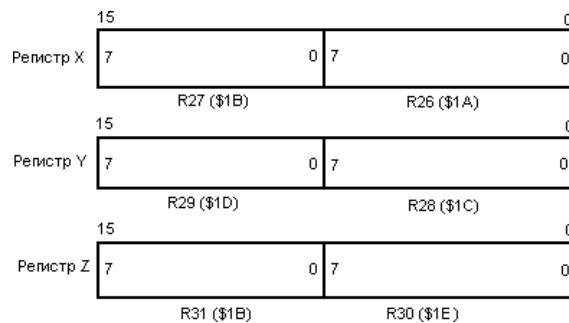


Рис. П4.3. Схемы регистров X, Y и Z

ваться как фиксированный адрес, для адресации с автоинкрементом или с автодекрементом.

## Арифметико-логическое устройство — АЛУ

АЛУ процессора непосредственно подключено к 32 регистрам общего назначения. За один машинный цикл АЛУ производит операции между регистрами регистрового файла. Команды АЛУ разделены на три основных категории — арифметические, логические и битовые.

## Загружаемая память программ

AT90S2313 содержит 2кБ загружаемой флэш-памяти для хранения программ. Поскольку все команды занимают одно 16- или 32-разрядное слово, флэш-память организована как 1К×16. Флэш-память выдерживает не менее 1000 циклов перезаписи.

Программный счетчик имеет ширину 10 бит и позволяет адресоваться к 1024 словам программной флэш-памяти.

Подробно загрузка флэш-памяти будет рассмотрена дальше.

Таблицы констант могут располагаться в диапазоне адресов 0–2К. (см. описание команды LPM).

## EEPROM память данных

AT90S2313 содержит 128 байт электрически стираемой энергонезависимой памяти (EEPROM). EEPROM организована как отдельная область данных, каждый байт которой может быть прочитан и перезаписан. EEPROM выдерживает не менее 100000 циклов записи/стирания. Доступ к энергонезависимой памяти данных рассмотрен ниже и задается регистрами адреса, данных и управления.

Дальше будет рассмотрена загрузка данных в EEPROM через SPI интерфейс.

## Статическое ОЗУ данных

На рис. П4.4 показана организация памяти данных в AT90S2313.

224 ячейки памяти включают в себя регистровый файл, память ввода/вывода и статическое ОЗУ данных.

Первые 96 адресов используются для регистрового файла и памяти ввода/вывода, следующие 128 — для ОЗУ данных.

При обращении к памяти используются пять различных режимов адресации: прямой, непосредственный со смещением, непосредственный, непосредственный с предварительным декрементом и непосредственный с постинкрементом. Регистры R26-R31 регистрового файла используются как указатели на ячейки памяти.

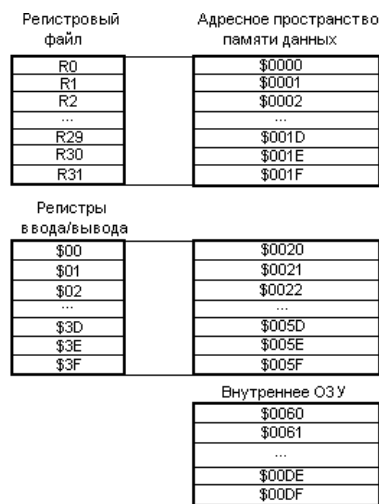


Рис. П4.4. Организация памяти данных AT90S2313

затели для непосредственной адресации.

Прямая адресация имеет доступ ко всей памяти данных.

Непосредственная адресация со смещением используется для доступа к 63-м ячейкам базовый адрес которых задается содержимым регистров Y или Z.

Для непосредственной адресации с инкрементом и декрементом адреса используются адресные регистры X, Y и Z.

При помощи любого из этих режимов производится доступ ко всем 32 регистрам общего назначения, 64 регистрам ввода/вывода и 128 ячейкам ОЗУ.

## Время выполнения команд

ЦПУ процессора AVR управляется системной частотой генерируемой внешним резонатором. Внутреннее деление частоты генератора не используется.

В процессоре организован буфер (pipeline) команд, при выборе команды из памяти программ происходит выполнение предыдущей команды. Подобная концепция позволяет достичь быстродействия 1MIPS на МГц, уникальных показателей стоимости, быстродействия и потребления процессора.

**Таблица П4.1.** Пространство ввода/вывода AT90S2313

Адрес	Наимен.	Назначение	
\$3F(\$5F)	SREG	Status REGister	Регистр состояния
\$3D(\$5D)	SPL	Stack pointer low	Указатель стека, мл. байт
\$3B(\$5B)	GIMSK	General Interrupt MaSK register	Общий регистр маски прерываний
\$3A(\$5A)	GIFR	General Interrupt Flag Register	Общий регистр флагов прерываний
\$39(\$59)	TIMSK	Timer/counter Interrupt mask register	Регистр флага прерывания от таймера/счетчика
\$38(\$58)	TIFR	Timer/counter Interrupt Flag register	Регистр флага прерывания таймера/счетчика
\$35(\$55)	MCUCR	MCU general Control Register	Общий регистр управления микроконтроллером
\$33(\$53)	TCCR0	Timer/Counter 0 Control Register	Регистр управления таймером /счетчиком 0
\$32(\$52)	TCNT0	Timer/Counter 0 (8-бит)	Таймер/счетчик 0 (8 бит)
\$2F(\$4F)	TCCR1A	Timer/Counter 1 Control Register A	Рег. А управления таймером /счетчиком 1
\$2E(\$4E)	TCCR1B	Timer/Counter 1 Control Register B	Рег. В управления таймером /счетчиком 1
\$2D(\$4D)	TCNT1H	Timer/Counter 1 High byte	Таймер/счетчик 1 старший байт
\$2C(\$4C)	TCNT1L	Timer/Counter 1 Low byte	Таймер/счетчик 1 младший байт
\$2B(\$4B)	OCR1H	Output Compare Register 1 high byte	Выход регистра совпадения 1 старший байт
\$2A(\$4A)	OCR1L	Output Compare Register 1 low byte	Выход регистра совпаден. 1 младший бвйт
\$25(\$45)	ICR1H	T/C 1 Input Cupture Register High Byte	Регистр захвата Т/С 1 старший байт
\$24(\$44)	ICR1L	T/C 1 Input Cupture Register Low Byte	Регистр захвата Т/С 1 младший байт
\$21(\$41)	WDTCR	Watchdog Timer Control Register	Регистр управления сторожевым таймером

Таблица П4.1. Окончание

Адрес	Наимен.	Назначение	
\$1E(\$3E)	EEAR	EEPROM Address Register	Регистр адреса энергонезависимой памяти
\$1D(\$3D)	EEDR	EEPROM Data Register	Регистр данных энергонезависимой памяти
\$1C(\$3C)	EECR	EEPROM Control Register	Регистр управления энергонезависимой памяти
\$18(\$38)	PORTB	Data Register, Port B	Регистр данных порта B
\$17(\$37)	DDRB	Data Direction Register Port B	Регистр направления данных порта B
\$16(\$36)	PINB	Input pins, Port B	Выводы порта B
\$12(\$32)	PORTD	Data Register, Port D	Регистр данных порта D
\$11(\$31)	DDRD	Data Direction Register Port D	Регистр направления данных порта D
\$10(\$30)	PIND	Input pins, Port D	Выводы порта D
\$0C(\$2C)	UDR	UART Data Register	Регистр данных последовательного порта
\$0B(\$2B)	USR	UART Status Register	Регистр состояния последовательного порта
\$0A(\$2A)	UCR	UART Control Register	Регистр управления последовательного порта
\$09(\$29)	UBRR	UART Baud Rate Register	Регистр скорости последовательного порта
\$08(\$28)	ACSR	Analog Comparator Control and Status Register	Регистр управления и состояния аналогового компаратора.

Примечание: зарезервированные и неиспользуемые ячейки не показаны.

Все устройства ввода/вывода и периферийные устройства AT90S2313 располагаются в пространстве ввода/вывода. Различные ячейки этого пространства доступны через команды **IN** и **OUT**, пересылающие данные между одним из 32-х регистров общего назначения и пространством ввода/вывода. К регистрам \$00-\$1F можно осуществлять побитовый доступ командами **SBI** и **CBI**. Значение отдельного бита этих регистров можно проверить командами **SBIC** и **SBIS**. Дополнительную информацию по этому вопросу можно найти в описании системы команд.

При использовании специальных команд **IN**, **OUT**, **SBIS** и **SBIC**, должны использоваться адреса \$00-\$3F. При доступе к регистру ввода/вывода как к ячейке ОЗУ, к его адресу необходимо добавить \$20. В приведенной выше таблице адреса регистров в памяти данных приведены в скобках.

## Регистр состояния — SREG

Регистр состояния расположен по адресу \$3F (\$5F) пространства ввода/вывода и определен следующим образом.

\$3F (\$5F) — SREG

Бит	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Чт./зап.	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальн.знач.	0	0	0	0	0	0	0	0

Бит 7 — **I**: общее разрешение прерываний. Для разрешения прерываний этот бит должен быть установлен в единицу. Управление отдельными прерываниями производится регистрами маски прерываний — **GIMSK** и **TIMSK**. Если флаг сброшен (0), независимо от состояния **GIMSK/TIMSK** прерывания запрещены. Бит **I** очищается аппаратно после входа в прерывание и восстанавливается командой **RETI**, для разрешения обработки последующих прерываний.

Бит 6 — **T**: хранение копируемого бита. Команды копирования битов **BLD (Bit Load)** и **BST (Bit Store)** используют этот бит как источник и приемник обрабатываемого бита. Бит из регистра регистрового файла может быть скопирован в **T** командой **BST**, бит **T** может быть скопирован в бит регистрового файла командой **BLD**.

Бит 5 — **N**: флаг половинного переноса. Этот флаг индицирует перенос из младшей половины байта при некоторых арифметических операциях. Более подробно об этом можно прочитать в описании системы команд.

Бит 4 — **S**: бит знака,  $S = N \text{ XOR } V$ . Бит **S** всегда равен исключающему ИЛИ между флагами **N** (отрицательный результат) и **V** (переполнение дополнения до двух). Более подробно об этом можно прочитать в описании системы команд.

Бит 3 — **V**: флаг переполнения дополнения до двух. Этот флаг поддерживает арифметику с дополнением до двух. Более подробно об этом можно прочитать в описании системы команд.

Бит 2 — **N**: флаг отрицательного результата. Этот флаг индицирует отрицательный результат различных арифметических и логических операций. Более подробно об этом можно прочитать в описании системы команд.

Бит 1 — **Z**: флаг нулевого результата. Этот флаг индицирует нулевой результат различных арифметических и логических операций. Более подробно об этом можно прочитать в описании системы команд.

Бит 0 — **C**: флаг переноса. Этот флаг индицирует перенос в арифметических и логических операциях. Более подробно об этом можно прочитать в описании системы команд.

## Указатель стека SP

Этот 8-разрядный регистр с адресом \$3D (\$5D) хранит указатель стека процессора AT90S2313. 8-ми разрядов достаточно, для адресации ОЗУ в пределах \$60-\$DF.

\$3D (\$5D) — SPL

Бит	7	6	5	4	3	2	1	0
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Чт./зап.	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Указатель стека указывает на область памяти, в которой расположен стек вызова подпрограмм и прерываний. Область стека в ОЗУ должна быть задана до того, как произойдет любой вызов подпрограммы или будут разрешены прерывания. Указатель стека уменьшается на 1 при записи данных в стек командой **PUSH** и уменьшается на 2 при вызове подпрограммы командой **CALL** или обработке прерывания. Указатель стека увеличивается на 1 при выборе данных из стека командой **POP** и



увеличивается на 2 при выполнении команд возврата из подпрограммы или обработчика прерывания (**RET** или **RETI**).

## Сброс и обработка прерываний

В AT90S2313 предусмотрены 10 источников прерываний. Эти прерывания и сброс имеют различные векторы в области памяти программ. Каждому из прерываний присвоен отдельный бит разрешающий данное прерывание при установке бита в 1, если бит I регистра состояния разрешает общее обслуживание прерываний.

Самые младшие адреса памяти программ определены как векторы сброса и прерываний. Полный список векторов прерываний приведен в таблице П4.2. Этот список определяет и приоритет различных прерываний. Меньшие адреса соответствуют более высокому уровню приоритета. Самый высокий уровень у сброса, следующий приоритет у INT0 — внешнего запроса прерывания 0 и т.д.

**Таблица П4.2.** Сброс и векторы прерываний

Номер вектора	Адрес	Источник	Описание прерывания
1	\$000	RESET	Вывод сброса и сброс от сторожевого таймера
2	\$001	INT0	Внешнее прерывание 0
3	\$002	INT1	Внешнее прерывание 1
4	\$003	TIMER1 CAPT1	Захват таймера/счетчика 1
5	\$004	TIMER1 COMP1	Совпадение таймера/счетчика 1
6	\$005	TIMER1 OVF1	Переполнение таймера/счетчика 1
7	\$006	TIMER0 OVF0	Переполнение таймера/счетчика 0
8	\$007	UART RX	Последоват.порт прием закончен
9	\$008	UART UDRE	Посл.порт регистр данных пуст
10	\$009	UART TX	Посл.порт передача закончена
11	\$00A	ANA_COMP	Аналоговый компаратор

Ниже приведена типичная программа обработки сброса и векторов прерываний.

Address	Labels	Code	Comments
\$000	rjmp	RESET	Обработка сброса
\$001	rjmp	EXT_INT0	Обработка IRQ0
\$002	rjmp	EXT_INT1	Обработка IRQ1
\$003	rjmp	TIM_CAPT1	Обработка захвата таймера 1
\$004	rjmp	TIM_COMP1	Обработка совпадения таймера 1
\$005	rjmp	TIM_OVF1	Обработка переполнения таймера 1
\$006	rjmp	TIM_OVF0	Обработка переполнения таймера 0
\$007	rjmp	UART_RXC	Обработка приема байта
\$008	rjmp	UART_DRE	Обработка освобождения UDR
\$009	rjmp	UART_TXC	Обработка передачи байта
\$00a	rjmp	ANA_COMP	Обработка аналогового компаратора
\$00b	MAIN:	<instr> xxx	Начало основной программы

## Источники сброса

AT90S2313 имеет три источника сброса.

- Сброс по включению питания. Процессор сбрасывается при подаче питания на выводы VCC и GND.

- Внешний сброс. Процессор сбрасывается при подаче низкого уровня на вывод RESET на время более двух периодов тактовой частоты.
- Сброс от сторожевого таймера. Процессор сбрасывается по окончании времени обработки сторожевого таймера, если разрешена его работа.

Во время сброса все регистры ввода/вывода устанавливаются в начальные значения, программа начинает выполняться с адреса \$000, по этому адресу должна быть записана команда RJMP — относительный переход на программу обработки сброса. Если в программе не разрешаются прерывания и векторы прерываний не используются, в первых адресах памяти может быть записана программа.

### Сброс по включению питания

Цепь сброса по включению питания обеспечивает запрет включения процессора до тех пор, пока напряжение питания не достигнет безопасного уровня. После того, как напряжение питания достигнет уровня включения, процессор не включается до тех пор, пока встроенный таймер не обработает несколько рабочих периодов сторожевого таймера. Общее время сброса состоит из суммы времени **Tpor** и времени **Tout** (см. таблицу ниже)

Таблица П4.3. Характеристики сброса ( $V_{CC}=5.0V$ )

Параметр		Min	Typ	Max	Единицы
<b>Vpor</b>	Напряжение срабатывания сброса по включению питания	1,8	2	2,2	В
<b>Vrst</b>	Напряжение срабатывания сброса по выводу RESET		VCC/2		В
<b>Tpor</b>	Сигнал сброса по включен. питания	2	3	4	мс
<b>Ttout</b>	Задержка на обработку сброса FSTRT не запрограммирован	11	16	21	мс
<b>Ttout</b>	Задержка на обработку сброса FSTRT запрограммирован	1,0	1,1	1,2	мс

Если к процессору подключен керамический резонатор или другой, обеспечивающий быстрое включение, для уменьшения времени сброса можно запрограммировать бит-перемычку **FSTRT**.

Поскольку к выводу **RESET** подключен подтягивающий резистор, этот вывод может оставаться неподключенным, если не требуется внешний сброс. Подключение вывода **RESET** к напряжению питания дает тот же эффект. Время включения после подачи питания может быть увеличено удержанием вывода сброса на низком уровне.

### Внешний сброс

Внешний сброс обрабатывается по низкому уровню на выводе **RESET**. Вывод должен удерживаться в низком состоянии, по крайней мере, два периода тактовой частоты. После достижения напряжения **Vrst** запускается таймер задержки, через промежуток времени **Tout** процессор запускается.

## Сброс по сторожевому таймеру

После отработки цикла сторожевого таймера, он вырабатывает короткий импульс сброса. По спадающему фронту этого импульса запускается внутренний счетчик, отсчитывающий время **Tout**.

## Обработка прерываний

AT90S2313 имеет два регистра маски прерываний **GIMSK** — общий регистр маски прерываний и **TIMSK** — регистр маски прерываний от таймера/счетчика.

Когда возникает прерывание бит глобального разрешения прерываний, **I** сбрасывается (ноль) и все прерывания запрещаются. Программа пользователя может установить этот бит для разрешения прерываний. Флаг разрешения прерываний **I** устанавливается в 1 при выполнении команды выхода из прерывания — **RETI**.

Для прерываний включаемых статическими событиями (напр. совпадение значения счетчика/таймера 1 с регистром совпадения) флаг прерывания взводится при возникновении события. Если флаг прерывания очищен и присутствует условие возникновения прерывания, флаг не будет установлен, пока не произойдет следующее событие.

Когда программный счетчик устанавливается на текущий вектор прерывания для его обработки, соответствующий флаг, сгенерированный прерыванием, аппаратно сбрасывается. Некоторые флаги прерывания могут быть сброшены записью логической единицы в бит соответствующий флагу.

## Общий регистр маски прерываний GIMSK

\$3B(\$5B) — GIMSK

Бит	7	6	5	4	3	2	1	0
	INT0	INT0	-	-	-	-	-	-
Чт./зап.(R/W)	R	R/W	R	R	R	R	R	R
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **INT1**: Запрос внешнего прерывания 1 разрешен. Когда этот бит установлен, а также установлен бит **I** регистра состояния, разрешается прерывание от внешнего вывода. Биты управления запуском прерывания (**ISC11** и **ISC10**) в регистре управления микроконтроллером (**MCUCR**) определяют по какому событию отрабатывается прерывание: по спадающему или нарастающему фронту или же по уровню. Активность на выводе приводит к возникновению прерываний, даже если вывод сконфигурирован как выход. При возникновении прерывания выполняется программа, начинающаяся с адреса \$002 в памяти программ (см. также «Внешние прерывания»).

Бит 6 — **INT0**: запрос внешнего прерывания 0 разрешен. Когда этот бит установлен, а также установлен бит **I** регистра состояния, разрешается прерывание от внешнего вывода. Биты управления запуском прерывания (**ISC01** и **ISC00**) в регистре управления микроконтроллером (**MCUCR**) определяют по какому событию отрабатывается прерывание: по спадающему или нарастающему фронту или же по уровню. Активность на выводе приводит к возникновению прерываний, даже если вывод сконфигурирован как выход. При возникновении прерывания выполняется

программа, начинающаяся с адреса \$001 в памяти программ (см. также «Внешние прерывания»).

Биты 5–0 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

### Общий регистр флагов прерываний

\$3B(\$5B) — GIFR

Бит	7	6	5	4	3	2	1	0
	INTF1	INTF0	-	-	-	-	-	-
Чт./зап.(R/W)	R	R/W	R	R	R	R	R	R
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **INTF1**: флаг внешнего прерывания 1. При возникновении на выводе **INT1** события вызывающего прерывание, **INTF1** устанавливается в «1». Если установлены бит 1 регистра **SREG** и бит **INT1** в **GIMSK**, происходит переход на вектор прерывания по адресу \$002. Флаг очищается после выполнения обработчика прерывания. Кроме того, флаг можно очистить, записав в него логическую единицу.

Бит 6 — **INTF0**: флаг внешнего прерывания 0. При возникновении на выводе **INT0** события вызывающего прерывание, **INTF0** устанавливается в «1». Если установлены бит 1 регистра **SREG** и бит **INT0** в **GIMSK**, происходит переход на вектор прерывания по адресу \$001. Флаг очищается после выполнения обработчика прерывания. Кроме того, флаг можно очистить, записав в него логическую единицу.

Биты 5–0 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

### Регистр маски прерываний от таймера/счетчика — TIMSK

\$39(\$59) — TIMSK

Бит	7	6	5	4	3	2	1	0
	TOIE1	OCIE1A	-	-	TICIE1	-	TOIE0	-
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **TOIE1**: разрешение прерывания по переполнению таймера/счетчика 1. Если установлен этот бит и бит разрешения прерываний в регистре состояния, разрешены прерывания по переполнению таймера/счетчика 1. Соответствующее прерывание (вектор \$005) выполняется при переполнении таймера/счетчика 1. В регистре флагов таймеров/счетчиков (**TIFR**) устанавливается флаг переполнения. Если таймер/счетчик 1 работает в режиме ШИМ, флаг переполнения устанавливается при изменении направления счета, при значении \$0000.

Бит 6 — **OCIE1A**: разрешение прерывания по совпадению таймера/счетчика 1. Если установлены бит **OCIE1A** и бит разрешения прерывания в регистре состояния, разрешены прерывания по совпадению таймера/счетчика 1. Прерывание (вектор \$004) выполняется при равенстве таймера/счетчика 1 и регистра совпадения. Во флаговом регистре **TIFR** устанавливается («1») флаг совпадения.

Биты 5,4 — зарезервированы; в AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 3 — **TICIE1**: разрешение прерывания по входу захвата. Если установлены бит **TICIE1** и бит разрешения прерывания в регистре состояния, разрешены преры-

вания по входу захвата. Соответствующее прерывание (вектор \$003) выполняется по сигналу захвата на выводе 11 (PD6/ICP). Во флаговом регистре **TIFR** устанавливается («1») флаг захвата.

Бит 2 — зарезервирован; в AT90S2313 этот бит зарезервирован и всегда читается как 0.

Бит 1—**TOIE0**: разрешение прерывания по переполнению таймера/счетчика 0. Если этот бит установлен в 1, и бит **I** в регистре состояния установлен в 1, разрешены прерывания по переполнению таймера/счетчика 0. При возникновении переполнения выполняется соответствующий вектор прерывания (\$006). Флаг переполнения (**TOV0**) во флаговом регистре прерываний (**TIFR**) таймеров/счетчиков устанавливается в 1.

Бит 0 — зарезервирован; в AT90S2313 этот бит зарезервирован и всегда читается как 0.

### Регистр флагов прерываний от таймеров /счетчиков — **TIFR**

\$38(\$58) — **TIFR**

Бит	7	6	5	4	3	2	1	0
	TOV1	OCF1A	-	-	ICF1	TOV0	-	-
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **TOV1**: флаг переполнения таймера/счетчика 1. Флаг **TOV1** устанавливается («1») при возникновении переполнения таймера/счетчика 1. Флаг **TOV1** сбрасывается аппаратно при выполнении соответствующего вектора обработки прерывания. Кроме того, флаг можно сбросить, записав в него логическую единицу. Если установлены бит **I** в **SREG** и бит **TOIE1** в **TIMSK**, при установке бита **TOV1** выполняется прерывание по переполнению таймера/счетчика 1. В режиме ШИМ этот бит устанавливается, когда таймер/счетчик 1 изменяет направление счета при значении \$0000.

Бит 6 — **OCF1A**: флаг выхода совпадения 1А. Флаг устанавливается в «1» если происходит совпадение значения таймера/счетчика 1 и данных в регистре **OCR1A**. Флаг очищается аппаратно при выполнении соответствующего вектора прерывания. Кроме того, флаг можно сбросить записав в него логическую единицу. Если установлены бит **I** в **SREG** и бит **OCIE1A** в **TIMSK**, при установке бита **OCF1A** выполняется прерывание.

Биты 5,4 — зарезервированы; в AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 3 — **ICF1**: флаг входа захвата 1: бит устанавливается («1») при возникновении события захвата по входу, он индицирует, что значение таймера/счетчика 1 скопировано в регистр захвата по входу **ICR1**. **ICF1** очищается при выполнении соответствующего вектора обработки прерывания. Кроме того, флаг можно очистить, записав в него логическую единицу.

Бит 2 — зарезервирован; в AT90S2313 этот бит зарезервирован и всегда читается как 0.

Бит 1 — **TOV0**: флаг переполнения таймера счетчика 1; флаг **TOV0** устанавливается («1») при переполнении таймера/счетчика 0. Флаг сбрасывается аппаратно при выполнении соответствующего вектора прерывания. Кроме того, флаг можно очи-

стить, записав в него логическую единицу. Если установлены бит **I** в **SREG** и бит **TOIE0** в **TIMSK**, при установке бита **TOV0** выполняется прерывание по переполнению таймера/счетчика 0.

Бит 0 — зарезервирован; в AT90S2313 этот бит зарезервирован и всегда читается как 0.

## Внешние прерывания

Внешние прерывания управляются выводами **INT0** и **INT1**. Заметим, что прерывания обрабатываются, даже когда выводы сконфигурированы как выходы. Это позволяет генерировать программные прерывания. Внешние прерывания могут возникать по спадающему или нарастающему фронту, а также по низкому уровню. Это устанавливается в регистре управления процессором **MCUCR**. Если внешние прерывания разрешены и сконфигурированы на обработку по уровню, прерывание будет вырабатываться до тех пор, пока вывод удерживается в низком состоянии.

Управление работой внешних прерываний рассмотрено при описании регистра управления процессором **MCUCR**.

## Время реакции на прерывание

Минимальное время реакции на любое из предусмотренных в процессоре прерываний — 4 периода тактовой частоты. После четырех циклов вызывается программный вектор обрабатывающий данное прерывание. За эти 4 цикла программный счетчик записывается в стек, указатель стека уменьшается на 2. Программный вектор представляет собой относительный переход на подпрограмму обслуживания прерывания и этот переход занимает 2 периода тактовой частоты. Если прерывание происходит во время выполнения команды длящейся несколько циклов, перед вызовом прерывания завершается выполнение этой команды.

Выход из программы обслуживания прерывания занимает 4 периода тактовой частоты. За эти 4 периода из стека восстанавливается программный счетчик. После выхода из прерывания процессор всегда выполняет еще одну команду, прежде чем обслужить любое отложенное прерывание.

Заметим, что регистр состояния **SREG** аппаратно не обрабатывается процессором, как при вызове подпрограмм, так и при обслуживании прерываний. Если программа требует сохранения **SREG**, оно должно производиться программой пользователя.

## Регистр управления микроконтроллером — MCUCR

Этот регистр содержит биты общего управления микроконтроллером.  
\$35(\$55) — MCUCR

Бит	7	6	5	4	3	2	1	0
	-	-	SE	SM	ISC11	ISC10	ISC01	ISC00
Чт./зап.(R/W)	R	R	R/W	R/W	R	R	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Биты 7,6 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 5 — **SE**: разрешение режима **Sleep**. Этот бит должен быть установлен в 1, чтобы при выполнении команды **SLEEP** процессор переходил в режим пониженно-

го энергопотребления. Этот бит должен быть установлен в 1 до исполнения команды **SLEEP**.

Бит 4 — **SM**: режим **Sleep**. Этот бит выбирает один из двух режимов пониженного энергопотребления. Если бит сброшен (0), в качестве режима **Sleep** выбирается холостой режим (**Idle mode**). Если бит установлен, тогда выбирается экономичный режим (**Power down**). Особенности каждого из режимов будут рассмотрены ниже.

Биты 3,2 — **ISC11, ISC10**: биты управления срабатыванием прерывания 1. Внешнее прерывание активируется выводом **INT1**, если установлен флаг I регистра состояния **SREG** и установлена соответствующая маска в регистре **GIMSK**. Срабатывание по уровню и фронтам задается следующим образом.

**Таблица П4.4.** Управление срабатыванием прерывания 1

ISC11	ISC10	Описание
0	0	Запрос на прерыв. генерируется по низк. уровню на входе INT1
0	1	Зарезервировано
1	0	Запрос на прерывание по спадающему фронту на входе INT1
1	1	Запрос на прерывание по нарастающ. фронту на входе INT1

Примечание. При изменении битов ISC11/ISC10 прерывание INT1 должно быть запрещено очисткой соответствующего бита в регистре GIMSK. Иначе прерывание может возникнуть во время изменения битов.

**Таблица П4.5.** Управление срабатыванием прерывания 0

ISC01	ISC00	Описание
0	0	Запрос на прерыв. генерируется по низк. уровню на входе INT0
0	1	Зарезервировано
1	0	Запрос на прерывание по спадающему фронту на входе INT0
1	1	Запрос на прерывание по нарастающ. фронту на входе INT0

Примечание. При изменении битов ISC01 и ISC00, прерывания по входу INT0 должны быть запрещены сбросом бита разрешения прерывания в регистре GIMSK. Иначе прерывание может произойти при изменении значения битов.

Биты 1,0 — **ISC01, ISC00**: биты управления срабатыванием прерывания 0. Внешнее прерывание активируется выводом **INT0** если установлен флаг I регистра состояния **SREG** и установлена соответствующая маска в регистре **GIMSK**. В табл. П4.5 приведена установка битов для задания срабатывания по уровню и фронтам.

## Режимы пониженного энергопотребления

Для запуска режима пониженного энергопотребления должен быть установлен (1) бит **SE** регистра **MCUCR**, и должна быть исполнена команда **SLEEP**. Если во время нахождения в режиме пониженного потребления происходит одно из разрешенных прерываний, процессор начинает работать, исполняет подпрограмму обработки прерывания и продолжает выполнение программы с команды следующей за **SLEEP**. Содержимое регистрового файла и памяти ввода/вывода не изменяется. Если в режиме пониженного потребления происходит сброс, процессор начинает выполнение программы с вектора сброса.

Если для вывода из экономичного режима используется прерывание по уровню, низкий уровень должен удерживаться на время достаточное для запуска генератора тактовых импульсов — 16 мс. Иначе флаг прерывания может вернуться в 0 до того как процессор начнет работу.

## Режим холостого хода

Когда бит **SM** сброшен (0), команда **SLEEP** переводит процессор в режим холостого хода (**Idle mode**). ЦПУ останавливается, но таймеры/счетчики, сторожевой таймер и система прерываний продолжают работать. Это позволяет процессору возобновлять работу как от внешних прерываний, так и по переполнению таймеров/счетчиков или по сбросу от сторожевого таймера. Если прерывание от аналогового компаратора не требуется, аналоговый компаратор может быть отключен установкой бита **ACD** регистра **ACSR**. Это уменьшает потребляемую мощность в режиме холостого хода.

## Экономичный режим

Когда бит **SM** установлен (1), команда **SLEEP** переводит процессор в экономичный режим (**Power Down Mode**). В этом режиме останавливается внешний генератор тактовых импульсов. Пользователь может разрешить работу сторожевого таймера в этом режиме. Если сторожевой таймер разрешен, процессор выходит из экономичного режима после отработки периода сторожевого таймера. Если сторожевой таймер запрещен, выход из экономичного режима может произойти только по внешнему сбросу или внешнему прерыванию по уровню.

## Таймеры/счетчики

В AT90S2313 предусмотрены два таймера/счетчика общего назначения. 8-разрядный и 16-разрядный. Каждый из таймеров индивидуально подключается к одному из выходов 10-разрядного предварительного делителя частоты. Оба таймера могут использоваться как таймеры с внутренним источником импульсов или счетчики импульсов поступающих извне.

В качестве источника импульсов для таймеров можно выбрать сигнал с тактовой частотой процессора (СК), импульсы предварительного делителя (СК/8, СК/64, СК/256 или СК/1024) или импульсы с соответствующего внешнего вывода. Кроме того, таймеры могут быть остановлены, запретом прохождения импульсов на них.

### 8-разрядный таймер/счетчик 0

8-разрядный таймер/счетчик может получать импульсы тактовой частоты — СК, импульсы с предварительного делителя (СК/8, СК/64, СК/256 или СК/1024), импульсы с внешнего вывода или быть остановлен соответствующими установками регистра **TCCR0**. Флаг переполнения таймера находится в регистре **TIFR**. Биты управления таймером расположены в регистре **TCCR0**. Разрешение и запрещение прерываний от таймера управляется регистром **TIMSK**.

При работе таймера/счетчика от внешнего сигнала, внешний сигнал синхронизируется с тактовым генератором ЦПУ. Для правильной обработки внешнего сигнала, минимальное время между соседними импульсами должно превышать период тактовой частоты процессора. Сигнал внешнего источника обрабатывается по спадающему фронту тактовой частоты процессора.

8-разрядный таймер/счетчик можно использовать как счетчик с высоким разрешением, так и для точных применений с низким коэффициентом деления тактовой



частоты. Более высокие коэффициенты деления можно использовать для медленных функций или для измерения временных интервалов между редкими событиями.

### Регистр управления таймером/счетчиком — TCCR0

\$33(\$53) — TCCR0

Бит	7	6	5	4	3	2	1	0
	-	-	-	-	-	CS02	CS01	CS00
Чт./зап.(R/W)	R	R	R	R	R	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Биты 7–3 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0. Биты 2,1,0 — **CS02**, **CS01**, **CS00** — выбор тактовой частоты. Эти биты задают коэффициент деления предварительного делителя.

**Таблица П4.6.** Выбор коэффициента предварительного деления

CS02	CS01	CS00	Описание
0	0	0	Таймер/счетчик остановлен
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Внешний вывод T0, нарастающий фронт
1	1	1	Внешний вывод T0, спадающий фронт

Условие **Stop** запрещает/разрешает функционирование таймера/счетчика. В режимах деления используется частота тактового генератора. При использовании работы от внешнего источника предварительно должен быть установлен соответствующий бит регистра направления данных (0 — включает ножку на ввод).

### Таймер/счетчик 0 — TCNT0

\$32(\$52) — TCNT0

Бит	7	6	5	4	3	2	1	0
	MSB	-	-	-	-	-	-	LSB
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Таймер/счетчик реализован как нарастающий счетчик с возможностью чтения и записи. При записи таймера/счетчика, если присутствуют тактовые импульсы, таймер/счетчик продолжает счет в следующем за операцией записи тактовом цикле таймера.

### 16-разрядный таймер/счетчик 1

16-разрядный таймер/счетчик может получать импульсы тактовой частоты — СК, импульсы с предварительного делителя (СК/8, СК/64, СК/256 или СК/1024), импульсы с внешнего вывода или быть остановлен соответствующими установками регистра **TCCR1A**. Флаги состояния таймера (переполнения, совпадения и захвата) и управляющие сигналы находится в регистре **TIFR**. Разрешение и запрещение прерываний от таймера 1 управляется регистром **TIMSK**.

При работе таймера/счетчика 1 от внешнего сигнала, внешний сигнал синхронизируется с тактовым генератором ЦПУ. Для правильной обработки внешнего сигнала, минимальное время между соседними импульсами должно превышать период тактовой частоты процессора. Сигнал внешнего источника обрабатывается по спадающему фронту тактовой частоты процессора.

16-разрядный таймер/счетчик 1 можно использовать как счетчик с высоким разрешением, так и для точных применений с низким коэффициентом деления тактовой частоты. Более высокие коэффициенты деления можно использовать для медленных функций или измерения временных интервалов между редкими событиями.

Таймер/счетчик 1 поддерживает функцию совпадения используя регистр совпадения **OCR1A** в качестве источника для сравнения с содержимым счетчика. Функция совпадения поддерживает очистку счетчика и переключение выхода по совпадению.

Таймер/счетчик 1 можно использовать как 8-, 9- или 10-разрядный широтно-импульсный модулятор. В этом режиме счетчик и регистр **OCR1** работают как защищенный от дребезга независимый ШИМ с отцентрованными импульсами. Подробно эта функция будет описана ниже.

Функция захвата по входу предусматривает захват содержимого таймера/счетчика 1 в регистр захвата **ICR1** и управляется внешним сигналом на входе захвата — **ICP**. Работа режима захвата определяется управляющим регистром **TCCR1**.

При работе захвата по входу, может быть включена схема подавления шума, при этом сигнал захвата возникает только в том случае, если событие, управляющее захватом, наблюдается на протяжении 4-х машинных циклов.

#### Регистр управления таймером/счетчиком — **TCCR1A**

##### **\$2F(\$4F) — TCCR1A**

Бит	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	-	-	-	-	PWM11	PWM10
Чт./зап.(R/W)	R/W	R/W	R	R	R	R	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Биты 7,6 — **COM1A1**, **COM1A0**: режим выхода совпадения, биты 1 и 0. Эти управляющие биты задают отклик вывода **OC1** процессора на совпадение регистра сравнения и таймера/счетчика 1. Поскольку это альтернативная функция порта, соответствующий бит направления должен устанавливать вывод на выход. Конфигурация управляющих бит показана в табл. П4.7.

**Таблица П4.7.** Установка режима совпадения

<b>COM1A1</b>	<b>COM1A0</b>	<b>Описание</b>
0	0	Таймер/счетчик 1 отключен от вывода OC1
0	1	Переключение выхода OC1
1	0	Сброс (0) вывода OC1
1	1	Установка (1) вывода OC1

В режиме ШИМ эти биты имеют другие функции, которые указаны в табл. П4.11.

При изменении битов **COM1A1** и **COM1A0** прерывание по совпадению должно быть запрещено, очисткой соответствующего бита в регистре **TIMSK**. Иначе, прерывание может произойти во время изменения битов.

Биты 5–2 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Биты 1,0 — **PWM11**, **PWM10**: биты установки ШИМ. Эти биты устанавливают режим работы таймера/счетчика 1 в качестве ШИМ (см. табл. П4.8). Подробнее этот режим будет рассмотрен ниже.

**Таблица П4.8.** Установка режима работы ШИМ

PWM11	PWM10	Описание	PWM11	PWM10	Описание
0	0	Работа ШИМ запрещена	1	0	9-разрядный ШИМ
0	1	8-разрядный ШИМ	1	1	10-разрядный ШИМ

### Регистр В управления таймером/счетчиком 1 — TCCR1B

\$2E(\$4E) — TCCR1B

Бит	7	6	5	4	3	2	1	0
	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10
Чт./зап.(R/W)	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **ICNC1**: подавляет входные шумы на входе захвата. Если этот бит сброшен (0), подавление входного шума на входе захвата запрещено. При этом захват срабатывает по первому заданному (нарастающему или спадающему) фронту сигнала на выводе **ICP**. При установке бита обрабатываются четыре последовательные выборки сигнала на выводе **ICP**. Для срабатывания захвата все выборки должны соответствовать уровню заданному битом **ICES1**. Частота выборок равна тактовой частоте процессора.

Бит 6 — **ICES1**: выбор фронта сигнала захвата: Если бит **ICES1** сброшен (0) содержимое таймера/счетчика 1 переписывается в регистр захвата по спадающему фронту сигнала на выводе **ICP**. Если бит установлен по нарастающему фронту сигнала.

Биты 5,4 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 3 — **CTC1**: очистка таймера счетчика 1 по совпадению. Если бит установлен (1), таймер/счетчик 1 устанавливается в \$0000 в такте, следующем за событием совпадения. Если бит сброшен, таймер/счетчик 1 продолжает считать пока не будет остановлен, сброшен, произойдет его переполнение или изменение направления счета. В режиме ШИМ этот бит не работает.

Биты 2,1,0 — **CS12**, **CS11**, **CS10**: выбор тактирования: Эти биты определяют источник счетных импульсов для таймера/счетчика 1.

**Таблица П4.9.** Выбор источника счетных импульсов

CS12	CS11	CS10	Описание
0	0	0	таймер/счетчик 1 остановлен
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	спадающий фронт на выводе T1
1	1	1	нарастающий фронт на выводе T1

**Таймер/счетчик 1 — TCNT1H и TCNT1L**

\$2D(\$4D) — TCNT1H; \$2C(\$4C) — TCNT1L

Нач.знач.	0	0	0	0	0	0	0	0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	15	14	13	12	11	10	9	8
	MSB	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LSB
Бит	7	6	5	4	3	2	1	0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Это 16-разрядный регистр, содержащий текущее значение таймера/счетчика 1. Чтобы чтение и запись двух байт счетчика происходило синхронно, для работы с ним используется временный регистр (**TEMP**).

Запись в таймер/счетчик 1: при записи старшего байта в **TCNT1H**, записываемые данные помещаются в регистр **TEMP**. Затем, при записи младшего байта, он вместе с данными из **TEMP** переписывается в таймер/счетчик 1. Таким образом, при записи 16-разрядного значения первым должен записываться байт в **TCNT1H**.

Чтение таймера/счетчика 1: При чтении младшего байта из **TCNT1L**, он посылается в процессор, а данные из **TCNT1H** переписываются в регистр **TEMP**, т. е. одновременно читаются все 16-разрядов. При последующем чтении регистра **TCNT1H**, данные берутся из регистра **TEMP**.

Таймер/счетчик 1 организован как суммирующий счетчик (в режиме ШИМ — суммирующий/вычитающий) с возможностью чтения и записи. Если выбран источник тактовых импульсов для таймера/счетчика 1, после записи в него нового значения, он продолжает счет в следующем после записи периоде тактовой частоты.

**Регистр совпадения А таймера / счетчика 1 — OCR1AH и OCR1AL**

\$2B(\$4B) — OCR1AH; \$2A(\$4A) — OCR1AL

Нач.знач.	0	0	0	0	0	0	0	0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Бит	15	14	13	12	11	10	9	8
	MSB	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LSB
Бит	7	6	5	4	3	2	1	0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Регистр совпадения — 16-разрядный регистр, доступный для чтения и записи.

В этом регистре хранятся данные, которые непрерывно сравниваются с текущим значением таймера/счетчика 1. Действие по совпадению задается регистрами управления таймером/счетчиком 1 и регистром состояния.

Поскольку регистр **OCR1A** является 16-разрядным, при записи нового значения в регистр, для того чтобы оба байта регистра записывались одновременно, используется временный регистр. При записи старшего байта, данные помещаются во временный регистр, который переписывается в **OCR1AH** при записи младшего байта в **OCR1AL**. Таким образом, для записи в регистр первым должен записываться старший байт.

**Регистр захвата таймера/счетчика 1 — ICR1H и ICR1L**

\$25(\$45) — ICR1H; \$24(\$44) — ICR1L

Нач.знач.	0	0	0	0	0	0	0	0
Чт./зап.(R/W)	R	R	R	R	R	R	R	R
Бит	15	14	13	12	11	10	9	8
	MSB	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LSB
Бит	7	6	5	4	3	2	1	0
Чт./зап.(R/W)	R	R	R	R	R	R	R	R
Нач.знач.	0	0	0	0	0	0	0	0

Регистр захвата 16-разрядный регистр доступный только для чтения.

По нарастающему или спадающему фронту (в соответствии с выбором фронта импульса захвата **ICES1**) сигнала на выводе **ICP** текущее значение таймера/счетчика 1 переписывается в регистр захвата ICR1. В это же время устанавливается флаг захвата **ICF1**.

Поскольку регистр захвата является 16-разрядным, для чтения его значения, чтобы оба байта прочитались одновременно, используется временный регистр. При чтении младшего байта **ICR1L**, он посылается в ЦПУ, а старший байт регистра **ICR1H** переписывается во временный регистр. При чтении старшего байта, он принимается из временного регистра. Таким образом, для чтения 16-разрядного регистра первым должен читаться младший байт.

**Таймер/счетчик в режиме ШИМ**

При выборе режима широтно-импульсной модуляции (ШИМ), таймер/счетчик 1 и регистр совпадения OCR1A формируют 8, 9 или 10-разрядный непрерывный свободный от «дрожания» и правильный по фазе сигнал, выводимый на ножку PB3(OC1). Таймер/счетчик 1 работает как реверсивный счетчик считающий от 0 до конечного значения (см. табл. П4.10). При достижении конечного значения счетчик начинает считать в обратную сторону до нуля, после чего рабочий цикл повторяется. Когда значение счетчика совпадает с 8, 9 или 10-ю младшими битами регистра **OCR1A**, вывод **PD1(OC1)** устанавливается или сбрасывается в соответствии с установками бит **COM1A1** и **COM1A0** в регистре **TCCR1** (см. табл. П4.11).

**Таблица П4.10.** Конечное значение таймера и частота ШИМ

Разрешение ШИМ	Конечное значение таймера	Частота ШИМ
8 бит	\$00FF (255)	Ftc1/510
9 бит	\$01FF (511)	Ftc1/1022
10 бит	\$03FF (1023)	Ftc1/2046

**Таблица П4.11.** Установка режима совпадения при работе ШИМ

COM1A1	COM1A0	Влияние на вывод OC1
0	0	Не подключен
0	1	Не подключен
1	0	Очищается при совпадении, для возрастания счетчика и сбрасывается для уменьшения (не инвертирующий ШИМ)
1	1	Очищается при совпадении, для уменьшения счетчика и сбрасывается для возрастания (инвертирующий ШИМ)

В режиме ШИМ, при записи в регистр **OCR1A**, 10 младших бит передаются во временный регистр и переписываются только при достижении таймером/счетчиком конечного значения. При этом устраняется появление несимметричных импульсов (дрожания), которые неизбежны при асинхронной записи **OCR1A**.

Если **OCR1A** содержит значение \$0000 или конечное значение (TOP), вывод **OC1** остается в том состоянии, которое определяется установками **COM1A1** и **COM1A0**. Это показано в табл. П4.12.

**Таблица П4.12.** Выход ШИМ для **OCR=\$0000** или **TOP**

<b>COM1A1</b>	<b>COM1A0</b>	<b>OCR1A</b>	<b>Вывод OC1</b>
1	0	\$0000	Низкий
1	0	TOP	Высокий
1	1	\$0000	Высокий
1	1	TOP	Низкий

В режиме ШИМ флаг переполнения таймера 1 (**TOV1**) устанавливается, когда счетчик изменяет направление счета в точке \$0000. Прерывание по переполнению таймера 1 работает как при нормальном режиме работы таймера/счетчика, т.е. оно выполняется, если установлен флаг **TOV1** и разрешены соответствующие прерывания. То же самое касается флага совпадения и прерывания по совпадению.

### Сторожевой таймер

Сторожевой таймер работает от отдельного встроенного генератора работающего на частоте 1 МГц (это типовое значение частоты для питания 5В). Управляя предварительным делителем сторожевого таймера, можно задавать интервал сброса таймера от 16 до 2048 мс. Команда **WDR** сбрасывает сторожевой таймер. Для работы сторожевого таймера можно выбрать одно из 8-ми значений частоты, что позволяет в широких пределах изменять время между исполнением команды **WDR** и сбросом процессора. При отработке периода работы сторожевого таймера, если не поступила команда **WDR**, AT90S2313 сбрасывается, выполнение программы продолжается с вектора сброса.

Для предотвращения нежелательного отключения сторожевого таймера, для его запрещения должна выполняться определенная последовательность, которая описана при рассмотрении регистра **WDTCR**.

### Регистр управления сторожевым таймером — **WDTCR**

\$21(\$41) — **WDTCR**

<b>Бит</b>	7	6	5	4	3	2	1	0
	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0
<b>Чт./зап. (R/W)</b>	R	R	R	R	R/W	R/W	R/W	R/W
<b>Нач.знач.</b>	0	0	0	0	0	0	0	0

Биты 7–5 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 4 — **WDTOE** — разрешение выключения сторожевого таймера. При очистке бита **WDE** этот бит должен быть установлен (1). Иначе, работа сторожевого таймера не прекращается. Через четыре такта после установки этого бита, он аппаратно сбрасывается.

Бит 3 — **WDE** — разрешение сторожевого таймера. Если бит установлен (1), работа сторожевого таймера разрешена, если бит сброшен — запрещена. Сброс бита производится только в том случае, если бит **WDTOE** установлен в 1. Для запрещения включенного сторожевого таймера должна выполняться следующая процедура.

1. Одной командой записать 1 в **WDTOE** и **WDE**. Единица в **WDE** должна записываться даже в том случае если этот бит был установлен перед началом процедуры остановки таймера.
2. В течение следующих четырех тактов процессора необходимо записать в **WDE** логический 0, при этом работа сторожевого таймера запрещается.

Биты 2–0 — **WDP2–0** — биты предварительного делителя сторожевого таймера. Если работа сторожевого таймера разрешена, эти биты определяют предварительный коэффициент деления для сторожевого таймера. В табл. П4.13 приведены различные значения установок предварительного делителя и соответствующие им временные интервалы для напряжения питания  $V_{cc}=5V$ .

**Таблица П4.13.** Установки предварительного делителя сторожевого таймера

WDP2	WDP1	WDP0	Период времени	WDP2	WDP1	WDP0	Период времени
0	0	0	16 мс	1	0	0	256 мс
0	0	1	32 мс	1	0	1	512 мс
0	1	0	64 мс	1	1	0	1024 мс
0	1	1	128 мс	1	1	1	2048 мс

### Чтение и запись в энергонезависимую память

Регистры доступа к энергонезависимой памяти (**EEPROM**) расположены в пространстве ввода/вывода.

Время записи находится в диапазоне 2,5–4 мс и зависит от напряжения питания. Это самотактируемая функция, которая, однако, позволяет пользователю определить, можно ли записывать следующий байт. Если программа пользователя производит запись в энергонезависимую память, должны быть предприняты некоторые меры предосторожности. При использовании в источнике питания конденсаторов большой емкости, напряжение питания нарастает и спадает достаточно медленно. Это приводит к тому, что процессор некоторое время работает при напряжении питания ниже минимума достаточного для нормальной работы схем тактирования. При этом ЦПУ может совершать нежелательные переходы, попадая на части программы, производящие запись в EEPROM. В таких случаях, для защиты содержимого EEPROM, необходимо использовать внешние схемы формирующие сигнал сброса при уменьшении напряжения питания.

Для защиты от нежелательной записи в EEPROM необходимо следовать некоторым правилам, которые будут рассмотрены ниже, при описании управляющего регистра энергонезависимой памяти.

При записи или чтении EEPROM процессор приостанавливается на 2-х машинных цикла до начала выполнения следующей команды.

#### Регистр адреса EEPROM — **EEAR**

\$1E (\$3E) — **EEAR**

Бит	7	6	5	4	3	2	1	0
	-	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR	EEAR0
Чт./зап.(R/W)	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — зарезервирован. В AT90S2313 этот бит зарезервирован и всегда читается как 0.

Биты 6–0 — **EEAR6–0** — адрес EEPROM. Адресный регистр EEPROM задает адрес в 128-байтном пространстве EEPROM. Байты данных EEPROM адресуются линейно в диапазоне 0–127.

### Регистр данных EEPROM — EEDR

#### \$1D (\$3D) — EEDR

Бит	7	6	5	4	3	2	1	0
	MSB	-	-	-	-	-	LSB	
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Биты 7–0 — **EEDR7–0** — данные EEPROM. При записи регистр **EEDR** содержит данные, которые записываются в EEPROM по адресу в регистре **EEAR**. Для операции чтения в этот регистр читаются данные, прочитанные из EEPROM по адресу заданному в регистре **EEAR**.

### Регистр управления EEPROM — EECR

#### \$1C (\$3C) — EECR

Бит	7	6	5	4	3	2	1	0
	-	-	-	-	-	EEMWE	EEWE	EERE
Чт./зап.(R/W)	R	R	R	R	R	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Биты 7–3 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

Бит 2 — **EEMWE** — управление разрешением записи. Этот бит определяет, будут ли записаны данные при установке **EEWE**. Если бит **EEMWE** установлен, при установке **EEWE** данные записываются по выбранному адресу EEPROM. Если этот бит сброшен, установка **EEWE** не имеет эффекта. После программной установки этот бит сбрасывается аппаратно через четыре такта процессора.

Бит 1 — **EEWE** — разрешение записи в EEPROM. Сигнал **EEWE** является стробом записи в EEPROM. После установки правильного адреса и данных для записи в EEPROM необходимо установить бит **EEWE**. При записи «1» в бит **EEWE** должен быть установлен бит **EEMWE**, тогда происходит запись в EEPROM. Для записи в EEPROM должна соблюдаться следующая последовательность.

1. Ждем обнуления **EEWE**.
2. Записываем адрес в **EEAR** (не обязательно).
3. Записываем данные в **EEDR** (не обязательно).
4. Устанавливаем в 1 бит **EEMWE**.
5. Не позже чем через 4 такта после установки **EEMWE** устанавливаем **EEWE**.



После того как время записи истечет (типично 2,5 мс для  $V_{CC}=5V$  и 4мс для  $V_{CC}=2,7V$ ), бит **EEWE** очищается аппаратно. Пользователь может отслеживать этот бит и ожидать его установки в ноль, перед тем как записывать следующий байт. При установке **EEWE**, ЦПУ останавливается на два цикла перед исполнением следующей команды.

Бит 0 — **EERE** — разрешение чтения из EEPROM. Сигнал **EERE** является стробом чтения из EEPROM. После установки нужного адреса в регистре **EEAR**, необходимо установить бит **EERE**. После того как бит **EERE** будет аппаратно очищен, в регистре **EEDR**. Чтение EEPROM занимает одну команду и не требует отслеживания бита **EERE**. При установке бита **EERE**, ЦПУ останавливается на два цикла перед тем как будет выполнена следующая команда. Перед чтением пользователь должен проверять состояние бита **EEWE**, если регистры данных или адреса изменяются во время операции записи, запись в ячейку прерывается и результат операции записи становится неопределенным.

### Универсальный асинхронный приемо-передатчик

В состав AT90S2313 входит универсальный асинхронный приемо-передатчик (UART), его основные особенности:

- генерация произвольных значений скорости;
- высокая скорость при низких тактовых частотах;
- 8 или 9 бит данных;
- фильтрация шума;
- определение переполнения;
- детектирование ошибки кадра;
- определение неверного стартового бита;
- три отдельных прерывания — завершение передачи, очистка регистра передачи и завершение приема.

#### Передача данных

Передача данных инициируется записью передаваемых данных в регистр ввода/вывода данных UART — **UDR**. Данные пересылаются из **UDR** в сдвиговый регистр передатчика когда.

- Новый символ записывается в **UDR** после выдвижения стопового бита для предыдущего символа. При этом сдвиговый регистр загружается сразу.
- Новый символ записывается в **UDR** до того, как выдвинут стоповый бит для предыдущего символа. При этом сдвиговый регистр записывается сразу после выдвижения стопового бита предыдущего символа.

При этом в регистре состояния UART — **USR** устанавливается бит-признак очистки регистра данных — **UDRE**. Когда этот бит установлен, UART готов к приему следующего символа. При перезаписи **UDR** в 10(11)-разрядный сдвиговый регистр, бит 0 сдвигового регистра обнуляется (стартовый бит), а бит 9 или 10 устанавливается (стоповый бит). Если выбрано 9-битовое слово данных (установлен бит **CHR9** в регистре **UCR**), бит **TXB8** из **UCR** переписывается в 9-й бит сдвигового регистра передатчика.

После тактового импульса, следующего с частотой передачи, стартовый бит выдвигается на вывод **TXD**. Затем выдвигаются данные, начиная с младшего бита. После того как выдвинут стоповый бит, в сдвиговый регистр загружаются новые данные, если они были записаны в **UDR** во время передачи. При загрузке устанавливается бит **UDRE**. Если до выдвигания стопового бита в регистр **UDR** не поступают новые данные, **UDRE** остается установленным до последующей записи **UDR**. Если новые данные не поступили и на выводе **TXD** появляется стоповый бит, в регистре **USR** устанавливается флаг окончания передачи — **TXC**.

Установка бита **TXEN** в **UCR** разрешает работу передатчика. При очистке бита **TXEN**, вывод **PD1** можно использовать для ввода/вывода данных. Если бит **TXEN** установлен, передатчик **UART** подключен к выводу **PD1** независимо от установки бита **DDD1** в регистре **DDRD**.

### Прием данных

Логическая схема приемника обрабатывает сигнал на выводе **RXD** с частотой в 16 больше скорости передачи (для обработки одного бита принимаемой последовательности, производится 16 выборок входного сигнала). В состоянии ожидания одна выборка логического нуля интерпретируется, как спадающий фронт стартового бита, после чего запускается последовательность обнаружения стартового бита. Если в первой выборке сигнала обнаружен нулевой отсчет, приемник обрабатывает 8, 9 и 10 выборки сигнала на выводе **RXD**. Если хотя бы две из трех выборок равны логической единице, стартовый бит считается шумом и приемник ждет следующего перехода из 1 в 0.

Если обнаружен стартовый бит, начинается обработка бит данных. Решение об уровне данных также производится по 8, 9 и 10 выборкам входного сигнала, уровень входного сигнала определяется по равенству двух выборок. После того как уровень данных определен, данные вдвигаются в сдвиговый регистр приемника.

Для определения стопового бита хотя бы две из трех выборок входного сигнала должны быть равны 1. Если это условие не выполняется, в регистре **USR** устанавливается флаг ошибки кадра **FE**. Перед чтением данных из регистра **UDR** пользователь должен проверять бит **FE** для обнаружения ошибок кадра.

Независимо от принятия правильного стопового бита по окончанию приема символа принятые данные переписываются в **UDR** и устанавливается флаг **RXC** в регистре **USR**. Физически регистр **UDR** состоит из двух отдельных регистров, один используется для передачи данных, другой — для приема. При чтении **UDR** происходит доступ к регистру приемника, при записи — к регистру передатчика. При обмене 9-битовыми данными 9-й бит принятых данных записывается в бит **RXB8** регистра **UCR**.

Если при приеме символа из регистра **UDR** не был прочитан предыдущий символ, в регистре **UCR** устанавливается флаг переполнения — **OR**. Установка этого бита означает, что последний принятый байт данных не переписывается из сдвигового регистра в регистр **UDR** и будет потерян. Бит **OR** буферизован и обновляется при чтении правильных данных из **UDR**. Таким образом, пользователь всегда может проверить состояние **OR** после чтения **UDR** и обнаружить происшедшее переполнение.

При сбросе бита **RXEN** в регистре **UCR** прием данных запрещается. При этом вывод **PD0** можно использовать для ввода/вывода общего назначения. При уста-

новке RXEN, приемник подключен к выводу **PD0** независимо от состояния бита **DDD0** в регистре **DDRD**.

## Управление UART

### Регистр ввода/вывода UART

\$0C (\$2C) — UDR

Бит	7	6	5	4	3	2	1	0
	MSB	-	-	-	-	-	-	LSB
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Физически регистр **UDR** является двумя отдельными регистрами, доступ к которым происходит по одному адресу. При записи происходит запись в регистр передатчика, при чтении — читается регистр приемника.

### Регистр состояния UART — USR

\$0B (\$2B) — USR

Бит	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	OR	-	-	-
Чт./зап.(R/W)	R	R	R	R	R	R	R	R
Нач.знач.	0	0	0	0	0	0	0	0

Регистр **USR** доступен только для чтения, в нем хранится информация о состоянии UART.

Бит 7 — **RXC** — прием завершен. Этот бит устанавливается в 1 когда принятый символ переписывается из сдвигового регистра приемника в регистр **UDR**. Бит устанавливается независимо от обнаружения ошибки кадра. Если установлен бит **RXCIE** в регистре **UCR**, при установке бита выполняется прерывание по завершению приема символа. **RXC** сбрасывается при чтении **UDR**. При использовании приема данных по прерыванию, обработчик прерывания должен читать регистр **UDR** для сброса **RXC**, иначе при выходе из прерывания оно будет вызвано снова.

Бит 6 — **TXC** — передача завершена. Этот бит устанавливается в 1 если символ из сдвигового регистра передатчика (включая стоповый бит) передан, а в регистр **UDR** не были записаны новые данные. Этот флаг особенно полезен при полудуплексной связи, когда передающее устройство должно перейти в режим приема и освободить линию связи сразу по окончании передачи. Если установлен бит **TXIE** в регистре **UCR**, при установке **TXC** выполняется прерывание по окончании передачи. **TXC** сбрасывается аппаратно при выполнении соответствующего вектора прерывания. Кроме того, бит можно сбросить, записав в него 1.

Бит 5 — **UDRE** — регистр данных **UART** пуст. Этот бит устанавливается в 1 когда данные, записанные в **UDR** переписываются в регистр сдвига передатчика. Установка этого бита означает, что передатчик готов принять следующий символ для передачи. Если установлен бит **UDRIE** в регистре **UCR**, тогда при установке этого бита выполняется прерывание окончания передачи. Бит **UDRE** сбрасывается при записи регистра **UDR**. При использовании передачи управляемой прерыванием, подпрограмма обслуживания прерывания должна записывать **UDR**, чтобы сбросить

бит **UDRE**, иначе при выходе из прерывания оно будет вызвано снова. При сбросе этот бит устанавливается в 1, чтобы проиндицировать готовность передатчика.

Бит 4 — **FE** — ошибка кадра. Этот бит устанавливается при обнаружении условия ошибки кадра, т.е. если стоповый бит принятого байта равен 0. Бит **FE** сбрасывается при приеме единичного стопового бита.

Бит 3 — **OR** — переполнение. Этот бит устанавливается при обнаружении условия переполнения, т.е. когда символ из регистра **UDR** не был прочитан до того, как заполнился сдвиговый регистр приемника. Этот бит буферизован, т.е. остается установленным до тех пор, пока из регистра **UDR** не будут прочитаны правильные данные. Бит **OR** сбрасывается тогда, когда принятые данные переписываются в **UDR**.

Биты 2–0 — зарезервированы. В AT90S2313 эти биты зарезервированы и всегда читаются как 0.

## Регистр управления UART — UCR

### \$0A (\$2A) — UCR

Бит	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R	W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **RXCIE** — разрешение прерывания по окончании приема. Если этот бит установлен (1), установка бита **RXC** в регистре **USR** приводит к выполнению прерывания по окончании приема (при условии, что разрешены прерывания).

Бит 6 — **TXCIE** — разрешение прерывания по окончании передачи. Если этот бит установлен, установка бита **TXC** в **USR** приводит к выполнению прерывания по окончании передачи (при условии, что прерывания разрешены).

Бит 5 — **UDRIE** — прерывание по очистке регистра данных последовательного порта. Если этот бит установлен, установка бита **UDRE** в **USR** приводит к выполнению прерывания по очистке регистра данных UART (при условии, что прерывания разрешены).

Бит 4 — **RXEN** — разрешение приемника. При установке этого бита разрешается работа приемника **UART**. Если приемник выключен, флаги **TXC**, **OR** и **FE** не устанавливаются. Если эти флаги установлены, сброс **RXEN** не очищает их.

Бит 3 — **TXEN** — разрешение передатчика. При установке этого бита разрешается работа передатчика **UART**. При запрещении работы передатчика во время передачи символа, он продолжает работать пока не будет очищен сдвиговый регистр и не будет передан символ, помещенный в **UDR**.

Бит 2 — **CHR9** — 9-битовые послыки. Если этот бит установлен, принимаемые и передаваемые символы имеют длину 9 бит. Для передачи и приема 9-го символа используются биты **RXB8** и **TXB8** соответственно. 9-й бит можно использовать как дополнительный стоповый бит или как признак четности.

Бит 1 — **RXB8** — Бит 8 принимаемых данных. Если установлен бит **CHR9**, сюда записывается 9-й бит принятых данных.

Бит 0 — **TXB8** — Бит 8 передаваемых данных. Если установлен бит **CHR9**, отсюда берется 9-й бит передаваемых данных.

## Генератор скорости передачи

Генератор скорости передачи этот делитель частоты, который генерирует скорости в соответствии с нижеприведенным выражением:

$$BAUD = F_{ck} / (16 * (UBRR + 1)),$$

где

$BAUD$  — скорость передачи (бод),

$F_{ck}$  — частота тактового генератора процессора,

$UBRR$  — содержимое регистра скорости передачи UART.

В табл. П4.14 приведены значения регистра UBRR и процентное отклонение от стандартной скорости передачи для стандартных частот кварцевых генераторов.

**Таблица П4.14.** Значения регистра UBRR и стандартные скорости передачи

Скор., бод	1,0000 МГц	Ошиб., %	Скор., бод	1,8432 МГц	Ошиб., %	Скор., бод	2,0000 МГц	Ошиб., %
2 400	25	0,2	2 400	47	0	2 400	51	0,2
4 800	12	0,2	4 800	23	0	4 800	25	0,2
9 600	6	7,5	9 600	11	0	9 600	12	0,2
14 400	3	7,8	14 400	7	0	14 400	8	3,7
19 200	2	7,8	19 200	5	0	19 200	6	7,5
28 800	1	7,8	28 800	3	0	28 800	3	7,8
57 600	0	7,8	57 600	1	0	57 600	1	7,8
115 200	0	84,3	115 200	0	0	115 200	0	7,8
Скор., бод	2,4576 МГц	Ошиб., %	Скор., бод	3,2768 МГц	Ошиб., %	Скор., бод	3,6864 МГц	Ошиб., %
2 400	63	0,0	2 400	84	0,4	2 400	95	0,0
4 800	31	0,0	4 800	42	0,8	4 800	47	0,0
9 600	15	0,0	9 600	20	1,6	9 600	23	0,0
14 400	10	3,1	14 400	13	1,6	14 400	15	0,0
19 200	7	0,0	19 200	10	3,1	19 200	11	0,0
28 800	4	6,3	28 800	6	1,6	28 800	7	0,0
57 600	2	12,5	57 600	3	12,5	57 600	3	0,0
115 200	0	25,0	115 200	1	12,5	115 200	1	0,0
Скор., бод	4,0000 МГц	Ошиб., %	Скор., бод	4,6080 МГц	Ошиб., %	Скор., бод	7,3728 МГц	Ошиб., %
2 400	103	0,2	2 400	119	0,0	2 400	191	0,0
4 800	51	0,2	4 800	59	0,0	4 800	95	0,0
9 600	25	0,2	9 600	29	0,0	9 600	47	0,0
14 400	16	2,1	14 400	19	0,0	14 400	31	0,0
19 200	12	0,2	19 200	14	0,0	19 200	23	0,0
28 800	8	3,7	28 800	9	0,0	28 800	15	0,0
57 600	3	7,8	57 600	4	0,0	57 600	7	0,0
115 200	1	7,8	115 200	2	20,0	115 200	3	0,0
Скор., бод	8,0000 МГц	Ошиб., %	Скор., бод	9,2160 МГц	Ошиб., %	Скор., бод	11,059 МГц	Ошиб., %
2 400	207	0,2	2 400	239	0,0	2 400	287	—
4 800	103	0,2	4 800	119	0,0	4 800	143	0,0
9 600	51	0,2	9 600	59	0,0	9 600	71	0,0
14 400	34	0,8	14 400	39	0,0	14 400	47	0,0
19 200	25	0,2	19 200	29	0,0	19 200	35	0,0

Таблица П4.14. Окончание

Скор., бод	8,0000 МГц	Ошиб., %	Скор., бод	9,2160 МГц	Ошиб., %	Скор., бод	11,059 МГц	Ошиб., %
28 800	16	2,1	28 800	19	0,0	28 800	23	0,0
57 600	8	3,7	57 600	9	0,0	57 600	11	0,0
115 200	3	7,8	115 200	4	0,0	115 200	5	0,0
Скор., бод	14,746 МГц	Ошиб., %	Скор., бод	16,0000 МГц	Ошиб., %	Скор., бод	18,432 МГц	Ошиб., %
2 400	383	—	2 400	416	—	2 400	479	—
4 800	191	0,0	4 800	207	0,0	4 800	239	0,0
9 600	95	0,0	9 600	103	0,0	9 600	119	0,0
14 400	63	0,0	14 400	68	0,0	14 400	79	0,0
19 200	47	0,0	19 200	51	0,0	19 200	59	0,0
28 800	31	0,0	28 800	34	0,0	28 800	39	0,0
57 600	15	0,0	57 600	16	0,0	57 600	19	0,0
115 200	7	0,0	115 200	8	0,0	115 200	9	0,0

## Регистр скорости передачи — UBRR

## \$09 (\$29) — UBRR

Бит	7	6	5	4	3	2	1	0
	MSB	-	-	-	-	-	-	LSB
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Это 8-разрядный регистр, который задает скорость передачи последовательного порта в соответствии с выражением приведенным выше.

## Аналоговый компаратор

Аналоговый компаратор сравнивает входные напряжения на положительном входе **PB0 (AIN0)** и отрицательном входе **PB1(AIN1)**. Когда напряжение на положительном входе больше напряжения на отрицательном, устанавливается бит **ACO** (Analog Comparator Output). Выход аналогового компаратора можно установить на работу с функцией захвата таймера/счетчика1. Кроме того, компаратор может вызывать свое прерывание. Пользователь может установить срабатывание прерывания по нарастающему или спадающему фронту, или по переключению.

## Регистр управления и состояния аналогового компаратора — ACSR

## \$08 (\$28) — ACSR

Бит	7	6	5	4	3	2	1	0
	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
Чт./зап.(R/W)	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

Бит 7 — **ACD** — запрещение аналогового компаратора. Когда этот бит установлен, питание от аналогового компаратора отключается. Для отключения компаратора этот бит можно установить в любое время. Обычно это свойство используется, если критично потребление процессора в холостом режиме, и восстановление работы процессора от аналогового компаратора не требуется. При изменении бита **ACD** прерывания от аналогового компаратора должны быть запрещены сбросом **ACIE** в

регистре **ACSR**. В противном случае прерывание может произойти во время изменения бита.

Бит 6 — зарезервирован. В AT90S2313 этот бит зарезервирован и всегда читается как 0.

Бит 5 — **ACO** — выход аналогового компаратора. Бит **ACO** непосредственно подключен к выходу аналогового компаратора.

Бит 4 — **ACI** — флаг прерывания от аналогового компаратора. Этот бит устанавливается когда переключение выхода компаратора совпадает с режимом прерывания установленным битами **ACIS1** и **ACIS0**. Программа обработки прерывания от аналогового компаратора выполняется если установлен бит **ACIE** (1) и установлен бит **I** в регистре состояния. **ACI** сбрасывается аппаратно при выполнении соответствующего вектора прерывания. Другой способ очистить **ACI** — записать во флаг логическую единицу.

Бит 3 — **ACIE** — разрешение прерывания от аналогового компаратора. Когда установлен этот бит и бит **I** регистра состояния, прерывания от аналогового компаратора отрабатываются. Если бит очищен (0), прерывания запрещены.

Бит 2 — **ACIC** — захват по выходу аналогового компаратора. Если этот бит установлен, функция захвата таймера/счетчика 1 управляется выходом аналогового компаратора. При этом выход компаратора подключается непосредственно к схеме обработки захвата, предоставляя удобные средства подавления шума и выбора фронта предусмотренные прерыванием захвата по входу. Когда бит очищен, схема захвата и компаратор разъединены. Чтобы компаратор мог управлять функцией захвата таймера/счетчика 1, должен быть установлен бит **TICIE1** в регистре **TIMSK**.

Биты 1,0 — **ACIS1**, **ACIS0** — выбор режима прерывания аналогового компаратора. Различные установки приведены в табл. П4.15.

**Таблица П4.15.** Установки ACIS1/ACIS0

ACIS1	ACIS0	Описание
0	0	Прерывание от компаратора по переключению выхода
0	1	Зарезервировано
1	0	Прерывание от компаратора по спадающему фронту выхода
1	1	Прерывание от компаратора по нарастающему фронту выхода

Примечание. При изменении битов ACIS1/ACIS0 прерывания от аналогового компаратора должны быть запрещены сбросом бита разрешения прерывания в регистре **ACSR**. Иначе прерывание может произойти при изменении битов.

## Порты ввода/вывода

### Порт В

Порт **В** 8-разрядный двунаправленный порт ввода/вывода.

Для обслуживания порта отведено три регистра: регистр данных **PORTB** (\$18, \$38), регистр направления данных — **DDRB** (\$17, \$37) и ножки порта В (\$16, \$36). Адрес ножек порта В предназначен только для чтения, в то время как регистр данных и регистр направления данных — для чтения/записи.

Все выводы порта имеют отдельно подключаемые подтягивающие резисторы. Выходы порта **В** могут поглощать ток до 20 мА и непосредственно управлять светодиодными индикаторами. Если выводы **PB0-PB7** используются как входы и замыкаются на землю, если включены внутренние подтягивающие резисторы, выво-

ды являются источниками тока (I<sub>il</sub>). Дополнительные функции выводов порта **B** приведены в табл. П4.16.

**Таблица П4.16.** Альтернативные функции выводов порта **B**

Вывод	Альтернативная функция
PB0	AIN0 (Положительный вход аналогового компаратора)
PB1	AIN1 (Отрицательный вход аналогового компаратора)
PB3	OC1 (Выход совпадения таймера/счетчика1)
PB5	MOSI (Вход данных для загрузки памяти)
PB6	MISO (Выход данных для чтения памяти)
PB7	SCK (Вход тактовых импульсов последовательного обмена)

При использовании альтернативных функций выводов, регистры **DDRB** и **PORTB** должны быть установлены в соответствии с описанием альтернативных функций.

#### Регистр данных порта **B** — **PORTB**

\$18 (\$38) — **PORTB**

Бит	7	6	5	4	3	2	1	0
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Чт./зап.	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

#### Регистр направления данных порта **B** — **DDRB**

\$17 (\$37) — **DDRB**

Бит	7	6	5	4	3	2	1	0
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

#### Выводы порта **B** — **PINB**

\$16 (\$36) — **PINB**

Бит	7	6	5	4	3	2	1	0
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Чт./зап.(R/W)	R	R	R	R	R	R	R	R
Нач.знач.	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

**PINB** не является регистром, по этому адресу осуществляется доступ к физическим значениям каждого из выводов порта **B**. При чтении **PORTB**, читаются данные из регистра-защелки, при чтении **PINB** читаются логические значения присутствующие на выводах порта.

#### Порт **B**, как порт ввода / вывода общего назначения

Все 8 бит порта **B** при использовании для ввода/вывода одинаковы.

Бит **DDBn** регистра **DDRB** выбирает направление передачи данных. Если бит установлен (1), вывод сконфигурирован как выход. Если бит сброшен (0) — вывод сконфигурирован как вход. Если **PORTBn** установлен и вывод сконфигурирован как вход, включается КМОП подтягивающий резистор. Для отключения резистора, **PORTBn** должен быть сброшен (0) или вывод должен быть сконфигурирован как выход.



Таблица П4.17. Влияние DDBn на выводы порта B

DDBn	PORTBn	Вх/Вых	Подт.резист	Комментарий
0	0	Вход	Нет	Третье состояние (Hi-Z)
0	1	Вход	Да	PBn источник тока I <sub>il</sub> , если извне соединен с землей
1	0	Выход	Нет	Выход установлен в 0
1	1	Выход	Нет	Выход установлен в 1

Где n — номер вывода. n = 7, 6...0.

## Альтернативные функции PORTB

**SCK** — PORTB, Bit 7 — вход тактовой частоты для загрузки/чтения памяти.

**MISO** — PORTB, Bit 6 — выход данных для чтения памяти.

**MOSI** — PORTB, Bit 5 — вход данных для загрузки памяти.

**OC1** — PORTB, Bit 3 — выход совпадения. Этот вывод может быть сконфигурирован для внешнего вывода события совпадения таймера1. Для этого бит DDB3 должен быть установлен в 1 (вывод сконфигурирован как выход).

**AIN1** — PORTB, Bit 1 — отрицательный вход аналогового компаратора. Если этот вывод сконфигурирован как вход (DDB1=0) и отключен внутренний подтягивающий резистор, этот вывод работает как отрицательный вход внутреннего аналогового компаратора.

**AIN0** — PORTB, Bit 0 — положительный вход аналогового компаратора. Если этот вывод сконфигурирован как вход (DDB0=0) и отключен внутренний подтягивающий резистор, этот вывод работает как положительный вход внутреннего аналогового компаратора.

## Порт D

Для порта **D** зарезервированы 3 ячейки памяти — регистр PORTD \$12 (\$32), регистр направления данных — DDRD \$11 (\$31) и выводы порта D — PIND \$10 (\$30). Регистры данных и направления данных могут читаться/записываться, ячейка PIND — только для чтения.

Порт **D** — 7-разрядный двунаправленный порт с встроенными подтягивающими регистрами. Выходные буферы порта могут поглощать ток до 20 мА. Если выводы используются как входы, и на них подан низкий уровень, они являются источниками тока I<sub>il</sub>, если подключены подтягивающие резисторы. Некоторые из выводов порта имеют альтернативные функции, как показано в таблице П4.18.

Таблица П4.18. Альтернативные функции порта D

Вывод порта	Альтернативная функция
PD0	RXD (вход данных UART)
PD1	TXD (выход данных UART)
PD2	INT0 (вход внешнего прерывания 0)
PD3	INT1 (вход внешнего прерывания 1)
PD4	T0 (внешний вход таймера счетчика 0)
PD5	T1 (внешний вход таймера счетчика 1)
PD6	ICP (вход захвата таймера счетчика 1)

Если выводы порта используются для обслуживания альтернативных функций, они должны быть сконфигурированы на ввод/вывод в соответствии с описанием функции.

**Регистр данных порта D — PORTD****\$12 (\$32) — PORTD**

Бит	7	6	5	4	3	2	1	0
	-	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
Чт./зап.	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

**Регистр направления данных порта B — DDRB****\$11 (\$31) — DDRB**

Бит	7	6	5	4	3	2	1	0
	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
Чт./зап.(R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

**Выводы порта B — PINB****\$10 (\$30) — PINB**

Бит	7	6	5	4	3	2	1	0
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
Чт./зап.(R/W)	R	R	R	R	R	R	R	R
Нач.знач.	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

PIND не является регистром, по этому адресу осуществляется доступ к физическим значениям каждого из выводов порта **D**. При чтении PORTD, читаются данные из регистра-защелки, при чтении PIND читаются логические значения, присутствующие на выводах порта.

**Порт D, как порт ввода/вывода общего назначения**

Бит DDDn регистра DDRD выбирает направление передачи данных (табл. П4.19). Если бит установлен (1), вывод сконфигурирован как выход. Если бит сброшен (0) — вывод сконфигурирован как вход. Если PORTDn установлен и вывод сконфигурирован как вход, включается КМОП подтягивающий резистор. Для отключения резистора, PORTDn должен быть сброшен (0) или вывод должен быть сконфигурирован как выход.

**Таблица П4.19.** Влияние DDDn на выводы порта D

DDn	PORTDn	Вх/Вых	Подт. резист	Комментарий
0	0	Вход	Нет	Третье состояние (Hi-Z)
0	1	Вход	Да	PDn источник тока I <sub>il</sub> , если извне соединен с землей
1	0	Выход	Нет	Выход установлен в 0
1	1	Выход	Нет	Выход установлен в 1

Где n — номер вывода. n = 6...0.

**Альтернативные функции порта D**

**ICP** — порт **D**, бит 6 — вход захвата таймера/счетчика 1.

**T1** — порт **D**, бит 5 — тактовый вход таймера/счетчика 1.

**T0** — порт **D**, бит 4 — тактовый вход таймера/счетчика 0.

Подробную информацию смотрите в описании таймеров.

**INT1** — порт **D**, бит 3 — вход внешних прерываний 1.

**INT0** — порт **D**, бит 2 — вход внешних прерываний 0.

Подробнее смотрите в описании прерываний.

**TXD** — порт **D**, бит 1 — выход передатчика UART. Если разрешена работа передатчика UART, независимо от состояния **DDRD1** этот вывод сконфигурирован как выход.

**RXD** — порт **D**, бит 0 — выход приемника UART. Если разрешена работа приемника UART, независимо от состояния **DDRD0** этот вывод сконфигурирован как выход. Когда UART использует вывод для приема данных, единица в **PORTD0** подключает встроенный подтягивающий резистор.

## Программирование памяти

### Программирование битов блокировки памяти

Микроконтроллер AT90S2313 имеет два бита блокировки, которые могут быть оставлены незапрограммированными (1) или программироваться (0), при этом достигаются свойства приведенные в табл. П4.20.

**Таблица П4.20.** Режимы защиты и биты блокировки

Биты блокировки			Тип защиты
Режим	LB1	LB2	
1	1	1	Защита не установлена
2	0	1	Дальнейшее программирование флэш-памяти и EEPROM запрещено
3	0	0	Как режим 2, но запрещено и чтение

ПРИМЕЧАНИЕ: биты блокировки стираются только при полном стирании памяти.

### Биты конфигурации (Fuse bits)

В AT90S2313 предусмотрено два бита конфигурации — **SPIEN** и **FSTRT**. Когда бит **SPIEN** (0) запрограммирован, разрешен режим последовательного программирования. По умолчанию бит запрограммирован (0).

Когда запрограммирован бит **FSTRT** (0), используется укороченное время запуска. По умолчанию этот бит не запрограммирован (1). Можно заказывать микросхемы с предварительно запрограммированным битом.

Эти биты недоступны при последовательном программировании и не изменяются при стирании памяти.

## Код устройства

Все микроконтроллеры фирмы *Atmel* имеют 3-байтовый сигнатурный код, по которому идентифицируется устройство. Этот код может быть прочитан в параллельном и последовательном режимах. Эти три байта размещены в отдельном адресном пространстве и для AT90S2313 имеют следующие значения:

- ♦ \$000: \$1E — код производителя — *Atmel*;
- ♦ \$001: \$91 — 2 кБ флэш-памяти;
- ♦ \$002: \$01 — при \$01=\$91 — м/сх AT90S2313.

Если запрограммированы биты блокировки, байты сигнатуры в последовательном режиме не читаются.

## Программирование флэш-памяти и EEPROM

AT90S2313 имеет 2кБ перепрограммируемой флэш-памяти программ и 128 байт энергонезависимой памяти данных.

При поставке AT90S2313 флэш-память и память данных стерты (содержат \$FF) и готовы к программированию. Эта микросхема поддерживает высоковольтный (12В) параллельный режим программирования и низковольтный режим последовательного программирования. Напряжение +12В используется только для разрешения программирования, этот вывод не потребляет тока. Последовательный режим программирования предусмотрен для загрузки программы и данных в AT90S2313 в системе пользователя (внутрисистемное программирование).

Память программ и данных в AT90S2313 программируются байт за байтом в обоих режимах программирования. Для энергонезависимой памяти предусмотрен режим автоматического стирания при последовательном программировании.

## Параллельное программирование

Ниже рассмотрено параллельное программирование флэш-памяти программ, энергонезависимой памяти данных, битов блокировки и конфигурации. Некоторые выводы AT90S2313 ниже называются именами, отражающими функциональное назначение сигналов при параллельном программировании. Выводы, не приведенные в табл. П4.21, называются своими обычными именами.

**Таблица П4.21. Имена выводов**

Имя сигнала при программировании	Имя вывода	Вх./вых.	Функция
RDY/BSY	PD1	Вых	0: м/сх занята программированием; 1: м/сх готова к приему команды
OE	PD2	Вх	Разрешение выходов (активный 0)
WR	PD3	Вх	Импульс записи (активный 0)
BS	PD4	Вх	Выбор байта
XA0	PD5	Вх	Действие XTAL бит 0
XA1	PD6	Вх	Действие XTAL бит 1

Биты XA0 и XA1 определяют действие, происходящее по положительному импульсу XTAL1. Установки битов приведены в табл. П4.22.

**Таблица П4.22. Установка XA1 и XA0**

XA1	XA0	Действие при подаче импульса XTAL1
0	0	Загрузка адреса памяти программ или данных (старший/младший байт задается выводом BS)
0	1	Загрузка данных (старший/младший байт для флэш-памяти задается выводом BS)
1	0	Загрузка команды
1	1	Не работает, холостой ход

По импульсу WR или OE загружается команда, определяющая действие по вводу или выводу. В байте команды каждому биту присвоена функция, как показано в табл. П4.23.

**Таблица П4.23.** Биты конфигурации командного байта

Бит	Значение при установке
7	Стирание кристалла
6	Запись битов конфигурации, расположенных в следующих позициях байта данных: D5-SPIEN, D0-RCEN (0 для программирования, 1 для стирания)
5	Запись битов блокировки, расположенных в следующих позициях байта данных: D2-LB2, D1-LB1 (для программирования устанавливать 0)
4	Запись памяти программ или данных (определяется битом 0)
3	Чтение сигнатуры
2	Чтение битов блокировки и конфигурации, расположенных в следующих позициях байта данных: D7-LB1, D6-LB2, D5-SPIEN, D0-RCEN
1	Чтение памяти программ или данных (определяется битом 0)
0	0 - доступ к памяти программ, 1 - к памяти данных

### Вход в режим программирования

Следующий алгоритм переводит устройство в режим параллельного программирования.

1. Подать напряжение 4.5–5.5 В на выводы питания.
2. Установить RESET и BS в 0 и выдержать не менее 100 нс.
3. Подать 12В на RESET и выждать не менее 100 нс перед изменением BS. Любая активность на выводе BS в течение этого времени приведет к тому, что микросхема не включится в режим программирования.

### Стирание кристалла

При стирании кристалла стираются память программ и данных, а также биты блокировки. Биты блокировки не сбрасываются до полного стирания памяти программ и данных. Биты конфигурации не изменяются. (Стирание кристалла производится перед его программированием).

### Загрузка команды «стереть кристалл»

1. Установить XA1, XA0 в '10'. Это разрешает загрузку команды.
2. Установить BS в 0.
3. Установить PB в '1000 0000'. Это команда стирания кристалла.
4. Подать положительный импульс на XTAL1. При этом загружается команда и начинается стирание памяти программ и данных. После импульса XTAL1, подать отрицательный импульс на WR, чтобы раз решить стирание битов блокировки по окончании цикла стирания.

Подождать около 10 мс до окончания цикла стирания. Стирание кристалла не генерирует сигнала RDY/BSY.

## Программирование флэш-памяти

Для программирования флэш-памяти необходимо выполнить следующие действия.

1. Загрузить команду «программирование флэш-памяти»:
  - ♦ установить XA1,XA0 в '10'. Это разрешает загрузку команды;
  - ♦ установить BS в 0;
  - ♦ установить PB в '0001 0000'. Это команда программирования флэш-памяти
  - ♦ подать положительный импульс на XTAL1. При этом загружается команда.
2. Загрузить младший байт адреса:
  - ♦ установить XA1,XA0 в '00'. Это разрешает загрузку адреса;
  - ♦ установить BS в 0. Это выбирает младший байт адреса;
  - ♦ установить на PB младший байт адреса;
  - ♦ подать положительный импульс на XTAL1. При этом загружается младший байт адреса.
3. Загрузить старший байт адреса:
  - ♦ установить XA1,XA0 в '00'. Это разрешает загрузку адреса;
  - ♦ установить BS в 1. Это выбирает старший байт адреса;
  - ♦ установить на PB старший байт адреса. (\$00..\$01);
  - ♦ подать положительный импульс на XTAL1. При этом загружается старший байт адреса.
4. Загрузить байт данных:
  - ♦ установить XA1,XA0 в '01'. Это разрешает загрузку данных;
  - ♦ установить на PB младший байт данных;
  - ♦ подать положительный импульс на XTAL1. При этом загружается младший байт данных.
5. Запись младшего байта данных:
  - ♦ установить BS в '0'. Это выбирает младший байт данных;
  - ♦ подать на WR отрицательный импульс. Это инициирует программирование байта. RDY/BSY переходит в низкое состояние;
  - ♦ перед программированием следующего байта подождать, пока RDY/BSY перейдет в высокое состояние.
6. Загрузить байт данных:
  - ♦ установить XA1,XA0 в '01'. Это разрешает загрузку данных;
  - ♦ установить на PB старший байт данных;
  - ♦ подать положительный импульс на XTAL1. При этом загружается старший байт данных.
7. Запись старшего байта данных:
  - ♦ установить BS в '1'. Это выбирает старший байт данных;

- ♦ подать на WR отрицательный импульс. Это инициирует программирование байта. RDY/BSY переходит в низкое состояние;
- ♦ перед программированием следующего байта подождать, пока RDY/BSY перейдет в высокое состояние.

Загруженные адрес и данные сохраняются в устройстве после программирования, при этом процесс программирования упрощается.

- Команду программирования флэш-памяти необходимо подать только перед программированием первого байта
- Старший байт адреса можно менять только перед программированием следующей страницы памяти программ (256 слов).

### **Программирование памяти данных**

Алгоритм программирования памяти данных следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0001 0001'.
2. Загрузить младший адрес памяти данных (\$00-\$7F).
3. Загрузить младший байт данных.
4. Подать отрицательный импульс на WR и подождать перевода RDY/BSY в '1'.

Загрузка команды необходима только перед программированием первого байта.

### **Чтение памяти программ**

Алгоритм чтения флэш-памяти следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0000 0010'.
2. Загрузить младший байт адреса (\$00-\$FF).
3. Загрузить старший байт адреса (\$00-\$03).
4. Установить OE в '0', BS в '0'. Теперь на выводах PB можно прочесть младший байт данных.
5. Установить BS в '1'. Теперь на выводах PB можно прочесть старший байт данных.
6. Установить OE в '1'.

Загрузка команды необходима только перед чтением первого байта.

### **Чтение памяти данных**

Алгоритм чтения памяти данных следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0000 0011'.
2. Загрузить младший байт адреса (\$00-\$7F).
3. Установить OE в '0', BS в '0'. Теперь на выводах PB можно прочесть байт данных.
4. Установить OE в '1'.

Загрузка команды необходима только перед чтением первого байта.

### Программирование битов конфигурации

Алгоритм программирования битов конфигурации следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0100 0000'.
2. Загрузить данные:
  - ♦ бит5=0 — запрограммировать бит SPIEN, бит5=1 — стереть бит SPIEN;
  - ♦ бит0=0 — запрограммировать бит FSTRT, бит0=1 — стереть бит FSTRT.
3. Подать на WR отрицательный импульс и ждать перехода RDY/BSY в '1'.

---

**Внимание!** WR должен удерживаться в низком состоянии не меньше 1мс.

---

### Программирование битов блокировки

Алгоритм программирования битов блокировки следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0010 0000'.
2. Загрузить данные:
  - ♦ бит2=0 — запрограммировать бит LB2;
  - ♦ бит1=0 — запрограммировать бит LB1.
3. Подать на WR отрицательный импульс и ждать перехода RDY/BSY в '1'.

Биты блокировки стираются только при стирании всей микросхемы.

### Чтение битов конфигурации и блокировки

Алгоритм чтения битов блокировки и конфигурации следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0000 0100'.
2. Установить OE в '0', BS в '1'. Теперь на выводах PB можно прочитать биты блокировки и конфигурации:
  - бит7 — LB1 — '0' запрограммирован;
  - бит6 — LB2 — '0' запрограммирован;
  - бит5 — SPIEN — '0' запрограммирован;
  - бит0 — FSTRT — '0' запрограммирован.
3. Установить OE в '1'.

Следите, чтобы бит BS был установлен в '1'.



## Чтение байтов сигнатуры

Алгоритм чтения байтов сигнатуры следующий (обратитесь к программированию памяти программ за описанием загрузки команды, адреса и данных).

1. Загрузить команду '0000 1000'.
2. Загрузить младший байт адреса (\$00-\$02), установить OE и BS в '0'. После этого выбранный байт сигнатуры можно прочитать на выводах PB.
3. Установить OE в '1'.

Команду необходимо выполнять только перед чтением первого байта.

## Последовательная загрузка

Память данных, как память программ может быть запрограммирована с использованием последовательной шины SPI, при этом вывод RESET должен быть подключен к земле. Последовательный интерфейс работает с выводами SCK, MOSI (вход) и MISO (выход). После подачи низкого уровня на RESET перепрограммированием/стиранием необходимо исполнить команду разрешения программирования.

При программировании памяти данных, во внутренний алгоритм программирования встроен цикл стирания (только при последовательном программировании), поэтому нет необходимости в выполнении команды стирания памяти. Команда стирания микросхемы переводит все ячейки памяти программ и данных в состояние \$FF.

Флэш-память программ и энергонезависимая память данных имеют отдельное адресное пространство: \$000-\$7FF для памяти программ и \$000-\$07F для памяти данных.

При программировании необходимо подавать внешнюю тактовую частоту на вывод XTAL1 или подключить внешний тактовый генератор к выводам XTAL1 и XTAL2. Минимальные длительности низкого и высокого уровня сигнала SCK определены следующим образом.

- Низкий: > 2 периодов XTAL1.
- Высокий: > 2 периодов XTAL1.

## Подтверждение данных

До завершения программирования нового байта, при чтении из памяти будет читаться значение \$7F. Когда микросхема будет готова для записи следующего байта, будет выдаваться записанное значение. Это используется для определения момента, когда можно записывать следующий байт. Этот способ не будет работать для байта \$7F, поэтому для записи этого числа перед программированием следующего байта придется выждать, по крайней мере, 4мс. Поскольку после стирания ячейки в ней устанавливается \$FF, при программировании ячейки, содержащие \$FF можно пропускать. Это не применимо при перезаписи EEPROM без стирания памяти программ. В этом случае подтверждение не работает для данных \$7F и \$FF, для этих значений перед программированием следующего байта необходимо выждать 4мс.

## Алгоритм последовательного программирования

Для программирования и проверки AT90S2313 в режиме последовательного программирования рекомендуется следующая последовательность действий (см. формат четырех байтовой команды в табл. П4.25).

Последовательность включения питания

1. Подать напряжение питания между VCC и GND, при этом RESET и SCK должны быть установлены в '0'. (Если программатор не гарантирует установки SCK в '0' при подаче питания, тогда после того как на SCK будет установлен '0', на RESET необходимо подать положительный импульс. Если кварцевый резонатор не подключен к выводам XTAL1 и XTAL2, подайте частоту от 0 до 16 МГц на вывод XTAL1.
2. Подождать 20 мс и разрешить последовательное программирование, послав команду разрешения на вывод MOSI/PB5.
3. При послышке трехбайтовой последовательности разрешения программирования, второй байт последовательности (\$53) будет возвращен при послышке третьего байта. В любом случае, должны быть посланы все четыре байта команды. Если число \$53 не получено обратно, подайте положительный импульс на SCK и повторите команду разрешения программирования. Если после 32 попыток не будет получено число \$53, микросхема неисправна.
4. При подаче команды стирания стирать микросхему (всегда при программировании памяти программ), подождать 10 мс, выдать положительный импульс на RESET и повторить п.2.
5. Память программ и память данных программируются по одному байту выдачей адреса и данных в команде записи. Перед записью новых данных в EEPROM ячейки памяти перед записью новых данных автоматически стираются. Чтобы определить время, когда можно записывать следующий байт, используется подтверждение данных. При записи предварительно стертой микросхемы записывать ячейки содержащие \$FF не обязательно. При записи в ячейку числа \$7F, перед программированием следующего адреса необходимо выждать 4 мс.
6. Любую ячейку памяти можно проверить, используя команду чтения, которая выдает содержимое указанной ячейки на последовательный вывод MISO/PB6.
7. По окончании программирования вывод RESET может быть установлен в '1' для возобновления нормальной работы схемы.

Последовательность выключения питания (если необходимо).

1. Установить XTAL1 в '0' (если не используется кварцевый резонатор).
2. Установить RESET в '1'.
3. Отключить питание.

При записи последовательных данных в AT90S2313 данные читаются по нарастающему фронту сигнала CLK. При чтении данных из AT90S2313 данные читаются по спадающему фронту сигнала CLK.

Таблица П4.25. Команды последовательного программирования для AT90S2313

Команды	Формат команды				Действие
	Байт 1	Байт 2	Байт 3	Байт 4	
Разрешение программир.	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Разрешение последовательного программир. после установки RESET в '0'
Стирание микросхем	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Стирание памяти прог. и данных
Чтение памяти программир.	0010 H000	0000 00aa	bbbb bbbb	oooo oooo	Читать старший или младший байт 0 по адресу ab
Запись памяти программир.	0100 H000	0000 00aa	bbbb bbbb	iiii iiii	Запись старш. или младш. байта i в память прог. по адресу ab
Чтение памяти данных	1010 0000	0000 0000	xbbb bbbb	oooo oooo	Читать данные 0 по адресу b
Запись памяти данных	1100 0000	0000 0000	xbbb bbbb	iiii iiii	Запись данных i по адресу b
Запись битов блокировки	1010 1100	111x x21x	xxxx xxxx	xxxx xxxx	Запись битов блокировки. Установить 1,2=0 для блокировки памяти
Читать код устройства	0010 0000	xxxx xxxx	xxxx xxbb	oooo oooo	Читать код устройства 0 с адреса b

Где a — старшие биты адреса; b — младшие биты адреса; o — выход данных; H = '0' — младший байт, '1' — старший байт; i — вход данных; x — не используется; 1 — 1-й бит блокировки; 2 — 2-й бит блокировки.

### Максимально допустимые параметры

Рабочая температура..... $-55^{\circ}\text{C}$  ...  $+125^{\circ}\text{C}$   
Температура хранения..... $-65^{\circ}\text{C}$  ...  $+150^{\circ}\text{C}$   
Напряжение на любом выводе кроме RESET..... $-1,0\text{ В}$  ...  $+7,0\text{ В}$   
Максимальное рабочее напряжение..... $6,6\text{ В}$   
Постоянный ток через вывод порта..... $40,0\text{ мА}$   
Постоянный ток между VCC и GND..... $140,0\text{ мА}$

Выход параметров за пределы указанные в таблице может привести к нарушению работоспособности микросхемы. Это предельные значения параметров, рабочие параметры микросхемы приведены выше.

Удержание предельных значений на выводах микросхемы в течение длительного времени может привести к потере ее работоспособности.

### Характеристики по постоянному току

$T_a = -40 \dots +85^{\circ}\text{C}$ ,  $V_{cc} = 2,7 \dots 6,0\text{ В}$  (если не указано иначе).  
Мин./Тип./Макс — эти значения показаны в следующих строках.  
Входное напряжение '0' —  $-0,5 / \dots / 0,2V_{cc} - 0,1\text{ В}$ .  
Входное напряжение '1' (кроме XTAL1 и RESET) —  $-0,2V_{cc} + 0,9 / \dots / V_{cc} + 0,5\text{ В}$ .

На XTAL1 и RESET —  $-0,7V_{CC} / \dots / V_{CC} + 0,5 \text{ В}$ .  
 Выходное напряжение '0' (порты В и D)  $I = 20 \text{ мА}$ ,  $V_{CC} = 5 \text{ В}$  —  $/0,5 \text{ В}$ .  
 $I = 10 \text{ мА}$ ,  $V_{CC} = 2,7 \text{ В}$  —  $/0,5 \text{ В}$ .  
 Выходное напряжение '1' (порты В и D)  $I_{OH} = 10 \text{ мА}$ ,  $V_{CC} = 5 \text{ В}$  —  $4,5 / \dots / \text{ В}$ .  
 $I_{OH} = 5 \text{ мА}$ ,  $V_{CC} = 2,7 \text{ В}$  —  $4,5 / \dots / \text{ В}$ .  
 Выходной ток (порты В и D)  $V_{CC} = 5 \text{ В}$  —  $/ \dots / 10 \text{ мА}$ ;  $V_{CC} = 2,7 \text{ В}$  —  $/ \dots / 5 \text{ мА}$ .  
 Поглощаемый ток (порты В и D)  $V_{CC} = 5 \text{ В}$  —  $/ \dots / 20 \text{ мА}$ ;  $V_{CC} = 2,7 \text{ В}$  —  $/ \dots / 10 \text{ мА}$ .  
 Подтягивающий резистор сброса —  $100 / \dots / 500 \text{ кОм}$ .  
 Подтягивающий резистор вывода порта —  $35 / \dots / 120 \text{ кОм}$ .  
 Потребляемый ток.  
 Активный режим, 3 В, 4 МГц —  $2,5 \text{ мА}$ .  
 Холостой ход (idle mode), 3 В, 4 МГц —  $800 \text{ мА}$ .  
 Пониженное потребление WDT включен, 3 В —  $50 \text{ мА}$ .  
 WDT выключен, 3 В —  $1 \text{ мА}$ .  
 Напряжение смещения аналогового компаратора  $V_{CC} = 5 \text{ В}$  —  $20 \text{ мВ}$ .  
 Входной ток утечки аналогового компаратора —  $10 \text{ нА}$ .  
 Время срабатывания аналогового компаратора  $V_{CC} = 2,7 \text{ В}$  —  $750 \text{ нс}$ .  
 $V_{CC} = 4,0 \text{ В}$  —  $500 \text{ нс}$ .

В рабочем состоянии ток через выводы должен ограничиваться следующими условиями:

- максимальный ток через вывод —  $20 \text{ мА}$ ;
- максимальный ток через все выводы —  $80 \text{ мА}$ .

Минимальное напряжение для режима пониженного потребления —  $2 \text{ В}$ .

Параметры внешнего тактового сигнала показаны в табл. П4.26

**Таблица П4.26.** Параметры внешнего тактового сигнала

Параметр	$V_{CC} = 2,7\text{--}6,6 \text{ В}$		$V_{CC} = 4,0\text{--}6,6 \text{ В}$		Единицы измерения
	min	max	min	max	
Частота	0	10	0	10	МГц
Период	100		100		нс
Длительность '1'	0		0		нс
Длительность '0'	0		0		нс
Длительность фронта		1,6		0,5	мкс

Информация для заказа микроконтроллеров показана в табл. П4.27.

**Таблица П4.27.** Информация для заказа

Частота	Напряжение питания	Маркировка	Корпус	Диапазон температур
4 MHz	2,7... 6,0В	AT90S2313-4PC	20P3	Коммерческий (0° С ... 70° С) Промышленный (-40° С ... 85° С)
		AT90S2313-4SC	20S	
		AT90S2313-4PI	20P3	
		AT90S2313-4SI	20S	
10 MHz	4,0 ... 6,0В	AT902313-10PC	20P3	Коммерческий (0° С ... 70° С) Промышленный (-40° С ... 85° С)
		AT90S2313-10SC	20S	
		AT90S2313-10PI	20P3	
		AT902313-10SI	20S	

Корпус 20P3 — PDIP, 20S — SOIC.

Список регистров приведен в табл. П4.28.

Адр	Имя	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$3F \$5F	SREG	I	T	H	S	V	N	Z	C
\$3E \$5E	Зарезервирован								
\$3D \$5D	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
\$3C \$5C	Зарезервирован								
\$3B \$5B	GIMSK	INT1	INT0	—	—	—	—	—	—
\$3A \$5A	GIFR	INTF1	INTF0	—	—	—	—	—	—
\$39 \$59	TIMSK	T9IE1	OC1E1A	—	—	TIC1E1	—	TO1E0	—
\$38 \$58	TIFR	TOV1	OCF1A	—	—	ICF1	—	TOV0	—
\$37 \$57	Зарезервирован								
\$36 \$56	Зарезервирован								
\$35 \$55	MCUCR	—	—	SE	SM	ISC11	ISC10	ISC01	ISC00
\$34 \$54	Зарезервирован								
\$33 \$53	TCCR0	—	—	—	—	—	CS02	CS01	CS00
\$32 \$52	TCNT0	Таймер/счетчик (8 бит)							
\$31 \$51	Зарезервирован								
\$30 \$50	Зарезервирован								
\$2F \$4F	TCCR1A	COM1A1	COM1A0	—	—	—	—	PWM11	PWM10
\$2E \$4E	TCCR1B	ICNC1	ICES1	—	—	CTC1	CS12	CS11	CS10
\$2D \$4D	TCNT1H	Таймер/счетчик 1 старший байт							
\$2C \$4C	TCNT1D	Таймер/счетчик 1 младший байт							
\$2B \$4B	OCR1AH	Таймер/счетчик 1 регистр сравнения старший байт							
\$2A \$4A	OCR1AL	Таймер/счетчик 1 регистр сравнения младший байт							
\$25 \$45	ICR1H	Таймер/счетчик 1 регистр захвата старший байт							
\$24 \$44	ICR1L	Таймер/счетчик 1 регистр захвата младший байт							

Таблица П4.28. Продолжение

Адр	Имя	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$23 \$43	Зарезервирован								
\$22 \$42	Зарезервирован								
\$21 \$41	WDTCR	—	—	—	WDTOE	WDE	WDP2	WDP1	WDP0
\$20 \$40	Зарезервирован								
\$1F \$3F	Зарезервирован								
\$1E \$3E	EEAR	Регистр адреса энергонезависимой памяти							
\$1D \$3D	EEDR	Регистр данных энергонезависимой памяти							
\$1C \$3C	EECR	—	—	—	—	—	EEMWE	EEWE	EERE
\$1B \$3B	Зарезервирован								
\$1A \$3A	Зарезервирован								
\$19 \$39	Зарезервирован								
\$18 \$38	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
\$17 \$37	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
\$16 \$36	PINB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
\$15 \$35	Зарезервирован								
\$14 \$34	Зарезервирован								
\$13 \$33	Зарезервирован								
\$12 \$32	PORTD	—	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
\$11 \$31	DDRD	—	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
\$10 \$30	PIND	—	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
\$0F \$2F	Зарезервирован								
\$0C \$2C	UDR	Регистр данных UART							
\$0B \$2B	USR	RXC	TXC	UDRE	FE	OR	—	—	—
\$0A \$2A	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXBB	TXBB

Таблица П4.28. Окончание

Адр	Имя	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$09 \$29	UBRR	Регистр скорости передачи UART							
\$08 \$28	ACSR	ACD	—	ACO	ACI	ACIE	—	ACIS1	ACIS0
\$00 \$20	Зарезервирован								

## Набор команд МК AT90S2313

В табл. П4.29 и П4.30 приведен полный набор команд, используемых при создании программ для управления микроконтроллерами фирмы *Atmel*.

Таблица П4.29. Набор команд AT902313

Мне-моника	Операнды	Описание	Действие	Флаги	Цикл
АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ КОМАНДЫ					
ADD	Rd, Rr	Сложить два регистра	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
ADC	Rd, Rr	Сложить с переносом	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
ADIM	Rd1, k	Сложить слово с константой	$Rdh, 1 \leftarrow Rdh, 1 + K$	Z, C, N, V, S	2
SUB	Rd, Rr	Вычесть два регистра	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
SUBI	Rd, K	Вычесть константу	$Rd \leftarrow Rd - K$	Z, C, N, V, H	1
SBIW	Rd1, K	Вычесть слово с константой	$Rdh, 1 \leftarrow Rdh, 1 - K$	Z, C, N, V, S	2
SBC	Rd, Rr	Вычесть с переносом	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H	1
SBCI	Rd, K	Вычесть с переносом	$Rd \leftarrow Rd - K - C$	Z, C, N, V, H	1
AND	Rd, Rr	Логическое И	$Rd \leftarrow Rd \text{ AND } Rr$	Z, N, V	1
ANDI	Rd, K	Логическое И	$Rd \leftarrow Rd \text{ AND } K$	Z, N, V	1
OR	Rd, Rr	Логическое ИЛИ	$Rd \leftarrow Rd \text{ OR } Rr$	Z, N, V	1
ORI	Rd, K	Логическое ИЛИ	$Rd \leftarrow Rd \text{ OR } K$	Z, N, V	1
EOR	Rd, Rr	Исключающее ИЛИ	$Rd \leftarrow Rd \text{ XOR } Rr$	Z, N, V	1
COM	Rd	Дополнение до 1	$Rd \leftarrow \$FF - Rd$	Z, C, N, V	1
NEG	Rd	Дополнение до 2	$Rd \leftarrow \$00 - Rd$	Z, C, N, V, H	1
SBR	Rd, K	Установить бит(ы) в регистре	$Rd \leftarrow Rd \text{ OR } K$	Z, N, V	1
CBR	Rd, K	Сбросить бит(ы) в регистре	$Rd \leftarrow Rd \text{ AND } (FFh - K)$	Z, N, V	1
INC	Rd	Увеличить до 1	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd	Уменьшить до 1	$Rd \leftarrow Rd - 1$	Z, N, V	1
TST	Rd	Проверить на 0 или др. отрицательное значение	$Rd \leftarrow Rd \text{ AND } Rd$	Z, N, V	1
CLR	Rd	Очистить регистр	$Rd \leftarrow Rd \text{ XOR } Rd$	Z, N, V	1
SER	Rd	Установить регистр	$Rd \leftarrow \$FF$	None	1
КОМАНДЫ ВЕТВЛЕНИЯ					
RJMP	k	Относительный переход	$PC \leftarrow PC + k + 1$	None	2
LJMP		Переход по адресу (z)	$PC \leftarrow Z$	None	2
RCALL	k	Относительный вызов подпрограммы	$PC \leftarrow PC + k + 1$	None	3
ICALL		Вызов подпрограммы по адресу (z)	$PC \leftarrow Z$	None	3

Таблица П4.29. Продолжение

Мне-моника	Операнды	Описание	Действие	Флаги	Цикл
RET		Выход из подпрограммы	PC<STACK	None	4
RETI		Выход из прерывания	PC<STACK		4
CPSE	Rd, Rr	Сравнить два регистра и пропускает следующую команду, если равно	If(Rd=Rr) PC<PC+2 или 3	None	1/2
CP	Rd, Rr	Сравнить два регистра	Rd – Rr		1
CPC	Rd, Rr	Сравнить с переносом	Rd – Rr – C		1
CPI	Rd, K	Сравнить с константой	Rd – K		1
SBRC	Rr, b	Пропуск, если бит в регистре сброшен	If(Rr(b)=0) PC<PC+2 или 3	None	1/2
SBRS	Rr, b	Пропуск если бит в регистре установлен	if(Rr(b)=1) PC<PC+2 или 3	None	1/2
SBIC	P, b	Пропуск если бит в регистре ввода/вывода сброшен	If(P(b)=0) PC<PC+2 или 3	None	1/2
SBIS	P, b	Пропуск если бит в регистре ввода/вывода установлен	If(P(b)=1) PC<PC+2 или 3	None	1/2
BRBS	s, k	Переход если установлен флаг s	if(SREG(s)=1) PC<PC+k+1	None	1/2
BRBC	s, k	Переход если сброшен флаг s	if(SREG(s)=0) PC<PC+k+1	None	1/2
BREQ	k	Переход если равно	if(Z=1) PC<PC+k+1	None	1/2
BRNE	k	Переход если не равно	if(Z=0) PC<PC+k+1	None	1/2
BRCS	k	Переход если установлен перенос	if(C=1) PC<PC+k+1	None	1/2
BRCC	k	Переход если сброшен перенос	if(C=0) PC<PC+k+1	None	1/2
BRSH	k	Переход если равно или больше	if(C=0) PC<PC+k+1	None	1/2
BRLO	k	Переход если меньше	if(C=1) PC<PC+k+1	None	1/2
BRMI	k	Переход если минус	if(N=1) PC<PC+k+1	None	1/2
BRPL	k	Переход если плюс	if(N=0) PC<PC+k+1	None	1/2
BRGE	k	Переход если больше или равно, со знаком	if(N XOR V=0) PC<PC+k+1	None	1/2
BRLT	k	Переход если меньше нуля, со знаком	if(N XOR =1) PC<PC+k+1	None	1/2
BRHS	k	Переход если установлен флаг H	if(H=1) PC<PC+k+1	None	1/2
BRHC	k	Переход если сброшен флаг H	if(H=0) PC<PC+k+1	None	1/2
BRTS	k	Переход если установлен флаг T	if(H=1) PC<PC+k+1	None	1/2
BRTC	k	Переход если сброшен флаг T	if(H=0) PC<PC+k+1	None	1/2
BRVS	k	Переход если установлен флаг V (переполнение)	if(H=1) PC<PC+k+1	None	1/2
BRVC	k	Переход если сброшен флаг V	if(H=0) PC<PC+k+1	None	1/2
BRIE	k	Переход если разрешены прерывания	if(I=1) PC<PC+k+1	None	1/2
BRID	k	Переход если запрещены прерывания	if(I=1) PC<PC+k+1	None	1/2
КОМАНДЫ ПЕРЕСЫЛКИ					
MOV	Rd, Rr	Пересылка между регистрами	Rd<Rr		
LDI	Rd, K	Загрузить константу	Rd<k		



Таблица П4.29. Окончание

Мне- моника	Опе- ранды	Описание	Действие	Флаги	Цикл
LD	Rd, X	Загрузить регистр непосредственно	$Rd \leftarrow (X)$		
LD	Rd, X+	Загрузить регистр непосредственно с пост инкрементом	$Rd \leftarrow (X), X \leftarrow X+1$		
LD	Rd, -X	Загрузить регистр непосредственно с пред. декрементом	$x \leftarrow x-1, Rd \leftarrow (x)$		
LD	Rd, Y	Загрузить регистр непосредственно	$Rd \leftarrow (Y)$		
LD	Rd, Y+	Загрузить регистр непосредственно с пост инкрементом	$Rd \leftarrow (Y), Y \leftarrow Y+1$		
LD	Rd, -Y	Загрузить регистр непосредственно с пред. декрементом	$Y \leftarrow Y-1, Rd \leftarrow (Y)$		
LDD	Rd, Y+q	Загрузить регистр непосредственно со смещен.	$Rd \leftarrow (Y+q)$		
LD	Rd, Z	Загрузить регистр непосредственно	$Rd \leftarrow (Z)$	None	1
LD	Rd, Z+	Загрузить регистр непосредственно с пост инкрементом	$Rd \leftarrow (Z), Z \leftarrow Z+1$	None	1
LD	Rd, -Z	Загрузить регистр непосредственно с предв. декрементом	$Z \leftarrow Z-1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Загрузить регистр непосредственно со смещением	$Rd \leftarrow (Z+q)$	None	2

Таблица П4.30. Продолжение набора команд AT90S2313

Мнемо- ника	Опе- ранды	Описание	Действие	Флаги	Цикл
LDS	Rd, k	Загрузить из ОЗУ	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Записать регистр непосредственно	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Записать регистр непосредственно с пост. инкрементом	$(X) \leftarrow Rr, X \leftarrow X+1$	None	2
ST	-X, Rr	Записать регистр непосредственно с пред. декрементом	$X \leftarrow X-1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Записать регистр непосредственно	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Записать регистр непосредственно с пост. инкрементом	$(Y) \leftarrow Rr, Y \leftarrow Y+1$	None	2
ST	-Y, Rr	Записать регистр непосредственно с пред. декрементом	$Y \leftarrow Y-1, (Y) \leftarrow Rr$	None	2
STD	Y+q, Rr	Записать регистр непосредственно со смещен.	$(Y+q) \leftarrow Rr$	None	2
ST	Z, Rr	Записать регистр непосредственно	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Записать регистр непосредственно с пост. инкрементом	$(Z) \leftarrow Rr, Z \leftarrow Z+1$	None	2
ST	-Z, Rr	Записать регистр непосредственно с пред. декрементом	$Z \leftarrow Z-1, (Z) \leftarrow Rr$	None	2

Таблица П4.30. Окончание

Мнемо-ника	Операнды	Описание	Действие	Флаги	Цикл
STD	Y+q, Rr	Записать регистр непосредственно со смещен.	$(Y+q) < Rr$	None	2
STS	k, Rr	Записать в ОЗУ	$(k) < Rr$	None	3
LPM		Загрузить из памяти программы	$R0 < (Z)$	None	3
IN	Rd, P	Ввод из порта	$Rd < P$	None	1
OUT	P, Rr	Вывод в порт	$P < Rr$	None	1
PUSH	Rr	Записать в стек	$STACK < Rr$	None	2
POP	Rr	Прочитать из стека	$Rr < STACK$	None	2
КОМАНДЫ РАБОТЫ С БИТАМИ					
SBI	P, b	Установить бит в регистр ввода/вывода	$I/O(P, b) < 1$	None	2
CBI	P, b	Сбросить бит в регистр ввода/вывода	$I/O(P, b) < 0$	None	2
LSL	Rd	Логический сдвиг влево	$Rd(n+1) < Rd(n), Rd(0) < 0$	Z, C, N, V	1
LSR	Rd	Логический сдвиг вправо	$Rd(n) < Rd(n+1), Rd(7) < 0$	Z, C, N, V	1
ROL	Rd	Сдвиг влево через C	$Rd(0) < C, C < Rd(7), Rd(n+1) < Rd(n)$	Z, C, N, V	1
ROR	Rd	Сдвиг вправо через C	$Rd(7) < C, C < Rd(0), Rd(n) < Rd(n+1)$	Z, C, N, V	1
ASR	Rd	Арифметический сдвиг вправо	$Rd(n) < Rd(n+1), n=0-6$	Z, C, N, V	1
SWAP	Rd	Обмен местами в тетради регистра	$Rd(3-0) < Rd(7-4), Rd(7-4) < Rd(3-0)$	None	1
BSET	s	Установить флаг	$SREG(s) < 1$	SREG(s)	1
BCLR	s	Сбросить флаг	$SREG(s) < 0$	SREG(s)	1
BST	Rr, b	Запомнить бит в T	$T < Rr(b)$	T	1
BLD	Rd, b	Прочитать бит из T	$Rd(b) < T$	None	1
SEC		Установить перенос	$C < 1$	C	1
CLC		Сбросить перенос	$N < 0$	C	1
SEN		Установить флаг N	$N < 1$	N	1
CLN		Сбросить флаг N	$N < 0$	N	1
SEZ		Установить флаг Z	$Z < 1$	Z	1
CLZ		Сбросить флаг Z	$Z < 0$	Z	1
SEI		Разрешить прерывания	$I < 1$	I	1
CLI		Запретить прерывания	$I < 0$	I	1
SES		Установить флаг S	$S < 1$	S	1
CLS		Сбросить флаг S	$S < 0$	S	1
SEV		Установить флаг V	$V < 1$	V	1
CLV		Сбросить флаг V	$V < 0$	V	1
SET		Установить флаг T	$T < 1$	T	1
CLT		Сбросить флаг T	$T < 0$	T	1
SEH		Установить флаг H	$H < 1$	H	1
CLH		Сбросить флаг H	$H < 0$	H	1
NOP		Нет операции		None	1
SLEEP		Останов		None	3
WDR		Сброс сторожевого таймера		None	1