

- (перевод с англ. Ю.Андрienко) -
- Fidonet - 2:461/44.14, -
- E-Mail - yurik@aqua.kharkov.ua -
- Phone - (057-2) 98-07-58, 44-85-40 -

### Микроконтроллеры AT90S2323 и AT90S2343 фирмы Atmel

AT90S2323 и AT90S2343 - экономичные 8 битовые КМОП микроконтроллеры, построенные с использованием расширенной RISC архитектуры AVR. Исполняя по одной команде за период тактовой частоты, AT90S2323 и AT90S2343 имеют 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.

AT90S2323 и AT90S2343 предлагают следующие возможности: 2кБ загрузаемой флэш памяти; 128 байт EEPROM; 128 байт статического ОЗУ; 3 (90S2323) или 5 (90S2343) линий ввода/вывода общего назначения; 32 рабочих регистра; 8 разрядный таймер/счетчик; внешние и внутренние прерывания; 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, AT90S2323 и AT90S2343 являются мощными микроконтроллерами, которые позволяют создавать достаточно гибкие и эффективные по стоимости устройства.

AT90S2323 и AT90S2343 поддерживаются полной системой разработки включающей в себя компилятор Си, макроассемблер, программный отладчик/симулятор, внутрисхемный эмулятор и отладочный комплект.

### Сравнение 90S2323 и 90S2343

AT90S2323 разработаны для использования с внешним керамическим или кварцевым резонатором. Время запуска резонатора можно устанавливать как 1 мс (для керамических резонаторов) или 16мс (для кварцевых резонаторов). Микросхема имеет три ножки ввода/вывода.

AT90S2343 работает с внешним источником тактовых импульсов или со встроенным RC генератором. Микросхема имеет пять ножек ввода/вывода.

Различия между двумя микросхемами сведены в таблицу:

Таблица 1. Различия между микросхемами

М/сх	90S2323	90S2343
Усилитель генератора на кристалле	Да	Нет
Встроенный RC генератор	Нет	Да
Использование PB3	никогда	Работа со встроенным генератором
Использование PB4	никогда	Всегда
Время запуска	1mS/16mS	16uS

AT90S2343				AT90S2323			
-----TT-----				-----TT-----			
	-+1	L-	20+VCC		-+1	L-	20+VCC
/RESET	L+		+-	/RESET	L+		+-
	-+2		19+PB2/SCK/T0		-+2		19+PB2/SCK/T0
(CLOCK) PB3	L+		+-	XTAL1	L+		+-
	-+3		18+PB1/MISO/INT0		-+3		18+PB1/MISO/
PB4	L+		+-	XTAL2	L+		+-INT0
	-+10		11+PD0/ICP		-+10		11+
GND	L+		+-	GND	L+		+-PD0/ICP
	L-----				L-----		

PDIP/SOIC

#### ОПИСАНИЕ ВЫВОДОВ AT90S2323

VCC - вывод источника питания

GND - земля

Port B (PB2..PB0) - Порт В является 3-битовым двунаправленным портом ввода/вывода. Для выводов порта предусмотрены внутренние подтягивающие резисторы (выбираются для каждого бита).

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.

#### ОПИСАНИЕ ВЫВОДОВ AT90S2343

VCC - вывод источника питания

GND - земля

Port B (PB4..PB0) - Порт В является 3-битовым двунаправленным портом ввода/вывода. Для выводов порта предусмотрены внутренние подтягивающие резисторы (выбираются для каждого бита). При работе от внешнего тактового генератора вывод PB3 используется как вход тактирующего сигнала (CLOCK).

RESET - Вход сброса. Удержание на входе низкого уровня в течение

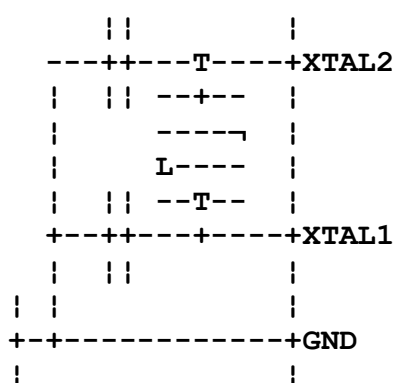
двух машинных циклов (если работает тактовый генератор), сбрасывает устройство.

XTAL1 - Вход инвертирующего усилителя генератора и вход внешнего тактового сигнала.

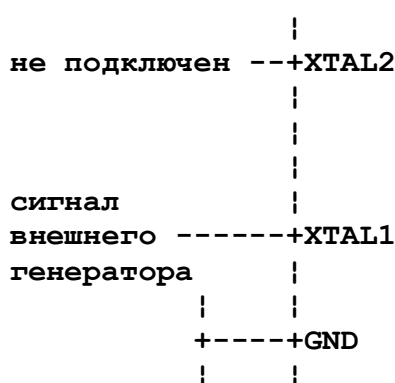
XTAL2 - Выход инвертирующего усилителя генератора.

#### ИСТОЧНИКИ ТАКТОВОГО СИГНАЛА

AT90S2323 содержит инвертирующий усилитель, который может использоваться для построения тактового генератора, как показано на рисунке ниже. XTAL1 и XTAL2 являются соответственно входом и выходом усилителя. Для генератора могут использоваться как керамические, так и кварцевые резонаторы. При использовании внешнего тактового сигнала рекомендуется применять AT90S2343, при этом добавляется одна ножка для ввода/вывода.



Подключение кристалла



Подключение внешнего генератора

AT90S2343 может использовать сигнал внешнего тактового генератора или встроенный RC генератор. Номинальная частота работы RC генератора - 1MHz ( $V_{CC}=5V$ ). При программировании бита конфигурации RCEN, расположенного во флэш-памяти, как источник сигнала используется встроенный генератор. AT90S2343 поставляются с запрограммированным битом RCEN.

#### Обзор архитектуры AT90S2323 и AT90S2343

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.

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, тамеры/счетчики и другие. Доступ к пространству ввода/вывода может осуществляться непосредственно, или 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.

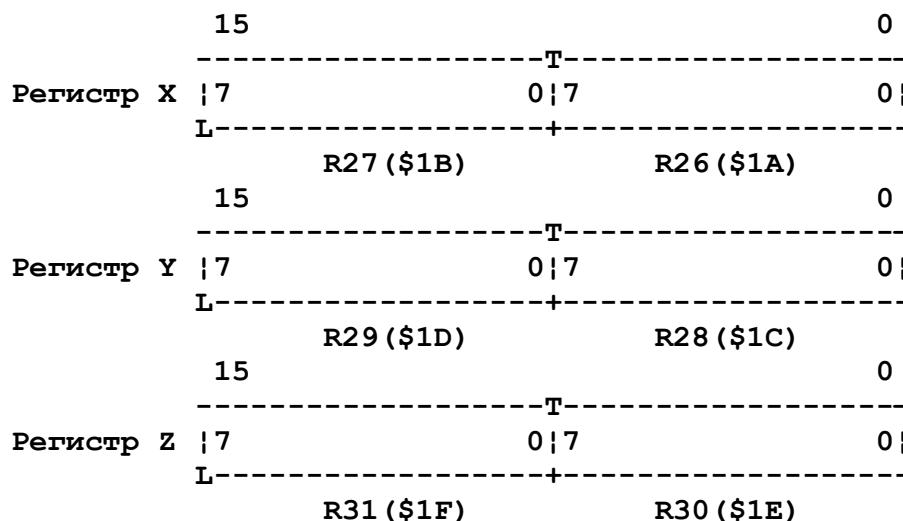
Гибкий модуль прерываний имеет собственный управляющий регистр в пространстве ввода/вывода, и флаг глобального разрешения прерываний в регистре состояния. Каждому прерыванию назначен свой вектор в начальной области памяти программ. Различные прерывания имеют приоритет в соответствии с расположением их 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 следующим образом:



При различных режимах адресации эти регистры могут использоваться 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.

АТ90S2323 и АТ90S2343 содержат 2кБ загружаемой флэш памяти для хранения программ. Поскольку все команды занимают одно 16- или 32-рядное слово, флэш память организована как 1Кх16. 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

АТ90S2323 и АТ90S2343 содержат 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 рисунке ниже показана организация памяти данных в АТ90S2323 и

AT90S2343.

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 для непосредственной адресации.

Прямая адресация имеет доступ ко всей памяти данных.

Непосредственная адресация со смещением используется для доступа к 63 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.

Регистровый файл

R0
+-----+
R1
+-----+
...
...
+-----+
R30
+-----+
R31
L-----

Регистры вв\выв.

\$00
+-----+
\$01
+-----+
...
...
+-----+
\$3E
+-----+
\$3F
L-----

Область адресов данных

\$00
+-----+
\$01
+-----+
...
...
+-----+
\$1E
+-----+
\$1F
L-----

\$20
+-----+
\$21
+-----+
...
...
+-----+
\$5E
+-----+
\$5F
L-----

Встроенное ОЗУ

\$60
+-----+
\$61
+-----+
...
...
+-----+

```

      |      $DE      |
      +-----+
      |      $DF      |
      L-----

```

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 MHz, уникальных показателей стоимости, быстродействия и потребления процессора.

Таблица 1. Пространство ввода/вывода AT90S2323 и AT90S2343

-----T-----T-----T-----			
\$3F(\$5F)	SREG	Status REGister	Регистр Состояния
-----+-----+-----+-----			
\$3D(\$5D)	SPL	Stack pointer low	Указатель стека, мл. байт
-----+-----+-----+-----			
\$3B(\$5B)	GIMSK	General Interrupt	Общий регистр маски
		MaSK register	прерываний
-----+-----+-----+-----			
\$3A(\$5A)	GIFR	General Interrupt	Общий регистр флагов
		Flag Register	прерываний
-----+-----+-----+-----			
\$39(\$59)	TIMSK	Timer/counter Inter-	Регистр маски прерываний от
		rupt mask register	таймера/счетчика
-----+-----+-----+-----			
\$38(\$58)	TIFR	Timer/counter Inter-	Регистр флага прерывания
		rupt Flag register	таймера/счетчика
-----+-----+-----+-----			
\$35(\$55)	MCUCR	MCU general Control	общий регистр управления
		Register	микроконтроллером
-----+-----+-----+-----			
\$34(\$54)	MCUSR	MCU status register	регистр состояния
-----+-----+-----+-----			
\$33(\$53)	TCCR0	Timer/Counter 0	Регистр управления таймером
		Control Register	счетчиком 0
-----+-----+-----+-----			
\$32(\$52)	TCNT0	Timer/Counter 0	Таймер/счетчик 0 (8 бит)
		(8-бит)	
-----+-----+-----+-----			
\$21(\$41)	WDTCR	Watchdog Timer	Регистр управления сторо-
		Control Register	жевым таймером
-----+-----+-----+-----			
\$1E(\$3E)	EEAR	EEPROM Address Register	Регистр адреса энергонеза-
			висимой памяти
-----+-----+-----+-----			
\$1D(\$3D)	EEDR	EEPROM Data Register	Регистр данных энергонеза-
			висимой памяти
-----+-----+-----+-----			

\$1C(\$3C)	EECR	EEPROM Control Register	Регистр управления энерго- независимой памяти
+\$18(\$38)	PORTB	Data Register, Port B	Регистр данных порта В
+\$17(\$37)	DDRB	Data Direction Register Port B	Регистр направления данных порта В
+\$16(\$36)	PINB	Input pins, Port B	Выводы порта В

Примечание: зарезервированные и неиспользуемые ячейки непоказаны

Все устройства ввода/вывода и периферийные устройства AT90S2323 и AT90S2343 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:

Бит	7	6	5	4	3	2	1	0	
	---	T---	T---	T---	T---	T---	T---	T---	
\$3F (\$5F)	I	T	H	S	V	N	Z	C	SREG
	L---	+	+	+	+	+	+	+	
Чт./зап. (R/W)	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 - H: 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 XOR V. Бит S всегда равен исключаящему ИЛИ между флагами N (отрицательный результат) и V (переполнение допол-



нения 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 SPL

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

Бит	7	6	5	4	3	2	1	0
	-----T-----	T-----	T-----	T-----	T-----	T-----	T-----	T-----
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0   SPL
	L-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----
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 AT90S2323 и AT90S2343 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. Список векторов прерываний приведен в таблице 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.

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 TIM_OVF0	; Обработка переполнения таймера 0
\$003 MAIN:	<instr> xxx	; Начало основной программы
— — —	—	—

Таблица 2. Сброс и векторы прерываний.

Номер вектора	Адрес	Источник	Описание прерывания
1	\$000	RESET	Ножка сброса и сброс от сторожевого таймера
2	\$001	INT0	Внешнее прерывание 0
3	\$006	TIMER0 OVF0	Переполнение таймера/счетчика 0

#### ИСТОЧНИКИ СБРОСА

AT90S2323 и AT90S2343 имеют 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 отработки 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.

AT90S2323 имеет программируемое время сброса. При программировании бита FSTR время сброса уменьшается. Микросхемы поставляются с незапрограммированным битом.

Время запуска для AT90S2343 фиксировано.

#### 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 T<sub>por</sub> и времени T<sub>out</sub> (см. таблицы ниже)

Таблица 3. Характеристики сброса. (Vcc=5.0V)

		Min	Typ	Max	
Vpor	Напряжение срабатывания сброса по включению питания	1.1	1.3	1.5	V
Vrst	Напряжение срабатывания сброса по выводу RESET		0.6Vcc		V
Ttout	Задержка на отработку сброса FSTRT незапрограммирован (90S2323)	11	16	21	mS
Ttout	Задержка на отработку сброса FSTRT запрограммирован 90S2323	1.0	1.1	1.2	mS
Ttout	Задержка на отработку сброса для AT90S2343	11	16	21	uS

Таблица 4. Характеристики сброса. (Vcc=3.0V)

		Min	Typ	Max	
Vpor	Напряжение срабатывания сброса по включению питания	1.1	1.3	1.5	V
Vrst	Напряжение срабатывания сброса по выводу RESET		0.6Vcc		V
Ttout	Задержка на отработку сброса FSTRT незапрограммирован (90S2323)	22	32	42	mS
Ttout	Задержка на отработку сброса FSTRT запрограммирован 90S2323	2.0	2.2	2.4	mS
Ttout	Задержка на отработку сброса для AT90S2343	22	32	42	uS

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.

#### РЕГИСТР СОСТОЯНИЯ ПРОЦЕССОРА - MCUSR

Этот регистр содержит информацию о том, что явилось причиной сброса процессора.

Бит	7	6	5	4	3	2	1	0
	---	T---	T---	T---	T---	T---	T---	T---
\$34 (\$54)	-	-	-	-	-	-	EXTRF	PORF   MCUSR
	L---	+	+	+	+	+	+	+
Чт./зап. (R/W)	R	R	R	R	R	R	R/W	R/W
Начальн. знач.	0	0	0	0	0	0	x	x

Биты 7..2 - зарезервированы. В AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются как 0.

Бит 1 - EXTRF - после сброса по включению питания состояние этого бита неопределено. Он устанавливается при внешнем сбросе. Сброс по сторожевому таймеру этот бит не меняет.

Бит 0 - PORF - этот бит устанавливается при сбросе по включению питания, при сбросе от других источников бит не изменяется.

В следующей таблице показаны значения двух бит регистр MCUSR для трех режимов сброса.

Таблица 5. Значение битов PORF и EXTRF после сброса

	---	T---	T---	T---	T---	T---	T---	T---
Источник сброса		PORF		EXTRF				
+-----+								
включение питания		1		неопределен				
+-----+								
внешний сброс		неизменяется		1				
+-----+								
сторожев. таймер		неизменяется		неизменяется				
L-----								

Чтобы определить при помощи этих битов условия сброса, программа пользователя как можно раньше должна сбросить оба бита. Проверка PORF и EXTRF производится до очистки битов. Если эти биты сброшены до того как произойдет сброс от сторожевого таймера или внешний сброс, источник сброса можно определить по следующей таблице:

Таблица 6. Определение источника сброса.

	---	T---	T---	T---	T---	T---	T---	T---
	PORF		EXTRF		Источник сброса			
+-----+								
	0		0		Сторожевой таймер			

+	-----	+	-----	+	-----	+	-----	+
	0		1		Внешний сброс			
+	-----	+	-----	+	-----	+	-----	+
	1		0		Сброс по питанию			
+	-----	+	-----	+	-----	+	-----	+
	1		1		Сброс по питанию			
L	-----	+	-----	+	-----	+	-----	+

#### ОБРАБОТКА ПРЕРЫВАНИЙ

АТ90S2323 и АТ90S2343 имеют 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.

Когда программный счетчик устанавливается на текущий вектор прерывания, флаг, вызвавший обработку этого прерывания аппаратно сбрасывается. Некоторые из флагов прерываний могут сбрасываться записью в них логической единицы.

#### ОБЩИЙ РЕГИСТР МАСКИ ПРЕРЫВАНИЙ GIMSK

Бит	7	6	5	4	3	2	1	0
	---	T---	T---	T---	T---	T---	T---	T---
\$3B (\$5B)	-	INT0	-	-	-	-	-	-
	L---	+	+	+	+	+	+	+
Чт./зап. (R/W)	R	R/W	R	R	R	R	R	R
Начальн. знач.	0	0	0	0	0	0	0	0

Бит 7 - зарезервирован. В АТ90S2323 и АТ90S2343 этот бит зарезервирован и всегда читается как 0.

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 АТ90S2323 и АТ90S2343 эти биты зарезервированы и всегда читаются как 0.

#### ОБЩИЙ РЕГИСТР ФЛАГОВ ПРЕРЫВАНИЙ

Бит	7	6	5	4	3	2	1	0
	-----	T-----	T-----	T-----	T-----	T-----	T-----	T-----
\$3B (\$5B)	-	INTF0	-	-	-	-	-	-
	L-----	+	+	+	+	+	+	+
Чт./зап. (R/W)	R	R/W	R	R	R	R	R	R
Начальн. знач.	0	0	0	0	0	0	0	0

Бит 7 - зарезервирован. В AT90S2323 и AT90S2343 этот бит зарезервирован и всегда читается как 0.

Бит 6 - INTF0: флаг внешнего прерывания 0: При возникновении на выводе INT0 события вызывающего прерывание, INTF0 устанавливается в "1". Если установлены бит I 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 AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются как 0.

#### РЕГИСТР МАСКИ ПРЕРЫВАНИЯ ОТ ТАЙМЕРА/СЧЕТЧИКА - TIMSK

Бит	7	6	5	4	3	2	1	0						
	-----T-----T-----T-----T-----T-----T-----T-----T-----┐													
\$39 (\$59)		-		-		-		-		TOIE0		-		TIMSK
	L-----+-----+-----+-----+-----+-----+-----+-----+-----┘													
Чт./зап.	R		R		R		R		R	W		R		R
Нач.знач.	0		0		0		0		0		0		0	

Биты 7..2 - зарезервированы. В AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются 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) в регистре флагов прерываний (TIFR) таймера/счетчика устанавливается в 1.

Бит 7 - зарезервирован. В AT90S2323 и AT90S2343 этот бит зарезервирован и всегда читается как 0.

#### РЕГИСТР ФЛАГОВ ПРЕРЫВАНИЙ ОТ ТАЙМЕРА/СЧЕТЧИКА - TIFR

Бит	7		6		5		4		3		2		1		0			
	-----T-----T-----T-----T-----T-----T-----T-----T-----┐																	
\$38 (\$58)		-		-		-		-		-		-		TOV0		-		TIFR
	L-----+-----+-----+-----+-----+-----+-----+-----+-----┘																	
Чт./зап.	R		R		R		R		R		R		R/W		R			
Нач.знач.	0		0		0		0		0		0		0		0			

Биты 7..2 - зарезервированы. В AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются 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 очистить 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 7 - зарезервирован. В AT90S2323 и AT90S2343 этот бит зарезервирован и всегда читается как 0.

## ВНЕШНИЕ ПРЕРЫВАНИЯ

Внешние прерывания управляются выводом INT0. Заметим, что прерывания обрабатываются даже когда вывод сконфигурирован как выход. 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 отработку 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.

Бит	7	6	5	4	3	2	1	0
	----	T----	T----	T----	T-----	T-----	T-----	T-----
\$35 (\$55)	-	-   SE	SM	-	-	ISC01	ISC00	MCUCR
	L----	+----	+----	+----	+-----	+-----	+-----	+-----
Чт./зап. (R/W)	R	R	R/W	R/W	R	R	R/W	R/W
Начальн. знач.	0	0	0	0	0	0	0	0

Биты 7..6 - зарезервированы. В AT90S2323 и AT90S2343 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 пониженного 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, - выбирается экономичный режим (Power down). Особенности каждого из ре-

жимов будут рассмотрены ниже.

Биты 3..2 - зарезервированы. В AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются как 0.

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 таблице 7 приведена установка битов для задания срабатывания по уровню и фронтам.

Таблица 7. Управление срабатыванием прерывания 0.

-----Т-----Т-----		
ISC01	ISC00	Описание
0	0	Запрос на прерыв.генерир-ся 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.

#### 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 mS. 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.

#### 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.

#### ТАЙМЕР/СЧЕТЧИК

В AT90S2323 и AT90S2343 предусмотрен 8-разрядный таймер/счетчик общего назначения. Таймер/счетчик получает импульсы от 10-битового предварительного делителя. Таймер/Счетчик можно использовать как таймер, с внутренним источником импульсов или как счетчик, управляемый внешними счетными импульсами.

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 измерения временных интервалов между редкими событиями.

#### РЕГИСТР УПРАВЛЕНИЯ ТАЙМЕРОМ/СЧЕТЧИКОМ 0 - TCCR0

Бит	7	6	5	4	3	2	1	0
	---	T---	T---	T---	T---	T---	T---	T---
\$33	-	-	-	-	-	CS02	CS01	CS00   TCCR0
	L---	+	+	+	+	+	+	+
Чт./зап. (R/W)	R	R	R	R	R	R/W	R/W	R/W
Начальн.знач.	0	0	0	0	0	0	0	0

Биты 7..3 - зарезервированы. В AT90S2323 и AT90S2343 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 8. 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													
L	---	L	---	L	---	L	---	L	-----	L	---	L	---	L	---	L	---	L	-----	L

Условие Stop запрещает/разрешает функционирование таймера/счетчика. В режимах деления частота берется непосредственно от тактовых импульсов. При использовании работы от внешнего вывода предварительно должен быть установлен вывод регистра направления данных (0 - включает ножку на ввод).

#### ТАЙМЕР/СЧЕТЧИК 0 - TCNT0.

Бит	7	6	5	4	3	2	1	0	
	---	T	---	T	---	T	---	T	---
\$32	MSB	-		-		-		-	LSB  TCNT0
	L	---	L	---	L	---	L	---	L
Чт./зап. (R/W)	R/W	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	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 цикле следующем за операцией записи.

#### 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, процессор 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

Бит	7	6	5	4	3	2	1	0
	---	T	---	T	---	T	---	T

\$21(\$41)		-		-		-		WDTOE		WDE		WDP2		WDP1		WDP0		WDTCR
	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Чт./зап. (R/W)		R		R		R		R/W		R/W		R/W		R/W		R/W		R/W
Начальн.знач.		0		0		0		0		0		0		0		0		0

Биты 7..5 - зарезервированы. В AT90S2323 и AT90S2343 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 исполняться 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 таблице 13 приведены различные значения установок предварительного делителя и соответствующие им временные интервалы для напряжения питания Vcc=5V.

Таблица 9. Установки предварительного делителя сторожевого таймера

-----T-----T-----T-----	-----T-----T-----T-----
WDP2 WDP1 WDP0 Период времени	WDP2 WDP1 WDP0 Период времени
+-----+-----+-----+-----+	+-----+-----+-----+-----+
0  0  0  16 mS	1  0  0  256 mS
+-----+-----+-----+-----+	+-----+-----+-----+-----+
0  0  1  32 mS	1  0  1  512 mS
+-----+-----+-----+-----+	+-----+-----+-----+-----+
0  1  0  64 mS	1  1  0  1024 mS
+-----+-----+-----+-----+	+-----+-----+-----+-----+
0  1  1  128 mS	1  1  1  2048 mS
L-----	L-----

## ЧТЕНИЕ И ЗАПИСЬ ЭНЕРГОНЕЗАВИСИМОЙ ПАМЯТИ

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

Время записи лежит в диапазоне 2.5-4 mS 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 машинных цикла до начала выполнения следующей команды.

#### РЕГИСТР АДРЕСА EEPROM - EEAR

Бит	7	6	5	4	3	2	1	0
	-----T-----	T-----	T-----	T-----	T-----	T-----	T-----	T-----
\$1E (\$3E)	-	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0
	L-----	-----	-----	-----	-----	-----	-----	-----
Чт./зап. (R/W)	R	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 - зарезервирован. В AT90S2323 и AT90S2343 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

Бит	7	6	5	4	3	2	1	0
	-----T-----	T-----	T-----	T-----	T-----	T-----	T-----	T-----
\$1D (\$3D)	MSB	-	-	-	-	-	-	LSB
	L-----	-----	-----	-----	-----	-----	-----	-----
Чт./зап. (R/W)	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

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

Бит	7	6	5	4	3	2	1	0
	-----T-----	T-----	T-----	T-----	T-----	T-----	T-----	T-----
\$1C (\$3C)		-	-	-	-	EEMWE	EEWE	EERE
	L-----	-----	-----	-----	-----	-----	-----	-----
Чт./зап. (R/W)	R	R	R	R	R	R/W	R/W	R/W
Начальн.знач.	0	0	0	0	0	0	0	0

Биты 7..3 - зарезервированы. В AT90S2323 и AT90S2343 эти биты зарезервированы и всегда читаются как 0.

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

Бит 1 - EEWЕ - Разрешение записи в EEPROM. Сигнал EEWЕ является стробом записи в EEPROM. После установки правильного адреса и данных для записи в EEPROM необходимо установить бит EEWЕ. При записи "1" в бит EEWЕ должен быть установлен бит EEMWE, тогда происходит запись в EEPROM. Для записи в EEPROM должна соблюдаться следующая последовательность:

1. Ждем обнуления EEWЕ
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 EEWЕ

После того как время записи истечет (типично 2.5 mS для Vcc=5V и 4mS для Vcc=2.7), бит EEWЕ очищается аппаратно. Пользователь может отслеживать этот бит и ожидать его установки в ноль, перед тем как записывать следующий байт. При установке EEWЕ, 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 EEWЕ, если регистры данных или адреса изменяются во время операции записи, запись в ячейку прерывается и результат операции записи становится неопределенным.

## ПОРТ ВВОДА/ВЫВОДА

Порт В является двунаправленным портом ввода/вывода, для AT90S2323 он 3-разрядный, для AT90S2343 - 5-разрядный.

В тексте ниже описание битов 3 и 4 для PORTB, DDRB и PINB не применимо к AT90S2323, для него эти биты доступны только для чтения и всегда ситаются как 0.

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 mA и непосредственно управлять светодиодными индикаторами. Если выводы порта используются как входы и замыкаются на землю, если включены внутренние подтягивающие резисторы, выводы являются источниками тока (Iil). Дополнительные функции выводов порта В приведены в таблице 10.

Таблица 10. Альтернативные функции выводов порта В

Вывод	Альтернативная функция
PB0	MOSI (Вход данных для загрузки памяти)
PB1	MISO (Выход данных для чтения памяти) INT0 (Вход внешнего прерывания 0)
PB2	SCK (Вход тактовых импульсов последовательного обмена) T0 (Тактовый вход таймера/счетчика 0)
PB3	CLOCK (Вход тактирования пр-ра, только AT90S2343)

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

## РЕГИСТР ДАННЫХ ПОРТА В - PORTB

Бит	7	6	5	4	3	2	1	0
\$18 (\$38)	-	-	-	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Чт./зап.	R	R	R	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

## РЕГИСТР НАПРАВЛЕНИЯ ДАННЫХ ПОРТА В - DDRB

Бит	7	6	5	4	3	2	1	0
\$17 (\$37)	-	-	-	DDB4	DDB3	DDB2	DDB1	DDB0
Чт./зап.	R	R	R	R/W	R/W	R/W	R/W	R/W
Нач.знач.	0	0	0	0	0	0	0	0

## ВЫВОДЫ ПОРТА В - PINB

Бит	7	6	5	4	3	2	1	0
\$16 (\$36)	-	-	-	PINB4	PINB3	PINB2	PINB1	PINB0
Чт./зап.	R	R	R	R	R	R	R	R
Нач.знач.	0	0	0	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 В. При чтении PORTB, читаются данные из регистра-защелки, при чтении PINB читаются логические значения присутствующие на выводах порта.

## ПОРТ В, КАК ПОРТ ВВОДА/ВЫВОДА ОБЩЕГО НАЗНАЧЕНИЯ

Все биты порта В 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 17. Влияние DDBn на выводы порта В

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

n = 4,3...0 – номер вывода

## АЛЬТЕРНАТИВНЫЕ ФУНКЦИИ PORTB

CLOCK – PORTB, бит 3 – Вход тактирования процессора (только для AT90S2343). Если запрограммирована бит RCEN и процессор работает со встроенным RC генератором, эта ножка используется для ввода/вывода общего назначения. Если бит RCEN незапрограммирован, на этот вывод должны подаваться тактовые импульсы для процессора.

SCK/T0 – PORTB, бит 2 – В режиме последовательного программирования этот вывод используется как внешний источник тактовых импульсов SCK. При нормальной работе процессора этот вывод может работать как вход для встроенного таймера/счетчика (см. описание таймера/счетчика). Если выбрано внешнее тактирование таймера/счетчика, этот вывод управляет счетчиком даже если сконфигурирован как выход.

MISO/INT0 – PORTB, бит 1 – В режиме последовательного программирования этот вывод работает как выход последовательных данных, MISO. При нормальной работе процессора этот вывод обслуживает внешнее прерывание. (см. описание прерываний). Изменение уровня на этом выводе может управлять прерываниями даже если этот вывод сконфигурирован как выход.

MOSI – PORTB, бит 0 – В режиме последовательного программирования этот вывод работает как вход последовательных данных.

## ПРОГРАММИРОВАНИЕ ПАМЯТИ

### Программирование битов блокировки памяти

Микроконтроллеры AT90S2323 и AT90S2343 имеют два бита блокировки, которые могут быть оставлены незапрограммированными (1) или запрограммироваться (0), при этом достигаются свойства приведенные в таблице 12.

Таблица 12. 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)

AT90S2343 имеет два бита блокировки SPIEN и RCEN, AT90S2323 - только один - SPIEN

\* Если бит SPIEN запрограммирован (0), разрешается последовательная загрузка программ. По умолчанию этот бит запрограммирован (0). Этот бит недоступен при последовательном программировании.

\* Если запрограммирован (0) бит RCEN в качестве источника тактовых сигналов процессора выбирается встроенный RC генератор. По умолчанию этот бит запрограммирован. Если этот бит изменяется при последовательном программировании, изменения вступают в силу при следующем сбросе по питанию.

Эти биты не стираются при стирании памяти микросхемы.

#### Код устройства

Все микроконтроллеры фирмы Atmel имеют 3-байтовый сигнатурный код, по которому идентифицируется устройство. Эти три байта размещены в отдельном адресном пространстве, для AT90S233 и AT90S2343 они имеют следующие значения:

1. \$000: \$1E - код производителя - Atmel
2. \$001: \$91 - 2 кБ флэш памяти
3. \$002: \$03 или \$04 - при \$01=\$91, \$02=03 - м/сх AT90S2343  
при \$01=\$91, \$02=04 - м/сх AT90S2323

Если запрограммированы оба бита блокировки, байты сигнатуры в последовательном режиме не читаются. В этом случае можно использовать специальный режим +12V.

#### Программирование флэш памяти и EEPROM

AT90S2323 и AT90S2343 имеют по 2кБ перепрограммируемой флэш памяти программ и 128 байт энергонезависимой памяти данных.

При поставке AT90S2323 и AT90S2343 флэш память и память данных стерты (содержат \$FF) и готовы к программированию. Эта микросхемы поддерживают низковольтный режим последовательного программирования. Этот



режим предусматривает внутрисистемное программирование микросхем.

Память программ и память данных в AT90S2323 и AT90S2343 программируются байт за байтом в предусмотренных режимах программирования. Для энергонезависимой памяти предусмотрен режим автоматического стирания при последовательном программировании.

Некоторые функции недоступны при последовательном программировании и требуют специального режима с подачей +12В. Напряжение +12В используется только для разрешения программирования, по цепи подачи высокого напряжения процессор практически не потребляет тока.

#### Специальный режим программирования +12В

Этот режим используется для обслуживания функций недоступных при последовательном программировании:

- \* чтение кода устройства при запрограммированных бита блокировки;
- \* программирование стирание бита SPIEN.

Во всех операциях сдвига первым передается старший бит, для тактирования используется вывод CLOCK/XTAL1.

Включение специального режима программирования.

1. Приложить напряжение 4.5–5.5В между выводами VCC и GND.
2. Установить RESET и PB0 в "0".
3. По крайней мере 8 раз переключить уровень на выводе CLOCK/XTAL1 и оставить на выводе "0".
4. Подождать 100нС
5. Подать 12В на вывод RESET и выждать 100нС.

#### Чтение кода устройства

1. Подать положительный импульс на CLOCK.
2. Последовательно подать на PB0 и PB1 следующие числа (старший байт посылается первым, для тактирования сдвига используется CLOCK)  
PB0: 0000 1000  
PB1: 0100 1100
3. Подать 3 положительных импульса на CLOCK/XTAL1
4. Последовательно подать на PB0 и PB1:  
PB0: 0000 0000 (адрес байта сигнатуры 0)  
PB1: 0000 1100
5. Подать три положительных импульса на CLOCK/XTAL1
6. Выдать на PB1 число 0110 1000
7. Подать два положительных импульса на CLOCK/XTAL1
8. Прочитать байт кода устройства с вывода PB2
9. Подать 4 положительных импульса на CLOCK/XTAL1
10. Выдать на PB1 0110 1100
11. Повторить шаги 3–10 используя адрес 0000 0001 и 0000 0010 для двух следующих байт кода устройства.
12. Подать положительный импульс на CLOCK/XTAL1

#### Программирование битов конфигурации

1. Подать положительный импульс на CLOCK.
2. Последовательно подать на PB0 и PB1 следующие числа  
PB0: 0100 0000  
PB1: 0100 1100
3. Подать 3 положительных импульса на CLOCK/XTAL1
4. Последовательно подать на PB0 и PB1:

- PB0: 00S0 000R (S/R=1 биты стерты, S/R=0 - запрограммированы)  
PB1: 0010 1100
5. Подать три положительных импульса на CLOCK/XTAL1
  6. Выдать на PB1 число 0110 1000
  7. Подать два положительных импульса на CLOCK/XTAL1
  8. Подождать 1 мс
  9. Подать положительный импульс на CLOCK/XTAL1
  10. Выдать на PB1 0110 1100
  11. Подать два положительных импульса на CLOCK/XTAL1

#### Чтение битов конфигурации и блокировки

1. Подать положительный импульс на CLOCK.
2. Последовательно подать на PB0 и PB1 следующие числа  
PB0: 0000 1000  
PB1: 0100 1100
3. Подать 3 положительных импульса на CLOCK/XTAL1
4. Выдать на PB1 число 0111 1000
5. Подать два положительных импульса на CLOCK/XTAL1
6. Прочитать байт данных с вывода PB2. Значение прочитанных бит следующее:  
бит 7 - бит блокировки 1 (0 - запрограммирован)  
бит 6 - бит блокировки 2 (0 - запрограммирован)  
бит 5 - бит SPIEN (0 - запрограммирован)  
бит 0 - бит RCEN (0 - запрограммирован)
7. Подать 4 положительных импульса на CLOCK/XTAL1
8. Выдать на PB1 0111 1100
9. Подать два положительных импульса на CLOCK/XTAL1

#### ПОСЛЕДОВАТЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Как память программ, так и память данных могут быть запрограммированы 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 перед программированием/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 можно подавать внешнюю тактовую частоту на вывод CLOCK/XTAL1 или использовать внутренний RC генератор. Минимальные длительности низкого и высокого уровня сигнала SCK определены следующим образом:

Низкий: > 2 периодов XTAL1

Высокий: > 2 периодов XTAL1

Для программирования и проверки процессора в режиме последовательного программирования рекомендуется следующая последовательность

действий (см. формат четырех байтовой команды в табл.13):

1. Последовательность включения питания:  
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, после того как на SCK будет установлен '0', на RESET необходимо подать положительный импульс. При внешнем тактировании подайте частоту от 0 до 8 MHz на вывод CLOCK/XTAL1. Если используется встроенный генератор, внешние импульсы не нужны).
2. Подождать 20 mS и разрешить последовательное программирование послав команду разрешения на вывод MOSI/PB0. (минимальные длительности уровней сигнала на выводе SCK были приведены выше).
3. При подаче команды стирания (всегда при программировании памяти программ), подождать 10 mS, выдать по ложительный импульс на RESET и повторить шаг 2.
4. 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. Следующий байт можно программировать через 4mS.
5. Любую ячейку памяти можно проверить используя команду чтения, которая выдает содержимое указанной ячейки на последовательный вывод MISO/PB1.
6. По окончании программирования вывод RESET может быть установлен в '1' для возобновления нормальной работы схемы.
6. Последовательность выключения питания (если необходимо)  
Установить CLOCK/XTAL1 в '0'  
Установить RESET в '1'.  
Отключить питание

Таблица 25. Команды последовательного программирования

Команда	Формат команды				Действие
	Байт 1	Байт 2	Байт 3	Байт 4	
Разрешение программир.	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Разрешен.послед. программирования после установки RESET в '0'
Стирание микросх.	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Стирание памяти прогр. и данных
Чтение памяти прогр.	0010 H000	0000 00aa	bbbb bbbb	oooo oooo	Читать старший или младший байт о по адресу ab
Запись памяти прогр.	0100 H000	0000 00aa	bbbb bbbb	iiii iiii	Запись старш. или младш. байта i в память прогр по адресу ab
Чтение памяти данных	1010 0000	0000 0000	xbbb bbbb	oooo oooo	Читать данные о по адресу b

Запись памяти данных	1100 0000	0000 0000	xbbb bbbb	iiii iiii	Запись данных i по адресу b
Чтение бит блокировки и конфигу.	0101 1000	xxxx xxxx	xxxx xxxx	12Sx xxxR	Чтение битов. 0 - запрограмир. 1 - незапрогр.
Запись битов блокир.	1010 1100	111x x21x	xxxx xxxx	xxxx xxxx	Запись битов блокировки. Установить 1,2=0 для блокировки памяти
Запись бита RCEN	1010 1100	101x xxxR	xxxx xxxx	xxxx xxxx	Запись RCEN
Читать код устройства	0011 0000	xxxx xxxx	xxxx xbbb	oooo oooo	Читать код устройства о с адреса b

L-----

а - старшие биты адреса; b - младшие биты адреса; о - выход данных  
Н = '0'-младший байт, '1'-старший байт; i - вход данных;  
х - не используется; 1 - 1-й бит блокировки; 2 - 2-й бит блокировки  
R - бит RCEN; S - бит SPIEN

Код устройства не читается если записаны оба бита блокировки.

При последовательном программировании бита RCEN изменения вступают в силу после отключения/включения питания.

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

#### Максимально допустимые параметры

Рабочая температура	-55°C - +125°C
Температура хранения	-65°C - +150°C
Напряжение на любом выводе кроме RESET	-1.0V - +7.0V
Максимальное рабочее напряжение	6.6V
Постоянный ток через вывод порта	40.0 mA
Постоянный ток между VCC и GND	140.0 mA

L-----

ПРИМЕЧАНИЕ: выход параметров за пределы указанные в таблице может привести к нарушению работоспособности микросхемы. Это предельные значения параметров, рабочие параметры микросхемы приведены ниже. Удержание предельных значений на выводах м/сх в течение длительного времени может привести к потере работоспособности м/сх.

#### ХАРАКТЕРИСТИКИ ПО ПОСТОЯННОМУ ТОКУ

Та = -40...+85°C, Vcc=2.7...6.0V (если не указано иначе)

Мин./Тип./Макс.

Входное напряжение '0' - -0.5/\_/0.3Vcc V

Входное напряжение '0' на выв.XTAL - -0.5/\_/0.1Vcc V

Входное напряжение '1' (кроме XTAL1 и RESET) -  $0.6V_{CC}/\_\_/V_{CC}+0.5\text{ V}$   
на XTAL1 -  $0.7V_{CC}/\_\_/V_{CC}+0.5\text{ V}$   
на RESET -  $0.85V_{CC}/\_\_/V_{CC}+0.5\text{ V}$   
Выходное напряжение '0' (Порт В) I1=20mA, Vcc=5V -  $\_\_/0.6\text{ V}$   
I1=10mA, Vcc=3V -  $\_\_/0.5\text{ V}$   
Выходное напряжение '1' (Порт В) I0h=3mA, Vcc=5V -  $4.3/\_\_/V$   
I0h=1.5mA, Vcc=3V -  $2.3/\_\_/V$   
Ток утечки по входам Vcc=6V -  $-8/\_\_/8\text{ uA}$   
Подтягивающий резистор сброса -  $100/\_\_/500\text{ kOm}$   
Подтягивающий резистор вывода порта 35/ $\_\_/120\text{ kOm}$   
Потребляемый ток:  
Активный режим, 3V, 4MHz -  $\_\_/3\text{ mA}$   
холостой ход (idle mode), 3V, 4MHz -  $\_\_/1.0/1.2\text{ mA}$   
пониженное потребление WDT включен, 3V -  $\_\_/9/15\text{ uA}$   
WDT выключен, 3V -  $\_\_/1/2\text{ uA}$

#### ПРИМЕЧАНИЯ:

- В рабочем состоянии ток через выводы должен ограничиваться следующими условиями:
  - Максимальный ток через вывод - 20 mA
  - Максимальный ток через все выводы - 80mA
При превышении тока напряжение "0" может увеличиться.  
Увеличение тока выше приведенных пределов не гарантируется.
- Минимальное напряжение для режима пониженного потребления - 2V

#### ПАРАМЕТРЫ ВНЕШНЕГО ТАКТОВОГО СИГНАЛА

Параметр	AT90LS2323/43		AT90S2323/43		Ед.
	min	max	min	max	
Частота	0	4	0	10	MHz
Период	250		100		nS
Длительность '1'	0		0		nS
Длительность '0'	0		0		nS
Длительность фронта		1.6		0.5	uS

#### СПИСОК РЕГИСТРОВ

Address	Name	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			3	A	P	E	3	E	P
\$3D \$5D	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
\$3C \$5C			3	A	P	E	3	E	P
\$3B \$5B	GIMSK	-	INT0	-	-	-	-	-	-
\$3A \$5A	GIFR	-	INTF0	-	-	-	-	-	-
\$39 \$59	TIMSK	-	-	-	-	-	-	TOIE0	-
\$38 \$58	TIFR	-	-	-	-	-	-	TOV0	-

\$37	\$57			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$36	\$56			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$35	\$55	MCUCR	-	-		SE		SM		-		-		ISC01		ISC00		
\$34	\$54	MCUSR	-	-		-		-		-		-		EXTRF		PORF		
\$33	\$53	TCCR0	-	-		-		-		-		-		CS02	CS01	CS00		
\$32	\$52	TCNT0		Таймер / счетчик (8 бит)														
\$31	\$51			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$30	\$50			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
...	...			-														
\$23	\$43			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$22	\$42			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$21	\$41	WDTCSR	-	-		-		WDTOE	WDE		WDP2		WDP1		WDP0			
\$20	\$40			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$1F	\$3F			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$1E	\$3E	EEAR		Регистр адреса энергонезависимой памяти														
\$1D	\$3D	EEDR		Регистр данных энергонезависимой памяти														
\$1C	\$3C	EECR	-	-		-		-		-		EEMWE	EEWE		EERE			
\$1B	\$3B			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$1A	\$3A			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$19	\$39			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$18	\$38	PORTB	-	-		-		PORTB4	PORTB3	PORTB2	PORTB1		PORTB0					
\$17	\$37	DDRB	-	-		-		DDB4	DDB3	DDB2	DDB1		DDB0					
\$16	\$36	PINB	-	-		-		PINB4	PINB3	PINB2	PINB1		PINB0					
\$14	\$34			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$13	\$33			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
...	...			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	
\$00	\$20			З	А	Р	Е	З	Е	Р	В	И	Р	О	В	А	Н	

L-----+-----+-----+-----+-----+-----+-----+-----+-----+-----

### Набор команд

Мнемоника	Операнды	Описание	Действие	Флаги	Цкл
<b>АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ КОМАНДЫ</b>					
ADD	Rd, Rr	Сложить два регистра	Rd<Rd+Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Сложить с переносом	Rd<Rd+Rr+C	Z,C,N,V,H	1
ADIW	Rdl, K	Сложить слово с конст	Rdh, l<Rdh, l+K	Z,C,N,V,S	2
SUB	Rd, Rr	Вычесть два регистра	Rd<Rd-Rr	Z,C,N,V,H	1
SUBI	Rd, K	Вычесть константу	Rd<Rd-K	Z,C,N,V,H	1
SBIW	Rdl, K	Вычесть слово с конст	Rdh, l<Rdh, l-K	Z,C,N,V,S	2
SBC	Rd, Rr	Вычесть с переносом	Rd<Rd-Rr-C	Z,C,N,V,H	1
SBCI	Rd, K	Вычесть с переносом	Rd<Rd-K-C	Z,C,N,V,H	1
AND	Rd, Rr	Логическое И	Rd<Rd AND Rr	Z,N,V	1
ANDI	Rd, K	Логическое И	Rd<Rd AND K	Z,N,V	1
OR	Rd, Rr	Логическое ИЛИ	Rd<Rd OR Rr	Z,N,V	1
ORI	Rd, K	Логическое ИЛИ	Rd<Rd OR K	Z,N,V	1
EOR	Rd, Rr	Исключающее ИЛИ	Rd<Rd XOR Rr	Z,N,V	1
COM	Rd	Дополнение до 1	Rd<\$FF - Rd	Z,C,N,V	1
NEG	Rd	Дополнение до 2	Rd<\$00 - Rd	Z,C,N,V,H	1
SBR	Rd, K	Установ. бит(ы) в рег.	Rd<Rd OR K	Z,N,V	1
CBR	Rd, K	Сброс. бит(ы) в рег.	Rd<RdAND (FFh-K)	Z,N,V	1
INC	Rd	Увеличить на 1	Rd<Rd+1	Z,N,V	1
DEC	Rd	Уменьшить на 1	Rd<Rd-1	Z,N,V	1
TST	Rd	Проверить на 0 или -	Rd<Rd AND Rd	Z,N,V	1

CLR	Rd	Очистить регистр	Rd<Rd XOR Rd	Z,N,V	1
SER	Rd	Установить регистр	Rd<\$FF	None	1
КОМАНДЫ ВЕТВЛЕНИЯ					
RJMP	k	Относительный переход	PC<PC+k+1	None	2
LJMP		Переход по адресу (Z)	PC<Z	None	2
RCALL	k	Относ. вызов подпрогр.	PC<PC+k+1	None	3
ICALL		Вызов подпр по адр (Z)	PC<Z	None	3
RET		Выход из подпрограммы	PC<STACK	None	4
RETI		Выход из прерывания	PC<STACK	I	4
CPSE	Rd,Rr	Сравнить, пропуск если равно	if (Rd=Rr) PC<PC+2 или 3	None	1/2
CP	Rd,Rr	Сравнить	Rd-Rr	Z,N,V,C,H	1
CPC	Rd,Rr	Сравнить с переносом	Rd-Rr-C	Z,N,V,C,H	1
CPI	Rd,K	Сравнить с константой	Rd-K	Z,N,V,C,H	1
SBRC	Rr,b	Пропуск если бит в регистре сброшен	if (Rr(b)=0) PC<PC+2 или 3	None	1/2
SBRB	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 рег ввода/выв. сброшен	if (P(b)=0) PC<PC+2 или 3	None	1/2
SBIS	P, b	Пропуск если бит в рег ввода/выв. установлен	if (P(b)=1) PC<PC+2 или 3	None	1/2
BRBS	s, k	Переход если установл. 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 неравно	if (Z=0) PC<PC+k+1	None	1/2
Мнемоника	Операнды	Описание	Действие	Флаги	Цкл
BRCS	k	Переход если установл. 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 V=1) PC<PC+k+1	None	1/2
BRHS	k	Переход если установл. 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 установл. 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 установл. 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 рег.	Rd<Rr	None	1
LDI	Rd,K	Загрузить константу	Rd<K	None	1
LD	Rd,X	Загрузить регистр непосредственно	Rd<(X)	None	2
LD	Rd,X+	Загрузить регистр непосредств. с пост инкрем.	Rd<(X), X<X+1	None	2
LD	Rd,-X	Загрузить регистр непосредств. с предв. декрем.	X<X-1, Rd<(X)	None	2
LD	Rd,Y	Загрузить регистр непосредственно	Rd<(Y)	None	2
LD	Rd,Y+	Загрузить регистр непосредств. с пост инкрем.	Rd<(Y), Y<Y+1	None	2
LD	Rd,-Y	Загрузить регистр непосредств. с предв. декрем.	Y<Y-1, Rd<(Y)	None	2
LDD	Rd,Y+q	Загрузить регистр непосредств. со смещением	Rd<(Y+q)	None	2
LD	Rd,Z	Загрузить регистр непосредственно	Rd<(Z)	None	2
LD	Rd,Z+	Загрузить регистр непосредств. с пост инкрем.	Rd<(Z), Z<Z+1	None	2
LD	Rd,-Z	Загрузить регистр непосредств. с предв. декрем.	Z<Z-1, Rd<(Z)	None	2
LDD	Rd,Z+q	Загрузить регистр непосредств. со смещением	Rd<(Z+q)	None	2
LDS	Rd,k	Загрузить из ОЗУ	Rd<(k)	None	3
ST	X,Rr	Записать регистр непосредственно	(X)<Rr	None	2
ST	X+,Rr	Записать регистр непосредств. с пост инкр.	(X)<Rr, X<X+1	None	2
ST	-X,Rr	Записать регистр непосредств. с пред. декрем.	X<X-1, (X)<Rr	None	2
ST	Y,Rr	Записать регистр непосредственно	(Y)<Rr	None	2
ST	Y+,Rr	Записать регистр непосредств. с пост инкр.	(Y)<Rr, Y<Y+1	None	2
ST	-Y,Rr	Записать регистр непосредств. с пред. декрем.	Y<Y-1, (Y)<Rr	None	2
STD	Y+q,Rr	Запис. рег. непосредств. со смещением	(Y+q)<Rr	None	2
ST	Z,Rr	Записать регистр не-			



		посредственно	$(Z) < Rr$	None	2
ST	Z+, Rr	Записать регистр не- поср.с пост инкр.	$(Z) < Rr, Z < Z+1$	None	2
ST	-Z, Rr	Записать регистр не- поср.с пред.декрем.	$Z < Z-1, (Z) < Rr$	None	2
STD	Y+q, Rr	Запис.рег.непоср.со смещением	$(Y+q) < Rr$	None	2
STS	k, Rr	Записать в ОЗУ	$(k) < Rr$	None	3
LPM		Загр.из памяти прогр.	$R0 < (Z)$	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 рег. ввода/вывода	$I/O(P, b) < 1$	None	2
CBI	P, b	Сбросить бит в рег. 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	Логич.сдвиг вправо	$Rd(n) < Rd(n+1),$ $Rd(7) < 0$	Z, C, N, V	1
ROL	Rd	сдвиг влево через C	$Rd(0) < C,$ $Rd(n+1) < Rd(n),$ $C < Rd(7)$	Z, C, N, V	1
ROR	Rd	сдвиг вправо через C	$Rd(7) < C,$ $Rd(n) < Rd(n+1),$ $C < Rd(0)$	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 нибблов	$Rd(3-0) < Rd(7-4)$ $Rd(7-4) < Rd(3-0)$	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	$C < 0$	C	1
SEN		Установить флаг N	$N < 1$	N	1
CLN		Сбросить флаг N	$N < 0$	N	1

Мнемоника	Операнды	Описание	Действие	Флаги	Цкл
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 тай- мера		None	1

---