Микроконтроллеры AT90S2333 и AT90S4433 фирмы Atmel

AT90S2333 и AT90S4433 - экономичные 8-битовые КМОП микроконтрол­леры, построенные с использованием расширенной RISC архитектуры AVR. Исполняя по одной команде за период тактовой частоты, AT90S2333 и AT90S4433 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. AT90S2333/4433 предлагают следующие возможности: 2кБ/4кБ загружа­емой флэш памяти; 128/256 байт EEPROM; 128 байт статического ОЗУ, 20 линий ввода/вывода общего назначения; 32 рабочих регистра; настраивае­мые таймеры/счетчики с режимом совпадения; внешние и внутренние преры­вания; программируемый универсальный последовательный порт; 6-каналь­ный 10-разрядный АЦП; 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, AT90S2333/4433 являются мощными микроконтроллерами, которые позволяют создавать достаточно гибкие и эффективные по стоимости устройства. AT90S2333/4433 поддерживаются полной системой разработки включаю­щей в себя компиляторы Си, макроассемблеры, программные отладчики/си­муляторы, внутрисхемные эмуляторы и отладочные комплекты.

|  |  |  |
| --- | --- | --- |
| назначение вывода | номер вывода | номер вывода PDIP |
| RESET | 29 | 1 |
| PD0/RXD | 30 | 2 |
| PD1/TXD | 31 | 3 |
| PD2/INT0 | 32 | 4 |
| PD3/INT1 | 1 | 5 |
| PD4/T0 | 2 | 6 |
| VCC | 4 | 7 |
| GND | 5 | 8 |
| XTAL1 | 7 | 9 |
| XTAL2 | 8 | 10 |
| PD5/T1 | 9 | 11 |
| PD6/AIN0 | 10 | 12 |
| PD7/AIN1 | 11 | 13 |
| PB0/ICP | 12 | 14 |
| PB1/OC1 | 13 | 15 |
| PB2/SS | 14 | 16 |
| PB3/MOSI | 15 | 17 |
| PB4/MISO | 16 | 18 |
| PB5/SCK | 17 | 19 |
| AVCC | 18 | 20 |
| AREF | 20 | 21 |
| AGND | 21 | 22 |
| PC0/ADC0 | 23 | 23 |
| PC1/ADC1 | 24 | 24 |
| PC2/ADC2 | 25 | 25 |
| PC3/ADC3 | 26 | 26 |
| PC4/ADC4 | 27 | 27 |
| PC5/ADC5 | 28 | 28 |

ОПИСАНИЕ ВЫВОДОВ

GND - земля

Port B (PB5..PB0) - Порт B является 6-битовым двунаправленным портом ввода/вывода с внутренними подтягивающими резисторами. Выходные буферы порта B могут поглощать ток до 20мА. Если выводы PB0..PB5 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.

Port С (PС5..PС0) - Порт С является 6-битовым двунаправленным портом ввода/вывода с внутренними подтягивающими резисторами. Выходные буферы порта С могут поглощать ток до 20мА. Если выводы PС0..PС5 испо­льзуются как входы и извне устанавливаются в низкое состояние, они яв­ляются источниками тока, если включены внутренние подтягивающие резис­торы. Кроме того Порт С обслуживает аналоговые входы АЦП.

Port D (PD5..PD0) - Порт D является 8-битовым двунаправленным портом ввода/вывода с внутренними подтягивающими резисторами. Выходные буферы порта B могут поглощать ток до 20мА. Если выводы PD0..PD7 испо­льзуются как входы и извне устанавливаются в низкое состояние, они яв­ляются источниками тока, если включены внутренние 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 ус-

тройство.

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

XTAL2 - Выход инвертирующего усилителя генератора.

AVCC - Вывод источника питания АЦП. Этот вывод через фильтр низ­кой частоты должен быть подключен к выводу питания процессора.

AREF - Вход опорного напряжения АЦП. Напряжение, подаваемое на этот вывод лежит в пределах 2.7В...AVCC.

AGND - Если плата имеет отдельный слой аналоговой земли, к нему подключается этот вывод. В противном случае этот вывод соединяется с GND.

## 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. Если сигнал ге­нератора необходимо использовать для управления внешними устройствами, сигнал с вывода XTAL2 снимается через одиночный буфер серии HC, при этом емкость конденсатора с вывода на землю уменьшается на 5pF. При подаче внешнего тактового сигнала вывод XTAL2 остается неподключенным, а XTAL1 подключается в выходу внешнего генератора.

**Обзор архитектуры процессоров.**

Регистровый файл быстрого доступа содержит 32 8-разрядных регист­ра общего назначения, доступ к которым осуществляется за один машинный цикл. Поэтому за один машинный цикл исполняется 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 c 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 00h-1Fh в области данных, обращаться к ним можно как к ячейкам памяти.

Пространство ввода/вывода 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 (20h- 5Fh).

Процессоры AVR построены по гарвардской архитектуре с раздельными областями памяти программ и данных. Доступ к памяти программ осуществ­ляется при помощи одноуровнего 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 статическому 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:

|  |  |
| --- | --- |
| **Регистр X** | 15 0 |
| 7 0 | 7 0 |
| **1Bh (R27)** | **1Ah (R26)** |

|  |  |
| --- | --- |
| **Регистр Y** | 15 0 |
| 7 0 | 7 0 |
| **1Dh (R29)** | **1Ch (R28)** |

|  |  |
| --- | --- |
| **Регистр Z** | 15 0 |
| 7 0 | 7 0 |
| **1Fh (R31)** | **1Eh (R30)** |

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.**

AT90S2333/4433 содержат 2/4 кБ загружаемой флэш памяти для хране­ния программ. Поскольку все команды занимают одно 16- или 32-разрядное слово, флэш память организована как 1/2 Kx16. Флэш-память выдерживает не менее 1000 циклов перезаписи. Программный счетчик имеет ширину 10/11 бит и позволяет адресоваться к 1024/2048 словам программной флэш-памяти.

Подробно загрузка флэш памяти будет рассмотрена дальше.

###### EEPROM память данных

AT90S2333/4433 содержат 128/256 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 приведенном ниже показана организация памяти данных в AT90S2333/4433.

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. Регисты 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.

**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.

|  |  |
| --- | --- |
| Регистровый файл | Область адресов данных |
| R0 | 00h |
| R1 | 01h |
| : | : |
| R30 | 1E |
| R31 | 1F |
| Регистры ввода\вывывода |  |
| 00h | 20h |
| 01h | 21h |
| : | : |
| 3Eh | 5Eh |
| 3Fh | 5Fh |
| - | Встроенное ОЗУ |
| - | 61h |
| - | : |
| - | DEh |
| - | DFh |

 **Пространство ввода/вывода AT90S2333/4433**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адреса** | **регистры** | **название** | **функции**  |
| 3Fh(5Fh) | SREG | Status REGister | Регистр Состояния |
| 3Dh(5Dh) | SP | Stack pointer low | Указатель стека |
| 3Bh(5Bh) | GIMSK | General Interrupt MaSK register | Общий регистр маски прерываний |
| 3Ah(5Ah) | GIFR | General Interrupt Flag Register | Общий регистр флагов прерываний |
| 39h(59h) | TIMSK | Timer/counter Inter­rupt mask register | Регистр маски прерываний от таймера/счетчика |
| 38h(58h) | TIFR | Timer/counter Inter­rupt Flag register | Регистр флага прерывания таймера/счетчика |
| 35h(55h) | MCUCR | MCU general Control Register | общий регистр управления микроконтроллером |
| 34h(54h) | MCUSR | MCU Status Register | рег.состояния микроконтрол. |
| 33h(53h) | TCCR0 | Timer/Counter 0 Control Register | Регистр управления таймером счетчиком 0 |
| 32h(52h) | TCNT0 | Timer/Counter 0 (8-бит) | Таймер/счетчик 0 (8 бит) |
| 2Fh(4Fh) | TCCR1A | Timer/Counter 1 Control Register A | Рег. A управления таймером счетчиком 1 |
| 2Eh(4Eh) | TCCR1B | Timer/Counter 1 Control Register B | Рег. B управления таймером счетчиком 1 |
| 2Dh(4Dh) | TCNT1H | Timer/Counter 1 High byte | Таймер/счетчик 1 старший байт |
| 2Ch(4Ch) | TCNT1L | Timer/Counter 1 Low byte | Таймер/счетчик 1 младший байт |
| 2Bh(4Bh) | OCR1H | Output Compare Register 1 high byte | Выход регистра совпаден. 1 старший байт |
| 2Ah(4Ah) | OCR1L | Output Compare Register 1 low byte | Выход регистра совпаден. 1 младший байт |
| 27h(47h) | ICR1H | T/C 1 Input Cupture Re­gister High Byte | Регистр захвата Т\С 1 старший байт |
| 26h(46h) | ICR1L | T/C 1 Input Cupture Re­gister Low Byte | Регистр захвата Т\С 1 младший байт |
| 21h(41h) | WDTCR | Watchdog Timer Control Register | Регистр управления сторо­жевым таймером |
| 1Eh(3Eh) | EEAR | EEPROM Address Register | Регистр адреса энергонеза­висимой памяти |
| 1Dh(3Dh) | EEDR | EEPROM Data Register | Регистр данных энергонеза­висимой памяти |
| 1Ch(3Ch) | EECR | EEPROM Control Register | Регистр управления энерго­независимой памяти |
| 18h(38h) | PORTB | Data Register, Port B | Регистр данных порта B |
| 17h(37h) | DDRB | Data Direction Register Port B | Регистр направления данных порта B |
| 16h(36h) | PINB | Input pins, Port B | Выводы порта B |
| 15h(35h) | PORTС | Data Register, Port С | Регистр данных порта С |
| 14h(34h) | DDRС | Data Direction Register Port С | Регистр направления данных порта С |
| 13h(33h) | PINС | Input pins, Port С | Выводы порта С |
| 12h(32h) | PORTD | Data Register, Port D | Регистр данных порта D |
| 11h(31h) | DDRD | Data Direction Register Port D | Регистр направления данных порта D |
| 10h(30h) | PIND | Input pins, Port D | Выводы порта D |
| 0Fh(2Fh) | SPDR | SPI I/O Data Register | Регистр данных порта SPI |
| 0Eh(2Eh) | SPSR | SPI Status Register | Регистр состоян. порта SPI |
| 0Dh(2Dh) | SPCR | SPI Control Register | Регистр управл.порта SPI |
| 0Ch(2Ch) | UDR | UART Data Register | Регистр данных последова­тельного порта |
| 0Bh(2Bh) | USR | UART Status Register | Регистр состояния последо­вательного порта |
| 0Ah(2Ah) | UCR | UART Control Register | Регистр управления последо­вательного порта |
| 09h(29h) | UBRR | UART Baud Rate Register | Регистр скорости последо­вательного порта |
| 08h(28h) | ACSR | Analog Comparator Cont­rol and Status Register | Регистр управления и состо­яния аналогового компарат. |
| 07h(27h) | ADMUX | ADC multiplexer Select register | Регистр коммутатора АЦП |
| 06h(26h) | ADCSR | ADC Control and Status Register | Регистр управления и состо­яния АЦП |
| 05h(25h) | ADCH | ADC data register High | Рег данных АЦП (старш.) |
| 04h(24h) | ADCL | ADC data register Low | Рег данных АЦП (младш.) |
| 03h(23h) | UBRRHI | UART Baud Rate Register HIgh | Регистр скорости последо­вательного порта (старш.) |

Примечание: зарезервированные и неиспользуемые ячейки не показаны

Все устройства ввода/вывода и периферийные устройства процессора 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 00h..1Fh 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. Для совместимости с другими устройствами при доступе к зарезерви­рованным битам в них должен записываться ноль, зарезервированные адре­са в пространстве ввода/вывода не должны записываться

Регистр состояния – SREG 3Fh(5Fh)

Регистр состояния расположен по адресу 3Fh (5Fh) пространства ввода/вывода и определен следующим образом:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **3Fh(5Fh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **I** | **T** | **H** | **S** | **V** | **N** | **Z** | **S** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | 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 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 3Dh (5Dh) хранит указатель стека процессора. 8-ми разрядов достаточно, для адресации ОЗУ в преде­лах 60h -DFh.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **3Dh(5Dh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **SP7** | **SP6** | **SP5** | **SP4** | **SP3** | **SP2** | **SP1** | **SP0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Указатель стека указывает на область памяти в которой расположен стек вызова подпрограмм и прерваний. 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).

\*\*\*Стек процессора работает с предварительным инкрементом и постдекрементом

**Сброс и обработка прерываний.**

В процессоре предусмотрены 13 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 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:

|  |  |  |
| --- | --- | --- |
| **000h** | rjmp RESET | Обработка сброса |
| **001h** | rjmp EXT\_INT0 | **Обработка IRQ0** |
| **002h** | rjmp EXT\_INT1 | **Обработка IRQ1** |
| **003h** | rjmp TIM1\_CAPT | **Обработка захвата таймера 1** |
| **004h** | rjmp TIM1\_COMP | **Обработка совпадения таймера 1** |
| **005h** | rjmp TIM1\_OVF | **Обработка переполнения таймера 1** |
| **006h** | rjmp TIM0\_OVF | **Обработка переполнения таймера 0** |
| **007h** | rjmp SPI\_STC | **Обработка передачи по SPI** |
| **008h** | rjmp UART\_RXC | **Обработка приема байта** |
| **009h** | rjmp UART\_DRE | **Обработка освобождения UDR** |
| **00Ah** | rjmp UART\_TXC | **Обработка передачи байта** |
| **00Bh** | rjmp ADC | **Обработка преобразования АЦП** |
| **00Ch** | rjmp EE\_RDY | **Обработка готовности EEPROM** |
| **00Dh** | rjmp ANA\_COMP | **Обработка аналогов. компаратора** |
| **00Eh** | Основная программа | **Начало основной программы** |

**Сброс и векторы прерываний.**

|  |  |  |  |
| --- | --- | --- | --- |
| Номер вектора | Адрес | Источник | Описание прерывания |
| 1 | 000h | RESET | Ножка сброса, сторожевой таймер Brown-Out reset |
| 2 | 001h | INT0 | Внешнее прерывание 0 |
| 3 | 002h | INT1 | Внешнее прерывание 1 |
| 4 | 003h | TIMER1 CAPT | Захват таймера/счетчика 1 |
| 5 | 004h | TIMER1 COMP | Совпаден. таймера/счетчика 1 |
| 6 | 005h | TIMER1 OVF | Переполнение таймера/счетчика 1 |
| 7 | 006h | TIMER0 OVF | Переполнение таймера/счетчика 0 |
| 8 | 007h | SPI, STC | Передача по SPI завершена |
| 9 | 008h | UART RX | Последоват.порт прием закончен |
| 10 | 009h | UART UDRE | Посл.порт регистр данных пуст |
| 11 | 00Ah | UART TX | Посл.порт передача закончена |
| 12 | 00Bh | ADC | Преобразование АЦП завершено |
| 13 | 00Ch | RDY | EEPROM готово |
| 14 | 00Dh | COMP | Аналоговый компаратор |

ИСТОЧНИКИ СБРОСА

AT90S2333/4433 имеют четыре 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 оконча­нию времени отработки сторожевого таймера, если разрешена его работа.

\* Brown-Out сброс сброс при падении Vcc ниже некоторого значения.

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

Импульс сброса по включению питания генерируется внутренней схе­мой. Уровень срабатывания схемы - 2.2В. Сброс производится когда на­пряжение питания превысит уровень срабатывания. Схема сброса по вклю­чению питания не дает процессору запускаться до тех пор, пока напряже­ние не достигнет безопасного уровня. При достижении безопасного уровня напряжения включается счетчик задержки определяющий длительность сбро­са. Эта длительность задается битами-перемычками и может устанавли­ваться в одно из восьми значений приведенных в таблице 4.

**Таблица 3. Хар актеристики сброса.(Vcc=5.0V)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип напряжения |  | Min | Typ | Max |
| **V**power | Напряжение срабатывания сброса по включению питания | 1.7v | 2.2v | 2.7v |
| **V**reset | Напряжение срабатывания сброса по выводу RESET |  | 0.6Vcc |  |
| **V**bodlevel | Напряжение срабатывания сброса по Brown-Out | BODLEVEL=1 | 2.6v | 2.7v | 2.8v |
| Напряжение срабатывания сброса по Brown-Out | BODLEVEL=0 | 3.8v | 4.0v | 4.2v |

**Таблица 4. Установка времени сброса**

|  |  |
| --- | --- |
| **CKSEL [2:0]** | **Время запуска** |
| **000** | **4mS + 6CK** |
| **001** | **6CK** |
| **010** | **64mS + 16K CK** |
| **011** | **4mS + 16K CK** |
| **100** | **16K CK** |
| **101** | **64mS + 1K CK** |
| **110** | **4mS + 1K CK** |
| **111** | **1K CK** |

# 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.

# BROWN-OUT

AT90S2333/4433 имеют встроенную схему отслеживания напряжения пи­тания. Работа этой схемы разрешается и запрещается битом-перемычкой BODEN. Если бит BODEN запрограммирован, при уменьшении напряжения ниже заданного уровня срабатывает схема сброса. Время сброса задается как и для сброса по включению питания (табл.4). Уровень сброса устанавлива­ется битом BODLEVEL на 2.7В если бит не запрограммирован или на 4В если

бит запрограммирован. Уровень срабатывания имеет гистерезис 50мВ.

Для того, чтобы произошел сброс падение напряжения до уровня сра­батывания должно продержаться не менее 3мкС для уровня срабатывания 4В (7мкС для 2.7В).

# СБРОС ПО СТОРОЖЕВОМУ ТАЙМЕРУ

По истечению периода работы сторожевого таймера генерируется им­пульс длительностью 1 период тактовой частоты. По заднему фронту этого импульса запускается таймер, отсчитывающий время сброса

# РЕГИСТР СОСТОЯНИЯ ПРОЦЕССОРА - MCUSR

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

# MCUSR

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **34h(54h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **-** | **-** | **WDRF** | **BORF** | **EXTRF** | **PORF** |
| **R** | **R** | **R** | **R** | **R** | **R** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Биты 7..4 - зарезервированы. В AT90S2333/4433 эти биты зарезервиро­ваны и всегда читаются как 0.

Бит 3 - WDRF - этот бит устанавливается при сбросе от сторожевого таймера. Бит обнуляется при сбросе по включению питания или записью нуля.

Бит 2 - BORF - этот бит устанавливается при сбросе от схемы слежения за напряжением питания. Бит обнуляется при сбросе по включению питания или записью нуля.

Бит 1 - EXTRF - этот бит устанавливается при внешнем сбросе Бит об­нуляется при сбросе по включению питания или записью нуля.

Бит 0 - PORF - этот бит устанавливается при сбросе по включению пи­тания, бит очищается записью нуля.

Чтобы определить источник сброса пользователь должен в самом на­чале программы прочитать регистр MCUSR и обнулить все биты. Источник сброса определяется проверкой соответствующих флагов сброса.

# ОБРАБОТКА ПРЕРЫВАНИЙ

AT90S2333/4433 имеют два регистра маскирования прерываний GIMSK - общий регистр маски прерываний и TIMSK - регистр маски прерываний от таймеров/счетчиков.

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 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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **3Bh(5Bh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **INT**1 | **INT0** | **-** | **-** | **-** | **-** | **-** | **-** |
| **R\W** | **R\W** | **R** | **R** | **R** | **R** | **R** | **R** |
| Начальное значение | **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 002h 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 AT90S2333/4433 эти биты зарезервиро­ваны и всегда читаются как 0.

# ОБЩИЙ РЕГИСТР ФЛАГОВ ПРЕРЫВАНИЙ

 **GIFR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **3Bh(5Bh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **INTF1** | **INTF0** | **-** | **-** | **-** | **-** | **-** | **-** |
| **R\W** | **R\W** | **R** | **R** | **R** | **R** | **R** | **R** |
| Начальное значение | **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 I регистра SREG и бит INT1 в GIMSK, происходит переход на вектор прерывания по адресу 002h. 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 I регистра SREG и бит INT0 в GIMSK, происходит переход на вектор прерывания по адресу 001h. Флаг очищается после вы­полнения обработчика прерывания. Кроме того, флаг можно очистить, за­писав в него логическую единицу.

Биты 5..0 - зарезервированы. В AT90S2333/4433 эти биты зарезервиро­ваны и всегда читаются как 0.

**РЕГИСТР МАСКИ ПРЕРЫВАНИЯ ОТ ТАЙМЕРА/СЧЕТЧИКА – TIMSK**

**TIMSK**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **39h(59h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **TOIE**1 | **OCIE**1**A** | **-** | **-** | **TICIE**1 | **-** | **TOIE0** | **-** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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 0000h.

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 004h) выполняется при равенстве таймера/счетчика 1 и регистра совпадения. Во флаговом регистре TIFR устанавливается ("1") флаг совпадения.

Биты 5,4 - зарезервированы; в AT90S2333/4433 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 прерывания по входу захвата. Соответствующее прерывание (вектор 003h) выполняется по сигналу захвата на выводе 11 (PD6/ICP). Во флаговом регистре TIFR устанавливается ("1") флаг захвата.

Бит 2 - зарезервирован; в AT90S2333/4433 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 (006h). Флаг переполнения (TOV0) во флаговом регистре прерыва­ний (TIFR) таймеров/счетчиков устанавливается в 1.

Бит 0 - зарезервирован; в AT90S2333/4433 этот бит зарезервирован и всегда читается как 0.

# РЕГИСТР ФЛАГОВ ПРЕРЫВАНИЙ ОТ ТАЙМЕРОВ/СЧЕТЧИКОВ – TIFR

**TIFR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **38h(58h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **TOV**1 | **OCF**1 | **-** | **-** | **ICF**1 | **-** | **TOV0** | **-** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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 0000h.

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 AT90S2333/4433 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 AT90S2333/4433 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 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 0 - зарезервирован; в AT90S2333/4433 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 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. 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.

**РЕГИСТР УПРАВЛЕНИЯ МИКРОКОНТРОЛЛЕРОМ – MCUCR**

Этот регистр содержит биты общего управления микроконтроллером.

**MCUCR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **35h(55h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **SE** | **SM** | **ISC**11 | **ISC**10 | **ISC**01 | **ISC**00 |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Биты 7,6 - зарезервированы. В AT90S2333/4433 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. Для устранения нежелательного перехода в режим пониженного энергопотребления рекомендуется устанавливать этот бит непосредственно перед выполнением 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 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:

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 6 приведена установка битов для задания срабатывания по уровню и фронтам.

**Таблица 5. Управление срабатыванием прерывания 1.**

|  |  |  |
| --- | --- | --- |
| **ISC11** | **ISC10** | **Описание** |
| **0** | **0** | Запрос на прерывание генерируется по низкому уровню напряжения на входе INT1 |
| **0** | **1** | Запрос по изменению уровня на входе INT1 |
| **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 6. Управление срабатыванием прерывания 0.**

|  |  |  |
| --- | --- | --- |
| **ISC01** | **ISC0** | **Описание** |
| **0** | **0** | Запрос на прерывание генерируется по низкому уровню напряжения на входе INT0 |
| **0** | **1** | Запрос по изменению уровня на входе INT0 |
| **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 дольше времени отработки сброса. Иначе процессор не 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. При выходе из режима холостого хода процессор запускается без задержки.

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, brown-out

сбросу или внешнему прерыванию по уровню.

Если для вывода из экономичного режима используется прерывание по уровню, низкий уровень должен удерживаться на время достаточное для запуска процессора. Это увеличивает устойчивость процессора к помехам. Изменение уровня дважды проверятся с периодом генератора сторожевого таймера, если обе выборки сигнала имеют необходимый уровень, процессор включается. Номинальный период сторожевого таймера 1uS при 5В питания и температуре 25 градусов Цельсия.

При выходе из экономичного режима, от времени появления условия выхода до включения процессора проходит некоторое время необходимое для запуска кварцевого генератора. Задержка включения определяется те­ми же битами CKSEL, что и время сброса. Длительность задержки на вклю­чение приведена в таблице 7.

Если условие включения исчезнет до того, как процессор запустит­ся, например, низкий уровень на входе прерывания продержится недоста­точно долго, процессор не выйдет из экономичного режима.

**Таблица 7. Установка задержки включения**

|  |  |
| --- | --- |
| **CKSEL [2:0]** | **Время запуска** |
| **000** | **6CK** |
| **001** | **6CK** |
| **010** | **16K CK** |
| **011** | **16K CK** |
| **100** | **16K CK** |
| **101** | **1K CK** |
| **110** | **1K CK** |
| **111** | **1K CK** |

**ТАЙМЕРЫ/СЧЕТЧИКИ**

В AT90S2333/4433 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 (CK), импульсы предварительного делите­ля (CK/8, CK/64, CK/256 или CK/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 - CK, импульсы с предварительного делителя (CK/8, CK/64, CK/256 или CK/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**

**TCCR0**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **33h(53h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **-** | **-** | **-** | **CS**02 | **CS**01 | **CS00** |
| **R** | **R** | **R** | **R** | **R** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Биты 7..3 - зарезервированы. В AT90S2333/4433 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** | **CK** |
| **0** | **1** | **0** | **CK/8** |
| **0** | **1** | **1** | **CK/64** |
| **1** | **0** | **0** | **CK/256** |
| **1** | **0** | **1** | **CK/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.**

**TCNT0**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **32h(52h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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.

**16-РАЗРЯДНЫЙ ТАЙМЕР/СЧЕТЧИК 1**

16-разрядный таймер/счетчик 1 может получать импульсы тактовой частоты - CK, импульсы с предварительного делителя (CK/8, CK/64, CK/256 или CK/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 A УПРАВЛЕНИЯ ТАЙМЕРОМ/СЧЕТЧИКОМ 1 - TCCR1A**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **2Fh(4Fh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **COM**1**A**1 | **COM**1**A**0 | **-** | **-** | **-** | **-** | **PWM**11 | **PWM**10 |
| **R\W** | **R\W** | **R** | **R** | **R** | **R** | **R\W** | **R\W** |
| Начальное значение | **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 таблице 9.

В режиме ШИМ эти биты имеют другие функции, которые будут указаны при рассмотрении работы ШИМ.

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. 10). Подроб­нее этот режим будет рассмотрен ниже.

Таблица 9. Установка режима совпадения

|  |  |  |
| --- | --- | --- |
| **COM1A1** | **COM1A0** | **Описание** |
| **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 |

Таблица 10. Установка режима работы ШИМ

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

**РЕГИСТР B УПРАВЛЕНИЯ ТАЙМЕРОМ/СЧЕТЧИКОМ 1 - TCCR1B**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **2Eh(4Eh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **ICNC**1 | **ICES**1 | **-** | **-** | **CTC**1 | **CS**12 | **CS**11 | **CS**10 |
| **R\W** | **R\W** | **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Бит7 - ICNC1: Подавитель входного шума входа захвата: Если этот бит сброшен (0), подавление входного шума входа захвата запрещено. При это 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 AT90S2333/4433 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 0000h в такте сле­дующем за событием совпадения. Если бит сброшен, таймер/счетчик 1 про­должает считать, совпадение не отражается на состоянии таймера/счетчика. Поскольку совпадение обнаруживается процессором в цикле следующем

за событием совпадения, в следующем после совпадения такте в счетчик запишется число на 1 большее значения регистра совпадения. При коэффи­циенте деления 1, установке регистра совпадения на величину С и уста­новленном бите CTC1, таймер принимает следующие значения:

...С-1,С,С+1,0,1...

при делении тактовой частоты на 8:

 ...С-1,С-1,С-1,С-1,С-1,С-1,С-1,С-1,С,С,С,С,С,С,С,С, С+1,0,0,0,0,0,0,0,0,1...

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 11. Выбор источника счетных импульсов

|  |  |  |  |
| --- | --- | --- | --- |
| CS12 | CS11 | CS10 | Описание |
| **0** | **0** | **0** | **Таймер/счетчик 1 остановлен** |
| **0** | **0** | **1** | **CK** |
| **0** | **1** | **0** | **CK/8** |
| **0** | **1** | **1** | **CK/64** |
| **1** | **0** | **0** | **CK/256** |
| **1** | **0** | **1** | **CK/1024** |
| **1** | **1** | **0** | **Спадающий фронт на выводе T1** |
| **1** | **1** | **1** | **Нарастающий фронт на выводе T1** |

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 1 - TCNT1H И TCNT1L**

**TCNT1H**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **2Dh(4Dh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**TCNT1L**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **2Ch(4Ch)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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). Этот вре-

менный регистр также используется при доступе к регистрам OCR1 и ICR1.

Если доступ к регистру с использованием TEMP производится и в основной программе и в обработчике прерывания, на время доступа к регистру из основной программы прерывания должны быть запрещены.

- 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, то есть одновременно читаются все 16-разрядов. При последующем чтении регистра TCNT1H, данные берутся из регистра TEMP. То есть при чтении 16-разрядного значения счетчика первым должен читаться регистр TCNT1L.

Таймер/счетчик 1 организован как суммирующий счетчик (в режиме ШИМ - суммирующий/вычитающий) с возможностью чтения и записи. Если за­дан источник тактовых импульсов для таймера/счетчика 1, после записи в него нового значения, счет продолжается с следующем за операцией записи

периоде тактовой частоты.

**РЕГИСТР СОВПАДЕНИЯ ТАЙМЕРА/СЧЕТЧИКА 1 - OCR1H И OCR1L**

**OCR1H**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **2Bh(4Bh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  |  |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**OCR1L**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  **3Dh(5Dh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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 (TEMP). 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 TEMP используется при доступе к TCNT1 и ICR1, поэтому ес­ли временный регистр используется в основной программе и в прерывани­ях, при доступе к TEMP из основной программы прерывания должны запрещаться.

**РЕГИСТР ЗАХВАТА ТАЙМЕРА/СЧЕТЧИКА 1 - ICR1H И ICR1L**

**ICR1H**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **25h(45h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  |  |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**ICR1L**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **24h(44h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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 (TEMP). При чтении старшего байта, он принимается из временного регис­тра. Таким образом для чтения 16-разрядного регистра первым должен чи­таться младший байт. Регистр TEMP используется при доступе к TCNT1 и OCR1, поэтому ес­ли временный регистр используется в основной программе и в прерывани­ях, при доступе к TEMP из основной программы прерывания должны запрещаться.

**ТАЙМЕР/СЧЕТЧИК 1 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.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.11).

Таблица 12. Конечное значение таймера и частота ШИМ.

|  |  |  |
| --- | --- | --- |
| Разрешение ШИМ | Конечное значение таймера | Частота ШИМ |
| 8 бит | 00FFh (255) | Ftc1/510 |
| 9 бит | 01FFh (511) | Ftc1/1022 |
| 10 бит | 03FFh (1023) | Ftc1/2046 |

Таблица 13. 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 (неинвертирующий 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. Во промежуток времени между записью во временный регистр и пере­писыванием его в OCR1, при обращении к OCR1 читается содержимое вре­менного регистра. Если OCR1A содержит значение 0000h или конечное значение (TOP), вывод OC1 остается в том состоянии, которое определяется установками COM1A1 и COM1A0. Это показано в табл. 14.

Таблица 14. Выход ШИМ для OCR=0000h или TOP

|  |  |  |  |
| --- | --- | --- | --- |
| COM1A1 | COM1A0 | OCR1A | вывод OC1 |
| **1****1** | **0****0** | 0000hTOP | низкийвысокий |
| **1****1** | **1****1** | 0000hTOP | высокийнизкий |

В режиме ШИМ флаг переполнения таймера 1 (TOV1) устанавливается когда счетчик изменяет направление счета в точке 0000h. 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 MHz (это типовое значение частоты для питания 5В). Управляя предварительным делителем сторожевого таймера можно за­давать интервал сброса таймера от 16 до 2048 mS. 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**

**WDTCR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **21h(41h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  | **WDTOE** | **WDE** | **WDP2** | **WDP1** | **WDP0** |
| **R** | **R** | **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Биты 7..5 - зарезервированы. В AT90S2333/4433 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 15 приведены различные значения установок предварительного дели­теля и соответствующие им временные интервалы для напряжения питания Vcc=5V.

Таблица 15. Установки предварительного делителя сторожевого таймера

|  |  |  |  |
| --- | --- | --- | --- |
| **WDP2** | **WDP1** | **WDP0** | **период времени** |
| **0** | **0** | **0** | **16K циклов** |
| **0** | **0** | **1** | **32K циклов** |
| **0** | **1** | **0** | **64K циклов** |
| **0** | **1** | **1** | **128K циклов** |
| **1** | **0** | **0** | **256 циклов** |
| **1** | **0** | **1** | **512 циклов** |
| **1** | **1** | **0** | **1024 циклов** |
| **1** | **1** | **1** | **2048 циклов** |

**ЧТЕНИЕ И ЗАПИСЬ В ЭНЕРГОНЕЗАВИСИМУЮ ПАМЯТЬ**

Регистры доступа к энергонезависимой памяти (EEPROM) расположены в пространстве ввода/вывода. Время записи лежитв диапазоне 2.5-4 mS и зависит от напряжения питания. Это самотактируемая функция которая, однако, позволяет пользователю определить, можно ли записывать следующий байт. Для определе­ния возможности записи в EEPROM можно использовать специальное преры­вание по готовности EEPROM. Инициированная запись в EEPROM заканчивается даже при возникнов­ении условия сброса. 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. При чтении из EEPROM процессор приостанавливается на четыре ма­шинных цикла перед тем как начнет выполняться следующая команда.

**РЕГИСТР АДРЕСА EEPROM – EEAR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1Eh(3Eh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **EEAR7** | **EEAR6** | **EEAR5** | **EEAR4** | **EEAR3** | **EEAR2** | **EEAR1** | **EEAR0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Регистр EEAR задает адрес одного из 128/256 байт адресного прост­ранства EEPROM. Байты данных адресуется линейно в диапазоне от 0 до 127/255. Начальное значение регистра EEAR неопределено, поэтому перед доступом к EEPROM в этот регистр должно быть записано требуемое число.

**РЕГИСТР ДАННЫХ EEPROM – EEDR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1Dh(3Dh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **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**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1Ch(3Ch)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  | **EERIE** | **EEMWE** | **EEWE** | **EERE** |
| **R** | **R** | **R** | **R** | **R** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Биты 7..4 - зарезервированы. В AT90S2333/4433 эти биты зарезервиро­ваны и всегда читаются как 0.

Бит 3 - EERIE - Разрешение прерывания по готовности EEPROM. Если ус­тановлены биты I в регистре SREG и EERIE, разрешается прерывание по готовности EEPROM. Это прерывание возникает непрерывно, если сброшен бит EEWE.

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 правильных 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 (порядок шагов 2 и 3 необязателен):

|  |  |
| --- | --- |
| **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 mS для Vcc=5V и 4mS для Vcc=2.7), 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.

**ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС SPI**

Интерфейс SPI позволяет производить высокоскоростной синхронный обмен данными между AT90S2333/4433 и периферийными устройствами или несколькими процессорами. SPI-интерфейс предлагает следующие возможно­сти:

1. Полностью дуплексная 3-проводная синхронная передача данных;
2. Работа в режиме ведущего или ведомого;
3. Передача начиная со старшего или младшего бита;
4. Четыре программируемые скорости передачи;
5. Флаг прерывания по окончанию передачи;
6. Флаг защиты от коллизий при записи
7. Выход из режима Idle.

Соединение между ведущим и ведомым контроллерами показана ниже. Вывод PB5(SCK) является выходом тактовых импульсов для ведущего кон­троллера и входом для ведомого. Запись в регистр данных SPI ведущего контроллера запускает тактовый генератор. Записанные данные сдвигаются через вывод PB3(MOSI) на вывод PB3(MOSI) ведомого контроллера. После того как байт будет выведен тактовый генератор останавливается и вы­ставляет флаг окончания передачи (SPIF). Если разрешены прерывания (установлен бит SPIE в регистре SPCR), вызывается соответствующее прерывание. Вывод PB2(SS) ведомого контроллера должен быть подключен на землю. Два сдвиговых регистра в ведущем и ведомом контроллерах мож­но рассматривать как один распределенный 16-разрядный регистр сдвига. Когда данные сдвигаются из ведущего контроллера в ведомый, то же самое происходит в обратном направлении. За один цикл сдвига ведущий и ведо­мый контроллеры обмениваются байтами данных.

Система имеет одиночный буфер в направлении передачи и двойной в направлении приема. Передаваемый символ не записывается в регистр дан­ных SPI до тех пор, пока передача не завершится. При приеме до завер­шения операции сдвига данные должны быть прочитаны из регистра данных. Иначе предыдущий символ теряется. При разрешении SPI выводы MOSI, MISO, SCK и SS устанавливаются на ввод/вывод в соответствии с таблицей.

Таблица 16. Установка выводов SPI

|  |  |  |
| --- | --- | --- |
| Вывод | Направление в режиме ведущего | Направление в режиме ведомого |
| MOSI | Определяется пользователем | Вход |
| MISO | Вход | Определяется пользователем |
| SCK | Определяется пользователем | Вход |
| SS | Определяется пользователем | Вход |

**РАБОТА ВЫВОДА SS**

Если SPI сконфигурирован как ведущий (установлен бит MSTR в SPCR), пользователь может задавать направление ввода для ножки SS. Ес­ли SS сконфигурирован на вывод, ножка работает как бит ввода/вывода общего назначения не влияющий на работу SPI. Если ножка сконфигуриро­вана на ввод, на нее необходимо подать высокий уровень, чтобы SPI ра­ботал в режиме ведущего. Если в режиме ведущего ножка SS установлена на ввод и внешним устройством устанавливается в низкий уровень, систе­мой SPI это воспринимается как выбор шины другим ведущим устройством и попытка передать данные ведомому. Для избежания конфликтов на шине си­стема SPI делает следующее:

1. Сбрасывается бит MSTR в SPCR, система SPI становится ведомой. При этом ножки MOSI и SCK становятся входами.

2. Устанавливается флаг SPIF в SPSR, если разрешены прерывания, будет исполнена программа обслуживания прерывания.

Таким образом если передача по SPI производится в режиме ведущего и существует возможность установки вывода SS в низкий уровень, преры­вание должно отслеживать установлен ли бит MSTR. Если бит MSTR будет сброшен при переходе в режим ведомого контроллера, обратно он должен быть установлен пользователем.

Если система SPI сконфигурирована как ведомая, вывод SS всегда является входом. Когда SS переводится в низкий уровень, система SPI активируется и ножка MISO, если это задано пользователем, становится выходом. Другие выводы являются входами. Если внешним устройством на вывод SS подан высокий уровень, все выводы становятся входами, а сис­тема SPI становится пассивной, т.е. не будет принимать данные.

# Режимы обмена данными

Биты CPHA и CPOL определяют четыре комбинации фазы и полярности SCK относительно битов данных. Битом CPOL задается полярность импульсов на выводе SCK, при CPOL=0 полярность импульсов положительная, при от­сутствии импульсов на выводе удерживается низкий уровень. При CPOL=1 полярность импульсов отрицательная, при отсутствии импульсов на выходе высокий уровень. Битом CPHA задается фронт по которому обрабатываются биты данных. При CPHA=0 обработка происходит по переднему фронту им­пульсов SCK (для CPOL=0 переход из низкого уровня в высокий, для CPOL=1 - из высокого в низкий). При CPHA=1 обработка происходит по за­днему фронту импульсов SCK.

**РЕГИСТР УПРАВЛЕНИЯ SPI – SPCR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Dh(2Dh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **SPIE** | **SPE** | **DORD** | **MSTR** | **CPOL** | **CPHA** | **SPR1** | **SPRO0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Бит 7 - SPIE - Разрешение прерываний от SPI - если этот бит установ­лен, при установке бита SPIF в регистре SPSR выполняется прерывание от SPI, если разрешены общие прерывания.

Бит 6 - SPE - Разрешение SPI - Этот бит должен быть установлен для осуществления любых операций по шине SPI.

Бит 5 - DORD - Порядок бит данных - Если этот бит установлен, при об­мене первым передается младший бит данных. Если бит сброшен – первым передается старший бит.

Бит 4 - MSTR - Выбор ведущего/ведомого - При установке этого бита ши­на SPI работает в режиме ведущего, при сбросе - в режиме ведомого. Ес­ли вывод SS сконфигурирован как вход и устанавливается в низкий уро­вень когда установлен бит MSTR, бит MSTR сбрасывается и устанавливает­ся бит SPIF в регистре SPSR. Для возобновления работы в режиме ведуще­го пользователь должен установить этот бит.

Бит 3 - CPOL - Полярность тактовых сигналов - Этим битом задается по­лярность импульсов на выводе SCK, при CPOL=0 полярность импульсов по­ложительная, при отсутствии импульсов на выводе удерживается низкий уровень. При CPOL=1 полярность импульсов отрицательная, при отсутствии импульсов на выходе высокий уровень.

Бит 2 - CPHA - Фазировка тактовых импульсов. - Этим битом задается фронт по которому обрабатываются биты данных. При CPHA=0 обработка происходит по переднему фронту импульсов SCK (для CPOL=0 переход из низкого уровня в высокий, для CPOL=1 - из высокого в низкий). При CPHA=1 обработка происходит по заднему фронту импульсов SCK.

Биты 1,0 - SPR1,SPR0 - Выбор тактовой частоты SPI. - Этими битами за­дается скорость работы SPI шины для ведущего устройства. Для ведомого контролерра установка этих бит не имеет значения. Соотношение тактовой частоты процессора и частоты SCK приведено в таблице:

Таблица 17. Соотношение между SCK и тактовой частотой процессора

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SPR1 | SPR0 | Частота SCK | SPR1 | SPR0 | Частота SCK |
| **0** | **0** | Fck/4 | 1 | 0 | Fck/64 |
| **0** | **1** | Fck/16 | 1 | 1 | Fck/128 |

**РЕГИСТР СОСТОЯНИЯ SPI – SPSR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Eh(2Eh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **SPIF** | **WCOL** | **-** | **-** | **-** | **-** | **-** | **-** |
| **R** | **R** | **R** | **R** | **R** | **R** | **R** | **R** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Бит 7 - SPIF - флаг прерывания SPI. После завершения обмена устанав­ливается бит SPIF и генерируется прерывание, если установлен бит SPIE в SPCR и разрешены глобальные прерывания. Этот флаг устанавливается и в том случае, если на вывод SS, сконфигурированный как вход, подается низкий уровень, когда SPI работает в режиме ведущего. Флаг SPIF сбра­сывается аппаратно при выполнении соответствующего вектора прерывания. Кроме того бит SPIF, если он был установлен, сбрасывается при первом же чтении регистра SPSR и последующем доступе к регистру данных SPI.

Бит 6 - WCOL - Ошибка записи. - Этот бит устанавливается при записи в регистр SPDR во время обмена. Чтение регистра во время обмена дает не­верный результат, запись в него не производится. Так же как и бит SPIF, WCOL сбрасывается (если он был установлен) при первом же чтении регистра SPSR и последующем доступе к регистру SPDR.

Биты 5..0 - зарезервированы. В AT90S2333/4433 эти биты зарезервиро­ваны и всегда читаются как 0.

SPI интерфейс также используется для загрузки и чтения памяти программ и содержимого EEPROM процессора, программирование процессора через SPI интерфейс будет рассмотрено ниже.

**РЕГИСТР ДАННЫХ SPI – SPDR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Fh(2Fh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Регистр данных SPI доступен для чтения и записи и используется для обмена данными между набором регистров и регистром сдвига SPI. За­пись в регистр инициирует передачу данных. При чтении регистра читают­ся данные и буфера регистра сдвига.

**УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПРИЕМО-ПЕРЕДАТЧИК**

В состав AT90S2333/4433 входит универсальный асинхронный приемо­передатчик (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 раздельных прерывания - завершение передачи, очистка регист­ра передачи и завершение приема.

- Режим мультипроцессорного обмена.

**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 того как был выдвинут стоповый бит 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 того, как будет выдвинут стоповый бит 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 выдвижения 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 буферирован 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 установке RXEN, приемник подключен к выводу PD0 независимо от со­стояния бита DDD0 в регистре DDRD.

Режим мультипроцессорного обмена Режим многопроцессорного обмена позволяет нескольким ведомым про­цессорам принимать данные от ведущего. Для этого сначала декодируется адресный байт, который определяет к какому из процессоров обращается ведущий.

Если ведомый процессор принял правильный адрес, последующие

байты он принимает как данные, в то время как остальные ведомые про­цессоры игнорируют принимаемые байты до приема следующего адреса. Для работы в режиме ведущего процессор должен установить 9-би­товый режим передачи (установлен бит CHR9 в UCSRB). Для передачи адре­сного байта девятый бит должен устанавливаться в 1, и сбрасываться для передачи байтов данных. В ведомых процессорах механизм приема слегка отличается для 8-ми и 9-ти битового режима приема. При приеме восьми бит (сброшен бит CHR9 в UCSRB), стоповый бит для адресного байта равен единице и равен нулю для байт данных. В 9-ти битовом режиме для адресного байта устанавли­вается 9-й бит, для байт данных он будет сброшен, стоповый бит всегда будет равен 1. Для обмена данными в многопроцессорном режиме необходимо выпол­нить следующую процедуру.

1. Все подчиненные процессоры устанавливают обмен в многопроцес­сорном режиме (установлен бит MPCM в UCSRA)

2. Ведущий процессор посылает адресный байт, все подчиненные про­цессоры читают и принимают этот байт. В ведомых процессорах устанавли­вается флаг RXC в UCSRA.

3. Каждый из ведомых процессоров читает регистр UDR и определяет был ли он выбран. Если процессор выбран, он сбрасывает флаг MPCM в UCSRA, иначе он будет ожидать следующего адресного байта.

4. Для каждого принятого байта данных в ведомом процессоре уста­навливается флаг завершения приема (RXC в UCSRA). Кроме того в 8-битовом режиме будет генерироваться ошибка кадра (FE в UCSRA), по­скольку стоповый бит будет равен 0. В других подчиненных процессорах

установлен бит MPCM, поэтому байты данных будут игнорироваться, ре­гистр UDR не записывается, флаги RXC и FE не устанавливаются.

5. После передачи последнего байта данных процесс повторяется с шага 2.

**Управление UART**

**РЕГИСТР ВВОДА/ВЫВОДА UART**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Ch(2Ch)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **MSB** |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **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 УПРАВЛЕНИЯ И СОСТОЯНИЯ UART (UСSRA)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Bh(2Bh)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **RXC** | **TXC** | **UDRE** | **FE** | **OR** | **-** | **-** | **MPCM** |
| **R** | **R\W** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** |

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 предающее устройство должно перейти в режим приема и освободить линию связи сразу по окон­чанию передачи. Если установлен бит TXIE в регистре UCR, при установке TXC выполняется прерывания по окончанию 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, 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 буферирован, т.е. остается установленным до тех пор, пока из регистра UDR не будут прочитаны правильные данные. Бит OR сбрасывается когда принятые данные переписываются в UDR.

Биты 2..1 - зарезервированы. В AT90S2333/4433 эти биты зарезерви­рованы и всегда читаются как 0.

Бит 0 - MPCM - режим мультипроцессорного обмена. Этот бит исполь­зуется для перехода в режим мультипроцессорного обмена. Этот бит уста­навливается когда ведомый процессор ожидает приема адресного байта. Ко­гда ведомый процессор распознает свой адрес он должен сбросить бит MPCM и начать прием данных.

**РЕГИСТР УПРАВЛЕНИЯ И СОСТОЯНИЯ UART (UСSRB)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0Ah(2Ah)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **RXCIE** | **TXCIE** | **UDRIE** | **RXEN** | **TXEN** | **CHR9** | **RXB8** | **TXB8** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R** | **W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Бит 7 - RXCIE - Разрешение прерывания по окончанию приема. Если этот бит установлен (1), установка бита RXC в регистре UCSRA приводит к вы­полнению прерывания по окончанию приема (при условии что разрешены прерывания).

Бит 6 - TXCIE - Разрешение прерывания по окончанию передачи. Если этот бит установлен, установка бита TXC в UCSRA приводит к выполнению прерывания по окончанию передачи (при условии, что прерывания разреше­ны).

Бит 5 - UDRIE - Прерывание по очистке регистра данных последователь­ного порта. Если этот бит установлен, установка бита UDRE в UCSRA 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 посылки. 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 это делитель частоты, который генери­рует скорости в соответствии с нижеприведенным выражением:

BAUD = Fck / (16\*(UBRR+1))

здесь BAUD - скорость передачи (бод)

Fck - частота тактового генератора процессора

UBRR - содержимое регистров UBRRH и UBRR (0...4095)

В следующей таблице приведены значения регистра UBRR и процентное отклонение от стандартной скорости передачи для стандартных частот кварцевых генераторов.

Таблица.18

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Скор.,бод | 1.0000MHz | Ош.% | Скор.,бод | 1.8432MHz | Ош.% | Скор.,бод | 2.0000MHz | Ош.% |
| 240048009600144001920028800384005760076800115200 | 251263211000 | 0.20.27.57.87.87.822.97.822.984.3 | 240048009600144001920028800384005760076800115200 | 4723117532110 | 0000000033.00 | 240048009600144001920028800384005760076800115200 | 5125128632110 | 0.20.20.23.77.57.87.87.822.97.8 |
| Скор.,бод | 3.2768MHz | Ош.% | Скор.,бод | 3.6864MHz | Ош.% | Скор.,бод | 4.0000MHz | Ош.% |
| 240048009600144001920028800384005760076800115200 | 844220131064321 | 0.40.81.61.63.11.66.312.512.512.5 | 240048009600144001920028800384005760076800115200 | 954723151175321 | 0.00.00.00.00.00.00.00.00.00.0 | 240048009600144001920028800384005760076800115200 | 1035125161286321 | 0.20.20.22.10.23.77.57.87.87.8 |
| Скор.,бод | 7.3728MHz | Ош.% | Скор.,бод | 8.0000MHz | Ош.% | Скор.,бод | 9.2160MHz | Ош.% |
| 240048009600144001920028800384005760076800115200 | 191954731231511753 | 0.00.00.00.00.00.00.00.00.00.0 | 240048009600144001920028800384005760076800115200 | 2071035134251612863 | 0.20.20.20.80.22.10.23.77.57.8 | 240048009600144001920028800384005760076800115200 | 2391195939291914974 | 0.00.00.00.00.00.00.00.06.70.0 |

**РЕГИСТР СКОРОСТИ ПЕРЕДАЧИ (UBRR)**

**UBRRHI**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **03h(23h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  | **MSB** | **-** | **-** | **-** |
| **R** | **R** | **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**UBRR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **09h(29h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | **LSB** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

Это 12-разрядный регистр, который задает скорость передачи последо­вательного порта в соответствии с выражением приведенным выше. В реги­стре UBRRHI хранятся 4 старших бита, в UBRR - 8 младших бит.

**АНАЛОГОВЫЙ КОМПАРАТОР**

Аналоговый компаратор сравнивает входные напряжение на положи­тельном входе PD6 (AIN0) и отрицательном входе PD7(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. Пользова-

тель может установить установить срабатывание прерывания по нарастаю­щему или спадающему фронту, или по переключению.

**РЕГИСТР УПРАВЛЕНИЯ И СОСТОЯНИЯ АНАЛОГОВОГО КОМПАРАТОРА (ACSR).**

**ACSR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **08h(28h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **ACD** | **AINBG** | **ACO** | **ACI** | **ACIE** | **ACIC** | **ACIS**1 | **ACIS0** |
| **R\W** | **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **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 - AINBG - Выбор напряжения сравнения аналогового компаратора. Когда этот бит установлен, вывод AIN0 подключается к напряжению 1.22 + 0.05В. Когда бит сброшен возобновляется нормальная работа вывода AIN0.

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/счетчика1 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/счетчика1, должен быть установлен бит TICIE1 в регистре TIMSK.

Биты 1,0 - ACIS1, ACIS0 - Выбор режима прерывания аналогового компа­ратора. Различные установки приведены ниже.

Таблица 19. 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.

**АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ**

Основные характеристики:

- разрешение 10 бит

- точность +- 1/2 младшего разряда

- время преобразования 65-260 uS

- 6 мультиплексирумых входов

- работа с полным размахом напряжения

- непрерывный режим или одиночные преобразования

- прерывание по завершению преобразования

- подавление шума в режиме Sleep

AT90S2333/4433 имеют 10-разрядный АЦП последовательного приближения. АЦП подключен к выходу шестивходового мультиплексора. Мультиплексор по­зволяет подключать на вход АЦП любой из входов порта C. В состав АЦП входит усилитель выборки-хранения, который позволяет сохранять на входе АЦП уровень напряжения постоянный за время преобразования.

АЦП имеет два отдельных вывода питания AVCC и AGND. ФПТВ должен подключаться к GND, напряжение на AVCC не должно отличаться от VCC более чем на +-0.3В. Внешнее опорное напряжение должно подаваться на вывод AREF. Это напряжение должно лежать в пределах AGND-AVCC.

**РАБОТА АЦП**

АЦП может работать в двух режимах - одиночного преобразования и непрерывном. В режиме одиночного преобразование каждое преобразование инициируется пользователем. В непрерывном режиме АЦП производит непре­рывную обработку входного сигнала и обновляет регистр данных АЦП. Пе­реключение режимов осуществляется битом ADFR в регистре ADCSR. Регистр ADMUX выбирает один из шести входов подключаемый ко входу АЦП. Кроме того в качестве входа можно использовать фиксированное опорное напряжение. Работа АЦП разрешается записью логической "1" в бит разрешения АЦП (ADEN в ADCSR). Перед первым преобразованием после разрешения АЦП выполняется одно "пустое" преобразование инициализирующее АЦП. Для по­льзователя единственное отличие первого преобразования от последующих заключается в том, что оно занимает 25 тактовых импульсов вместо обыч­ных 13.

Преобразование запускается записью логической 1 в бит запуска АЦП - ADCS. Этот бит остается установленным пока идет преобразование и ап­паратно сбрасывается, когда преобразование завершается. Если во время преобразования происходит переключение канала, пред переключением АЦП завершает текущее преобразование.

Операция выборки-хранения занимает один цикл после запуска преоб­разования. Это позволяет пользователю устанавливать номер входного ка­нала одновременно с запуском преобразования Поскольку результат преобразования занимает 10 разрядов, для хра­нения результата используется два регистра ADCH и ADCL, которые должны быть прочитаны для получения результата. Для защиты данных, чтобы обеспечить чтение результата одного преобразования из обоих регистров используется специальная логика защиты данных.

Этот механизм работает так:

При чтении данных первым должен читаться регистр ADCL. Доступ АЦП к регистру данных блокируется. Это приводит к тому, что если прочитан байт ADCL и преобразование завершилось до того как прочитан регистр ADCH, регистр данных не изменяется и результат преобразования будет потерян. Доступ АЦП к регистру данных разрешается после чтения ADCH. АЦП имеет собственное прерывание, которое выполняется по заверше­нию преобразования. Если доступ АЦП к регистру данных запрещен (между чтением ADCL и ADCH), прерывание вызывается даже если данные потеряны. Предварительный делитель АЦП работает с тактовыми частотами в диапазоне 50-200 kHz. Для полного преобразования АЦП необходимо 13 тактов, т.е. преобразование занимает от 65 до 260 uS. Корректное значение выхода АЦП гарантируется только для тактовых частот не выходящих из приведенного диапазона. Для генерации тактовой частоты АЦП из тактовой частоты процессора выше 100 кГц используются биты ADPS0-ADPS2. Предварительный делитель начинает счет в момент разрешения АЦП установкой бита ADEN в ADCSR. Предварительный делитель продолжает работать до тех пор, пока установлен бит ADEN и непрерывно сбрасывается если бит ADEN сброшен. Функция подавления шумов АЦП Одна из особенностей АЦП - функция подавления шумов, АЦП может осуществлять преобразования в режиме холостого хода, это позволят уменьшить шумы, наводимые ядром процессора. Чтобы воспользоваться этой возможностью, необходимо произвести следующую процедуру:

1. Удостоверьтесь, что АЦП разрешен и не занят преобразованием. Необходимо выбрать режим одиночного преобразования и разрешить прерывания по окончанию преобразования .

ADEN=1

ADSC=0

ADFR=0

ADIE=1

2. Перейти в режим холостого хода. АЦП начнет преобразование по­сле того как остановится процессор.

3. Если за время преобразования не произойдут другие прерывания, по окончанию преобразования процессор выйдет из режима холостого хода и выполнит прерывание по окончанию преобразования.

**РЕГИСТР УПРАВЛЕНИЯ МУЛЬТИПЛЕКСОРОМ АЦП – AMUX**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **07h(27h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **ADCBG** | **-** | **-** | **-** | **MUX2** | **MUX1** | **MUX0** |
| **R\W** | **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Бит 7 - зарезервирован; в AT90S2333/4433 этот бит зарезервирован и при записи должен устанавливаться в 0.

Бит 6 - ADCBG - выбор опорного напряжения АЦП - при установке этого бита на вход АЦП подается фиксированное напряжение 1.22+-0.05В, когда бит сброшен, вход АЦП подключается к одному из внешних входов согласно установке битов MUX2..MUX0.

Биты 5..3 - зарезервированы. В AT90S2333/4433 эти биты зарезервиро­ваны и при записи должны устанавливаться в 0.

Биты 2..0 - MUX2..MUX0 - выбор аналогового входа. Содержимое этих бит определяет какой из аналоговых входов подключен на вход АЦП.

**РЕГИСТР УПРАВЛЕНИЯ И СОСТОЯНИЯ АЦП – ADCSR**

**ADCSR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **06h(26h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **ADEN** | **ADSC** | **ADFR** | **ADIF** | **ADIE** | **ADPS2** | **ADPS1** | **ADPS0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Бит 7 - ADEN - Разрешение АЦП. - При записи логической "1" в этот бит разрешается работа АЦП. При установке бита в "0" АЦП выключается. При выключении АЦП до окончания преобразования, преобразование не за­вершается.

Бит 6 - ADSC - Запуск преобразования АЦП. - В режиме одиночного преобразования для запуска преобразования в этот бит должна быть запи­сана "1". При запуске преобразования битом ADSC время преобразования отчитывается ль начала следующего периода частоты на выходе делителя. При установке ADSC после разрешения АЦП или при одновременном разреше­нии АЦП и установке ADSC, первому преобразованию предшествует "пустой" цикл преобразования. Во время этого цикла происходит инициализация АЦП. Во время преобразования бит ADSC остается установленным и сбрасы­вается при завершении преобразования, но до того, как результат пере­писывается в регистр данных АЦП. Это позволяет запустить новое преоб­разование до того как завершится текущее. Новое преобразование будет запущено сразу по окончанию текущего. Если преобразованию предшествует "пустой" цикл, бит ADSC остается установленным пока не будет завершено реальное преобразование. Запись 0 в этот бит не дает результата.

Бит 5 - ADFR - Выбор непрерывного преобразования АЦП. - Когда этот бит установлен, включается режим непрерывного преобразования АЦП. В этом режиме АЦП непрерывно производит выборки сигнала и обновляет ре­гистр данных. При сбросе этого бита режим непрерывного преобразования отключается.

Бит 4 - ADIF - Флаг прерывания АЦП. - Этот флаг устанавливается ко­гда завершается цикл преобразования АЦП и обновляется регистр данных АЦП. Если установлены флаг глобального разрешения прерываний (I в

SREG) и бит ADIE, по завершению преобразования выполняется прерывание.

Флаг ADIF сбрасывается аппаратно при выполнении соответствующего преры­вания. Другой способ сбросить флаг - записать в него "1". Необходимо предостеречь, что при чтении-модификации-записи ADCSR может быть за­прещено отложенное прерывание. Это же касается и команд SBI и CBI.

Бит 3 - ADIE - Разрешение прерывания от АЦП. - При установке этого бита и бита I в регистре SREG разрешены прерывания по окончанию преоб­разования АЦП.

Биты 2..0 - ADPS2..ADPS0 - биты установки предварительного делите­ля. - Эти биты задают коэффициент деления тактовой частоты процессора и задают тактовую частоту работы АЦП.

Таблица 20. Выбор коэффициента деления АЦП

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ADPS2 | ADPS1 | ADPS0 | Коэф.деления | ADPS2 | ADPS1 | ADPS0 | Коэф.деления |
| **0** | **0** | **0** | 2 | **1** | **0** | **0** | 16 |
| **0** | **0** | **1** | 2 | **1** | **0** | **1** | 32 |
| **0** | **1** | **0** | 4 | **1** | **1** | **0** | 64 |
| **0** | **1** | **1** | 8 | **1** | **1** | **1** | 128 |

**РЕГИСТР ДАННЫХ АЦП - ADCL И ADCH**

**ADCH**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **05h(25h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **-** | **-** | **-** | **MSB** |  |  |
| **R** | **R** | **R** | **R** | **R** | **R** | **R** | **R** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**ADCL**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **04h(24h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | **LSB** |
| **R** | **R** | **R** | **R** | **R** | **R** | **R** | **R** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

**Сканирование аналоговых каналов**

Поскольку смена аналоговых каналов всегда происходит после завер­шения преобразования, для переключения каналов преобразователя можно использовать режим непрерывного преобразования. Обычно для переключе­ния каналов используется прерывание по завершению преобразования. Од­нако пользователь должен принимать во внимание следующие факторы:

- в режиме непрерывного преобразования, цикл выборки-хранения следующего преобразования начинается через полтора тактовых цикла по­сле того как результат текущего преобразования записывается в регистр данных АЦП и устанавливается флаг ADIF. Если канал АЦП переключится до того как произойдет операция выборки-хранения, результат преобразова­ния будет отражать новое состояние мультиплексора, если ADMUX изменит­ся после выборки-хранения, следующий результат будет использовать пре­жнее значение входного канала. Новый подключенный канал будет обслужен в следующем цикле преобразования. При чтении регистра ADMUX всегда возвращается последнее записанное в него число, независимо от того, какой канал используется для текущего преобразования.

**Техника подавления шума АЦП**

Цифровые схемы внутри и снаружи AT90S2333/4433 генерируют элект­ромагнитные излучения, которые могут повлиять на точность аналоговых измерений. Если точность преобразования важна, уровень ума может быть понижен при помощи следующей техники:

1. Аналоговая часть процессора и все аналоговые компоненты уст­ройства должны иметь отдельно разведенную на печатной плате землю. Аналоговая земля должна соединяться с цифровой только в одной точке печа­тной платы.

2. Путь прохождения аналогового сигнала должен быть коротким на­сколько можно. Старайтесь отделять аналоговые дорожки от цифровых ана­логовой землей и проводить их как можно дальше от высокоскоростных ци­фровых сигналов.

3. Вывод AVcc должен подключаться к источнику Vcc через RC цепоч­ку, как показано на рисунке:

4. Для уменьшения шума наводимого процессором используйте функцию подавления шума.

5. Если какие-то выводы порта С используются для вывода цифровых сигналов, они не должны переключаться во время преобразования.

**ХАРАКТЕРИСТИКИ АЦП (T=-40...+85 град.)**

Разрешение 10 бит

Интегральная нелинейность (Vref>2V), 0.2(тип),0.5(макс) ед.мл.разр.

Дифференц. нелинейность (Vref>2V), 0.2(тип),0.5(макс) ед.мл.разр.

Ошибка нуля (смещение) 1(тип) ед.мл.разр.

Время преобразования 65...260 мкС

Тактовая частота 50...200 кГц

Напряжение AVcc Vcc+-0.3В (не больше 6В)

Опорное напряжение Agnd...AVcc

Вх.сопр. по вх. опорн. напр. 6..10(тип)..13 кОм

Вх сопр. аналог. входа 100 (тип) МОм

**ПОРТЫ ВВОДА/ВЫВОДА**

Порт B

Порт B 6-разрядный двунаправленный порт ввода/вывода.

Для обслуживания порта отведено три регистра: регистр данных PORTB (18h, 38h), регистр направления данных - DDRB (17h, 37h) и ножки порта B - PINB (16h, 36h). Адрес ножек порта B 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 B могут поглощать ток до 20 mA и непосредственно управлять светодиодными индикаторами. Если выводы PB0..PB5 используют­ся как входы и замыкаются на землю, если включены внутренние подтяги­вающие резисторы, выводы являются источниками тока (Iil). Дополнитель­ные функции выводов порта B приведены в таблице 20.

Таблица 21. Альтернативные функции выводов порта B

|  |  |
| --- | --- |
| Вывод | Альтернативная функция |
| PB0 | ICP (Вход захвата таймера/счетчика 1) |
| PB1 | OC1 (Выход совпадения таймера,счетчика 1) |
| PB2 | SS (Выбор ведомого шины SPI) |
| PB3 | MOSI (Выход ведущего/ вход ведомого шины SPI) |
| PB4 | MISO (Вход ведущего/ выход ведомого шины SPI) |
| PB5 | SCK (Тактовые импульсы шины SPI) |

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

**РЕГИСТР ДАННЫХ ПОРТА B**

 **PORTB**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **18h(38h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PORTB5** |  |  |  |  | **PORTB0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**РЕГИСТР НАПРАВЛЕНИЯ ДАННЫХ ПОРТА B**

**DDRB**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **17h(37h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **DDB5** |  |  |  |  | **DDB0** |
| **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**ВЫВОДЫ ПОРТА B**

**PINB**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **16h(36h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PINB5** |  |  |  |  | **PINBO** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **Z** | **Z** | **Z** | **Z** | **Z** | **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 6 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 22. Влияние DDBn на выводы порта B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DDBn | PORTBn | Вх/Вых | Подт.резист | Комментарий |
| 0 | 0 | Вход | Нет |  |
| 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 = 5...0 - номер вывода

**АЛЬТЕРНАТИВНЫЕ ФУНКЦИИ PORTB**

SCK - PORTB, Bit 5 - Выход тактовой частоты ведущего и тактовый вход ведомого процессора канала SPI. Если работа SPI разрешена и шина скон­фигурирована как ведомая, этот вывод устанавливается на ввод независи­мо от установки DDB5. Если процессор работает как ведущий, направление передачи данных по этому выводу определяется DDB5. Когда вывод устана­вливается на ввод, подключение подтягивающего резистора состоянием би­та PORTB5. Подробнее см. описание порта SPI.

MISO - PORTB, Bit 4 - Вход данных ведущего, выход ведомого в канале SPI. Если разрешена работа SPI в качестве ведущего, вывод PB4 сконфи­гурирован как вход независимо от установки DDB4. Если SPI разрешен и работает как ведомый, направление передачи данных управляется состоя­нием DDB4. Когда вывод принудительно сконфигурирован как вход, подклю­чение подтягивающего резистора по-прежнему управляется состоянием бита PORTB4. Подробнее см. описание порта SPI.

MOSI - PORTB, Bit 3 - Выход данных ведущего, вход ведомого в канале SPI. Когда SPI работает как ведомый, этот вывод работает как вход не­зависимо от установки бита DDB3. При работе SPI ведомым направление передачи этого вывода управляется битом DDB3. Когда вывод принудитель­но сконфигурирован как вход, подключение подтягивающего резистора по

прежнему управляется состоянием бита PORTB3. Подробнее см. описание

порта SPI.

SS - PORTB, Bit 2. - Вход выбора ведомого. Если канал работает как ведомый, этот вывод работает как вход независимо то установки DDB2.

SPI активируется как ведомый при переводе этого вывода в низкое состо­яние. При работе SPI как ведущего, направление передачи данных через этот вывод управляется битом DDB2. Когда вывод переводится в состояние ввода, подключение подтягивающего резистора управляется битом PORTB2. Подробнее см. описание порта SPI.

OC1 - PORTB, Bit 1. Выход совпадения. Вывод PB1 может работать как внешний выход совпадения таймера/счетчика 1. Для обслуживания этой функции вывод должен быть сконфигурирован как выход (DDB1=1). Разреш­ение работы этой функции рассмотрено при описании таймера. Вывод OC1

также работает как выход при работе таймера в режиме ШИМ.

ICP - PORTB, Bit 0. Вход захвата. Вывод PB0 может работать как вне­шний вход захвата Таймера/счетчика 1. Для обслуживания этой функции вывод должен быть сконфигурирован как вход. Подробнее см. описание ра­боты таймера.

**Порт C**

Порт C 6-разрядный двунаправленный порт ввода/вывода.

Для обслуживания порта отведено три регистра: регистр данных PORTC (15h, 35h), регистр направления данных - DDRC (14h, 34h) и ножки порта C - PINC (13h, 33h). Адрес ножек порта C 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 C могут поглощать ток до 20 mA и непосредственно управлять светодиодными индикаторами. Если выводы PC0..PC5 используют­ся как входы и замыкаются на землю, если включены внутренние подтяги­вающие резисторы, выводы являются источниками тока (Iil). Дополнительные функции порта C - аналоговые входы АЦП. Если неко­торые из выводов порта сконфигурированы как выходы, во время преобра­зования не рекомендуется производить их переключение. В экономичном режиме триггеры Шмитта отключаются от цифровых вхо­дов. Это позволяет удерживать на входах аналоговое напряжение Vcc/2 без заметного увеличения потребления.

**РЕГИСТР ДАННЫХ ПОРТА C – PORTC**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **15h(35h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PORTC5** |  |  |  |  | **PORTC0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**РЕГИСТР НАПРАВЛЕНИЯ ДАННЫХ ПОРТА B**

**DDRC**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **14h(34h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **DDC5** |  |  |  |  | **DDC0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**ВЫВОДЫ ПОРТА B**

**PINC**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13h(33h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PINC5** |  |  |  |  | **PINC0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **Z** | **Z** | **Z** | **Z** | **Z** | **Z** |

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

**ПОРТ C, КАК ПОРТ ВВОДА/ВЫВОДА ОБЩЕГО НАЗНАЧЕНИЯ**

Все 6 бит порта C при использовании для ввода/вывода одинаковы.

Бит DDCn регистра DDRC выбирает направление передачи данных. Если бит установлен (1), вывод сконфигурирован как выход. Если бит сброшен (0) - вывод сконфигурирован как вход. Если PORTCn установлен и вывод сконфигурирован как вход, включается КМОП подтягивающий резистор. Для отключения резистора, PORTCn должен быть сброшен (0) или вывод должен быть сконфигурирован как выход.

Таблица 23. Влияние DDCn на выводы порта C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DDC | PORTC | In\Out | Подтягивающие резисторы | описание |
| **0** | **0** | **0** | Нет | Третье состояние |
| **0** | **1** | **1** | Да | PBn источник тока Iil, если извне соединен с землей |
| **1** | **0** | **0** | Нет | Выход установлен в 0 |
| **1** | **1** | **1** | Нет | Выход установлен в 1 |

**Порт D**

Порт D 8-разрядный двунаправленный порт ввода/вывода.

Для обслуживания порта отведено три регистра: регистр данных PORTD (12h, 32h), регистр направления данных - DDRD (11h, 31h) и ножки порта D - PIND (10h, 30h). Адрес ножек порта D 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 D могут поглощать ток до 20 mA и непосредственно управлять светодиодными индикаторами. Если выводы PD0..PD7 используют­ся как входы и замыкаются на землю, если включены внутренние подтяги­вающие резисторы, выводы являются источниками тока (Iil). Альтернативные функции порта приведены в таблице.

Таблица 24. 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** | AIN0 (неинвертирующ.вход компаратора) |
| **PD7** | AIN1 (инвертирующий вход компаратора) |

**РЕГИСТР ДАННЫХ ПОРТА D – PORTD**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **15h(35h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PORTD5** |  |  |  |  | **PORTD0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **14h(34h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **DDD5** |  |  |  |  | **DDD0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13h(33h)** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **-** | **-** | **PIND5** |  |  |  |  | **PIND0** |
| **R** | **R** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** | **R\W** |
| Начальное значение | **0** | **0** | **Z** | **Z** | **Z** | **Z** | **Z** | **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. 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 25. Влияние DDDn на выводы порта D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DDDn | PORTDn | In\Out | Подтягивающие резисторы | Описание |
| 0 | 0 | Вход | Нет | Третье состояние |
| 0 | 1 | Вход | Да | PDn источник тока Iil, если извне соединен с землей |
| 1 | 0 | Выход | Нет | Выход установлен в 0 |
| 1 | 1 | Выход | Нет | Выход установлен в 1 |

Альтернативные функции порта D

AIN1 - Порт D, бит7 - инвертирующий вход аналогового компаратора. Ес­ли вывод сконфигурирован как вход (DDD7=0) и отключен внутренний под­тягивающий резистор, этот вывод может работать как инвертирующий вход аналогового компаратора. В экономичном режиме триггер Шмитта отключен от цифрового входа. Это позволяет удерживать на входе напряжение близ­кое к Vcc/2 без заметного увеличения потребления.

AIN0 - Порт D, бит6 - неинвертирующий вход аналогового компаратора. Если вывод сконфигурирован как вход (DDD6=0) и отключен внутренний подтягивающий резистор, этот вывод может работать как инвертирующий вход аналогового компаратора. В экономичном режиме триггер Шмитта отключен от цифрового входа. Это позволяет удерживать на входе напря­жение близкое к Vcc/2 без заметного увеличения потребления.

T1 - Порт D, бит 5 - тактовый вход таймера/счетчика 1. Подробнее см. описание таймера.

T0 - Порт D, бит 4 - тактовый вход таймера/счетчика 0. Подробнее см. описание таймера.

INT1 - Порт D, бит 3 - вход внешних прерываний 1. Подробнее см. опи­сание прерываний.

INT0 - Порт D, бит 2 - вход внешних прерываний 0. Подробнее см. опи­сание 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 имеет два бита блокировки, которые могут быть ос­тавлены незапрограммированными (1) или программироваться (0), при этом достигаются свойства приведенные в таблице 20.

Таблица 26. Режимы защиты и биты блокировки

|  |  |
| --- | --- |
| Биты блокировки | Тип защиты |
| Режим | LB1 | LB2 | описание |
| 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 AT90S2333/4433 предусмотрены шесть бит конфигурации - SPIEN, BODLEVEL, BODEN, и CKSEL[2:0].

Когда запрограммирован бит SPIEN (0) разрешен режим

последовательного программирования. По умолчанию бит запрограммирован

(0). В режиме последовательного программирования этот бит недоступен.

BODLEVEL. Задает уровень срабатывания схемы сброса по пропаданию питания. Если бит незапрограммирован (1), этот уровень - 2.7В, для

запрограммированного бита - 4В. По умолчанию бит незапрограммирован.

BODEN. Когда этот бит запрограммирован (0), разрешен сброс по пропаданию питания. По умолчанию бит незапрограммирован.

CKSEL[2:0]. В таблице 4 приведены значения задержки при сбросе на которые влияют эти биты. По умолчанию биты установлены в 010 - 64mS+16K CK.

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 AT90S4433 имеют следующие значения:

1. $000: $1E - код производителя - Atmel

2. $001: $92 - 4 кБ флэш памяти

3. $002: $01 - при $01=$92 - м/сх AT90S4433 для AT90S2333:

1. $000: $1E - код производителя - Atmel

2. $001: $91 - 2 кБ флэш памяти

3. $002: $05 - при $01=$92 - м/сх AT90S2333

**Программирование флэш памяти и EEPROM**

AT90S2333/4433 имеют 2кБ или 4кБ перепрограммируемой флэш памяти программ и 256 байт энергонезависимой памяти данных.

При поставке флэш память и память данных стерты (содержат FFh) и готовы к программированию. Микросхемы поддерживают высоковольтный (12В) параллельный режим программирования и низковольтный режим после­довательного программирования. Напряжение +12В используется только для разрешения программирования, этот вывод почти не потребляет тока. По­следовательный режим программирования предусмотрен для загрузки про­граммы и данных в системе пользователя (внутрисистемное программирова­ние). В обоих режимах программирования память программ и данных програм­мируется байт за байтом. Для программирования EEPROM предусмотрен цикл автоматического стирания при программировании в последовательном режи­ме.

**Параллельное программирование.**

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

Таблица 27. Имена выводов

|  |  |  |  |
| --- | --- | --- | --- |
| Имя сигнала при программирован. | Имявывода | Вх/Вых | Функция |
| RDY/BSY | PD1 | Вых | 0: мкросхема занята программированием1: мкросхема 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 следующей таблице:

Таблица 28. 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 таблице 29.

Таблица 29. Биты конфигурации командного байта

|  |  |
| --- | --- |
| Бит | Значение при установке |
| 7 | Стирание кристалла |
| 6 | Запись битов конфигурации, расположенных в следующих позициях байта данных: D5-SPIEN, D4-BODLEVEL, D3-BODEN, D2-D0-CKSEL (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: D5-SPIEN, D4-BODLEVEL, D3-BODEN, D2-D0 -CKSEL (для BS=1) или D2-LB2, D1-LB1 (для BS=0) |
| 1 | Чтение памяти программ или данных (определяется битом 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...5.5В на выводы питания.

2. Установить RESET и BS в 0 и выдержать не меньше 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 PC(1..0) PB(5..0) 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 раз­решить стирание битов блокировки по окончанию цикла стирания. Подождать около 10 мС до окончания цикла стирания. Стирание кристалла не генерирует сигнала RDY/BSY.

**ПРОГРАММИРОВАНИЕ ФЛЭШ ПАМЯТИ**

Загрузить команду "программирование флэш памяти"

1. Установить XA1,XA0 в '10'. Это разрешает загрузку команды.

2. Установить BS в 0

 3. Установить PC(1..0) PB(5..0) в '0001 0000'. Это команда про­граммирования флэш памяти

4. Подать положительный импульс на XTAL1. При этом загружается команда.

Загрузить младший байт адреса

1. Установить XA1,XA0 в '00'. Это разрешает загрузку адреса.

2. Установить BS в 0. Это выбирает младший байт адреса.

3. Установить на PC(1..0) PB(5..0) младший байт адреса.

4. Подать положительный импульс на XTAL1. При этом загружается младший байт адреса.

Загрузить старший байт адреса

1. Установить XA1,XA0 в '00'. Это разрешает загрузку адреса.

2. Установить BS в 1. Это выбирает старший байт адреса.

3. Установить на PC(1..0) PB(5..0) старший байт адреса. ($00..$01)

4. Подать положительный импульс на XTAL1. При этом загружается старший байт адреса.

Загрузить байт данных

1. Установить XA1,XA0 в '01'. Это разрешает загрузку данных.

2. Установить на PC(1..0) PB(5..0) младший байт данных.

3. Подать положительный импульс на XTAL1. При этом загружается младший байт данных.

Запись младшего байта данных.

1. Установить BS в '0'. Это выбирает младший байт данных

2. Подать на WR отрицательный импульс. Это инициирует программи­рование байта. RDY/BSY переходит в низкое состояние.

3. Перед программированием следующего байта подождать, пока RDY/BSY перейдет в высокое состояние.

Загрузить байт данных

1. Установить XA1,XA0 в '01'. Это разрешает загрузку данных.

2. Установить на PC(1..0) PB(5..0) старший байт данных.

3. Подать положительный импульс на XTAL1. При этом загружается старший байт данных.

Запись старшего байта данных.

1. Установить BS в '1'. Это выбирает старший байт данных

2. Подать на WR отрицательный импульс. Это инициирует программи­рование байта. RDY/BSY переходит в низкое состояние.

3. 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 (00h - 7Fh/FFh).

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/$07)

4. Установить OE в '0', BS в '0'. Теперь на выводах PC(1..0) PB(5..0) можно прочитать байт данных.

5. Установить BS в '1'. Теперь на выводах PB можно прочитать 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 (00h - 7Fh)

3. Установить OE в '0', BS в '0'. Теперь на выводах PC(1..0) PB(5..0) можно прочитать 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 - программировать бит SPIEN, Бит5=1 - стереть бит SPIEN

Бит4=0 - программировать бит BODLEVEL,

Бит4=1 - стереть бит BODLEVEL

Бит3=0 - программировать бит BODEN,

Бит3=1 - стереть бит BODEN

Биты 2-0 =0 - программировать биты CKSEL,

Биты 2-0 =1 - стереть биты CKSEL

3. Подать на WR отрицательный импульс и ждать перехода RDY/BSY в единицу.

**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 - программировать бит LB2

Бит1=0 - программировать бит LB1

3. Подать на WR отрицательный импульс и ждать перехода RDY/BSY в единицу.

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 '0'. Теперь на выводах PC(1..0) PB(5..0) можно прочитать биты конфигурации.

Бит5 - SPIEN - '0' запрограммирован

Бит4 - BODLEVEL - '0' запрограммирован

Бит3 - BODEN - '0' запрограммирован

Биты 2-0 - CKSEL - '0' запрограммированы

3. Установить OE в '0', BS в '1'. Теперь на выводах PC(1..0) PB(5..0) можно прочитать биты блокировки.

Бит2 - LB2 - '0' запрограммирован

Бит1 - LB1 - '0' запрограммирован

3. Установить OE 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 PC(1..0) PB(5..0).

3. Установить OE в "1".

Команду необходимо подавать только перед чтением первого байта.

Таблица 31. Характеристики параллельного программирования

T=21..27 ±C, Vcc=4.5-5.5V

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Параметр | Min | Typ | Max |  |
| tDVXH | Задерж.между установ. данных и сигн.упр. и высоким уровнем XTAL1 | 67 |  |  | nS |
| tXHXL | Ширина импульса XTAL | 67 |  |  | nS |
| tXLDH | Удержание данных и сигналов управленияПосле установки XTAL1 = 1 | 67 |  |  | nS |
| tBVVL | Удержание BS после установки WR | 0 | 67 |  |  | nS |
| tWLWH | Ширина импульса WR | 67 |  |  | nS |
| tWHRL | Задержка между WR | 1 и RDY/BSY | 0 ( | ) |  | 20 |  | nS |
| tXLOL | Задержка между XL | 0 и OE | 0 | 67 |  |  | nS |
| tOLDV | Задержка между XL | 0 и установкой данных |  | 20 |  | nS |
| tWLRH | Задержка между WR | 0 и RDY/BSY | 1 ( | ) | 0.5 | 0.7 | 0.9 | mS |

Примечание: если tWHRL удерживается дольше, чем tWLRH импульс на RDY/BSY не появится.

**ПОСЛЕДОВАТЕЛЬНАЯ ЗАГРУЗКА**

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 FFh. Флэш память программ и энергонезависимая память данных имеют от­дельное адресное пространство: 000h - 3FFh/7FFh для памяти программ и 000h - 07Fh/0FFh 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 программировании нового байта, при чтении из памяти будет читаться значение FFh. После того как микросхема будет готова для записи следующего байта, при чтении можно прочитать записанное значение. Это используется для определения момента, когда можно запи­сывать следующий байт. Этот способ не будет работать для байта FFh, поэтому для записи этого числа перед программированием следующего бай­та придется выждать по крайней мере 4mS. Поскольку после стирания ячейки памяти устанавливаются в FFh, при программировании ячейки, со­держащие FFh можно пропускать. Это не применимо при перезаписи EEPROM без стирания памяти программ. В этом случае подтверждение не работает для данных FFh, для этого значения перед программированием следующего байта необходимо выждать 4mS.

**АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОГО ПРОГРАММИРОВАНИЯ**

Для программирования и проверки AT90S2333/4433 в режиме последо­вательного программирования рекомендуется следующая последовательность действий (см. формат четырех байтовой команды в табл.32):

1. Последовательность включения питания:

Подать напряжение питания между VCC и GND, при этом RESET и SCK должны быть установлены в '0'. Если кварцевый резонатор не подключен к выводам XTAL1 и XTAL2, подайте частоту на вывод XTAL1.(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, и удер­живать его по крайней мере два периода тактовой частоты после

установки вывода SCK в "0".

2. Подождать 20 mS и разрешить последовательное программирование послав команду разрешения на вывод MOSI/PB3.

3. При посылке команды разрешения программирования, второй байт последовательности (53h) будет возвращен при посылке третьего байта. В любом случае, должны быть посланы все четыре байта команды. Если число 53h не получено обратно, подайте положи­тельный импульс на SCK и повторите команду разрешения програм­мирования. Если после 32 попыток не будет получено число 53h, микросхема неисправна.

4. После подачи команды стирания (всегда при программировании па­мяти программ), необходимо подождать 10 mS, выдать положитель­ный импульс на RESET и продолжить с шага 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 FFh не обязательно.

6. Любую ячейку памяти можно проверить используя команду чтения, которая выдает содержимое указанной ячейки на последовательный вывод MISO/PB4.

7. По окончанию программирования вывод RESET может быть установ­лен в '1' для возобновления нормальной работы схемы.

8. Последовательность выключения питания (если необходимо) Установить XTAL1 в '0' (если не используется кварцевый резо­натор)

Установить RESET в '1'.

Отключить питание

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.

Таблица 32. Команды последовательного программирования

|  |  |  |
| --- | --- | --- |
| Команда | Формат команды | Действие |
|  | Байт 1 | Байт 2 | Байт 3 | Байт 4 |  |
| Разрешение рограммир. | 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 aaaa | bbbb bbbb | oooo oooo | Читать старший или младший байт o по адресу ab |
| Запись па­мяти прогр. | 0100 H000 | 0000 aaaa | bbbb bbbb | iiii iiii | Запись старш. или младш. байта i в память прогр по адресу ab |
| Чтение па­мяти данных | 1010 0000 | 0000 000a | bbbb bbbb | oooo oooo | Читать данные o по адресу ab |
| Запись па­мяти данных | 1100 0000 | 0000 000a | bbbb bbbb | iiii iiii | Запись данных i по адресу ab |
| Чтение би­тов блокир. | 0101 1000 | xxxx xxxx | xxxx xxxx | xxxx x21x | Биты 2 и 1 равны 0 – блокировка 1 - нет блокир. |
| Запись би­тов блокир. | 1010 1100 | 111x x21x | xxxx xxxx | xxxx xxxx | Запись битов бло кировки. Устано­вить 1,2=0 для блокировки памяти |
| Читать код устройства | 0011 0000 | xxxx xxxx | xxxx xxbb | oooo oooo | Читать код уст­ройства o с ад­реса b |
| Запись би­тов-перемыч | 1010 1100 | 11\_176543 | xxxx xxxx | xxxx xxxx | Запись битов-пе­ремычек. Устано­вить 3,4,5,6,7 в "0" чтобы запрограммировать |
| Чтение би­тов-перемыч | 1010 0000 | xxxx xxxx | xxxx xxxx | xx87 6543 | Чтение битов-пе­ремычек.0 – запрограммир 1 - стерт |

a - старшие биты адреса; b - младшие биты адреса; o - выход данных

H = '0'-младший байт, '1'-старший байт; i - вход данных;

x - не использ-ся; 1 - 1-й бит блокировки; 2 - 2-й бит блокировки;

3 - CKSEL0; 4 - CKSEL1; 5 - CKSEL2; 6 - BODEN; 7 - BODLEVEL;

8 - SPIEN;

Максимально допустимые параметры

|  |
| --- |
| Рабочая температура ................................. -40±С - +105±СТемпература хранения ................................ -65±С - +150±СНапряжение на любом выводе кроме RESET .............. -1.0V - +7.0VМаксимальное рабочее напряжение ..................... 6.6VПостоянный ток через вывод порта .................... 40.0 mAПостоянный ток между VCC и GND .......................140.0 mA |

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**

Ta = -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 '1'(кроме XTAL1 и RESET) - 0.7Vcc/\_/Vcc+0.5 V

на XTAL1 и RESET - 0.7Vcc/\_/Vcc+0.5 V Выходное напряжение '0'(Порты B,C,D) Il=20mA, Vcc=5V - \_/\_/0.5 V

Il=10mA, Vcc=3V - \_/\_/0.5 V Выходное напряжение '1'(Порты B,C,D) I0h=10mA, Vcc=5V - Vcc-0.5/\_/\_ V

I0h=5mA, Vcc=2.7V - Vcc-0.5/\_/\_ V Подтягивающий резистор сброса - 100/\_/500 кОм

Подтягивающий резистор вывода порта 10/50/100 кОм

Потребляемый ток:

Активный режим, 3V, 4MHz - \_/3.0/\_ mA

холостой ход (idle mode), 3V, 4MHz - \_/750/\_ uA

пониженное потребление WDT включен, 3V - \_/10/\_ uA

WDT выключен, 3V - \_/\_/1 uA

Напряжение смещения аналогового компаратора Vcc=5V - \_/\_/20 mV

Входной ток утечки аналогового компаратора - 1/5/10 nA

Время срабатывания аналогового компаратора Vcc=2.7V - \_/750/\_ nS

Vcc=4.0V - \_/500/\_ nS

**ПРИМЕЧАНИЯ:**

1. В рабочем состоянии ток через выводы должен ограничиваться следующими условиями:

- Максимальный ток через вывод - 20 mA

- Максимальный ток через все выводы - 80mA

При превышении выходного тока выше указанных пределов, напря­жение на выходе может отличаться от приведенного выше. Погло­щение выводами тока больше приведенного не гарантируется.

2. Минимальное напряжение для режима пониженного потребления - 2V

**ПАРАМЕТРЫ ВНЕШНЕГО ТАКТОВОГО СИГНАЛА**

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Vcc=2.7...6.0 V | Vcc=4.0...6.0 V | Ед. |
| min | max | min | max |  |
| Частота | 0 | 4 | 0 | 8 | MHz |
| Период | 250 |  | 125 |  | nS |
| Длительность '1' | 115 |  | 58.3 |  | nS |
| Длительность '0' | 115 |  | 58.3 |  | nS |
| Длительность фронта |  | 10 |  | 4.15 | nS |

**Список регистров.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Адрес регистра** | **название егистра** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| 3Fh (5Fh) | **SREG** | I |  T | H | S |  V | N | Z | C |
| 3Eh (5Eh) |  | ЗАРЕЗЕРВИРОВАН |
| 3Dh (5Dh) | **SP** | SP7  | SP6  | SP5 | SP4  | SP3  | SP2 | SP1 | SP0 |
| 3Ch (5Ch) |  | ЗАРЕЗЕРВИРОВАН |
| 3Bh (5Bh) | **GIMSK** | INT1  | INT0 | **-** | **-** | **-** | **-** | **-** | **-** |
| 3Ah (5Ah) | **GIFR** | INTF1 | INTF0 | **-** | **-** | **-** | **-** | **-** | **-** |
| 39h (59h) | **TIMSK** | TOIE1 | OCIE1 | **-** | **-** | TICE1 | **-** | TOIE0 | **-** |
| 38h (58h) | **TIFR** | TOV1  | OCF1 | **-** | **-** | ICF1 | **-** | TOV0 | **-** |
| 37h (57h) |  | ЗАРЕЗЕРВИРОВАН |
| 36h (56h) |  | ЗАРЕЗЕРВИРОВАН |
| 35h (55h) | **MCUCR** | **-** | **-** |  SE | SM  | ISC11 | ISC10 | ISC01 | ISC00 |
| 34h (54h) | **MCUSR** | **-** | **-** | **-** | **-** | WDRF  | BORF | EXTRF | PORF |
| 33h (53h) | **TCCR0** | **-** | **-** | **-** | **-** | **-** | CS02  | CS01 | CS00 |
| 32h (52h) | **TCNT0** | Таймер / счетчик (8 бит) |
| 31h (51h) |  | ЗАРЕЗЕРВИРОВАН |
| 30h (50h) |  | ЗАРЕЗЕРВИРОВАН |
| 2Fh (4Fh) | **TCCR1A** | COM11  | COM10 | **-** | **-** | **-** | **-** | PWM11 | PWM10 |
| 2Eh (4Eh) | **TCCR1B** | ICNC1  | ICES1 | **-** | **-** | CTC1 | CS12 | CS11 | CS10 |
| 2Dh (4Dh) | **TCNT1H** | Таймер/счетчик1 старший байт  |
| 2Сh (4Сh) | **TCNT1L** | Таймер/счетчик1 младший байт  |
| 2Bh (4Bh) | **OCR1AH** | Таймер/счетчик1 регистр сравнения старший байт  |
| 2Ah (4Ah) | **OCR1AL** | Таймер/счетчик1 регистр сравнения младший байт |
| - |  | ЗАРЕЗЕРВИРОВАН |
| 27h (47h) | **ICR1H** | Таймер/счетчик1 регистр захвата старший байт |
| 26h (46h)  | **ICR1L** | Таймер/счетчик1 регистр захвата младший байт |
| - |  | ЗАРЕЗЕРВИРОВАН |
| 21h (41h) | **WDTCR** | **-** | **-** | **-** | WDTOE  | WDE | WDP2 | WDP1 | WDP0 |
| 20h (40h) |  | ЗАРЕЗЕРВИРОВАН |
| 1Fh (3Fh) |  | ЗАРЕЗЕРВИРОВАН |
| 1Eh (3Eh) | **EEAR** | Регистр адреса энергонезависимой памяти |
| 1Dh (3Dh) | **EEDR** | Регистр данных энергонезависимой памяти |
| 1Ch (3Ch) | **EECR** | **-** | **-** | **-** | **-** | EERIE  | EEMWE | EEWE | EERE |
| - |  | ЗАРЕЗЕРВИРОВАН |
| 18h (38h) | **PORTB** | **-** | **-** | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 |
| 17h (37h) | **DDRB** | **-** | **-** | DDB5  | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 |
| 16h (36h) | **PINB** | **-** | **-** | PINB5  | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 |
| 15h (35h) | **PORTC** | **-** | **-** | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 |
| 14h (34h) | **DDRC** | **-** | **-** | DDC5  | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 |
| 13h (33h) | **PINC** | **-** | **-** | PINC5  | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 |
| 12h (32h) | **PORTD** | **-** | **-** | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 |
| 11h (31h) | **DDRD** | **-** | **-** | DDD5  | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 |
| 10h (30h) | **PIND** | **-** | **-** | PIND5  | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 |
| 0Fh (2Fh) | **SPDR** | Регистр данных SPI |
| 0Eh (2Eh) | **SPSR** | SPIF  | WCOL | **-** | **-** | **-** | **-** | **-** | **-** |
| 0Dh (2Dh) | **SPCR** | SPIE  | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 |
| 0Ch (2Ch) | **UDR** | Регистр данных UART |
| 0Bh (2Bh) | **USR** | RXC  | TXC | UDRE | FE |  OR | **-** | **-** | **-** |
| 0Ah (2Ah) | **UCR** | RXCIE  | TXCIE | UDRIE | RXEN | TXEN | CHR9 | RXB8 | TXB8 |
| 09h (29h) | **UBRR** | Регистр скорости передачи UART |
| 08h (28h) | **ACSR** | ACD  | AINBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 |
| 07h (27h) | **ADMUX** | **-** | ADCBG | **-** | **-** | **-** | MUX2  | MUX1 | MUX0 |
| 06h (26h) | **ADCSR** | ADEN  | ADSC | ADFR | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 |
| 05h (25h) | **ADCH** | **-** | **-** | **-** | **-** | **-** | - | ADC9 | ADC8 |
| 04h (24h) | **ADCL** | ADC7  | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 |
| 03h ( 23h) | **UBRRH** | Регистр скорости передачи UART (старший) |
| - | **-** | ЗАРЕЗЕРВИРОВАН |
| 00h (20h) | **-** | ЗАРЕЗЕРВИРОВАН |

Примечание: Для совместимости с другими устройствами, зарезервирован­ные биты при записи необходимо устанавливать в "0". Данные в зарезервированные ячейки памяти записывать нельзя.

**Набор команд**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Мнемоника | Операнды | Описание | Действие | Флаги | Цкл |
| Арифметические команды |
| 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 константой | Rdh,l<Rdh,l+K  | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Вычесть два регистра | Rd<Rd-Rr  |  |  |
| SUBI | Rd, K | Вычесть константу | Rd<Rd-K |  |  |
| SBIW | Rdl,K | Вычесть слово с константой | Rdh,l<Rdh,l-K |  |  |
| SBC | Rd, Rr | Вычесть с переносом | Rd<Rd-Rr-C |  |  |
| SBCI | Rd, K | Вычесть с переносом | Rd<Rd-K-C |  |  |
| AND | Rd, Rr | Логическое И | Rd<Rd AND Rr |  |  |
| ANDI | Rd, K | Логическое И | Rd<Rd AND K |  |  |
| OR | Rd, Rr | Логическое ИЛИ | Rd<Rd OR Rr |  |  |
| 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<$00 - Rd  | Z,C,N,V,H | 1 |
| SBR | Rd,K | Установ. бита в регистре | Rd<Rd OR K  | Z,N,V  | 1 |
| CBR | Rd,K | Сброс. бита в регистре | Rd<RdAND(FFh-K)  | Z,N,V  | 1 |
| INC | Rd | Увеличить на 1 | Rd<Rd+1 | Z,N,V  | 1 |
| DEC | Rd | Уменьшить на 1 | Rd<Rd-1 | Z,N,V  | 1 |
| TST | Rd | Проверить на 0 или 1 | Rd<Rd AND Rd | Z,N,V | 1 |
| CLR | Rd | Очистить регистр | Rd<Rd XOR Rd | Z,N,V | 1 |
| SER | Rd | Установить регистр | Rd<$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 | Относительный вызов подпрогрограммы | PC<PC+k+1  | None | 3 |
| ICALL |  | Вызов подпр по адресу (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 Rd=Rr | 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 |  |  |
| SBRC | Rr,b | Пропуск если бит в регистре сброшен | if(Rr(b)=0) | None | 1/2 |
| SBRS | Rr,b | Пропуск если бит в регистре установлен  | if(Rr(b)=1) | None | 1/2 |
| SBIC | P, b | Пропуск если бит в регистре I\O сброшен | if(P(b)=0) | None | 1/2 |
| SBIS  | P, b | Пропуск если бит в регистре I\O установлен | if(P(b)=1) | None | 1/2 |
| BRBS  | s, k | Переход если установл флаг s | if(SREG(s)=1) | None | 1/2 |
| BRBC  | s, k | Переход если сброшен флаг s | if(SREG(s)=0) | None | 1/2 |
| BREQ | k | Переход если равно Z=1 | if(Z=1) | None | 1/2 |
| BRNE | k | Переход если неравно if(Z=0)  | if(Z=0) | None | 1/2 |
| BRCS | k | Переход если установл перенос | if(C=1) | None | 1/2 |
| BRCC | k | Переход если сброшен перенос | if(C=0) | None | 1/2 |
| BRSH | k | Переход если равно или больше | if(C=0) | 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 | 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 непосредст. c постинкрем | Rd<(X),X<X+1 | None  | 2 |
| LD | Rd,-X | Загрузить регистр непоср. 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 непоср. c пост инкрем | Rd<(Y),Y<Y+1 | None | 2 |
| LD | Rd,-Y | Загрузить регистр непоср. с предв.декрем | Y<Y-1,Rd<(Y) | None | 2 |
| LDD | Rd,Y+q | Загрузить регистр непоср. со смещением | Rd<(Y+q) | None | 2 |
| LD | Rd,Z | Загрузить регистр непосредственно | Rd<(Z) | None | 2 |
| LD | Rd,Z+ | Загрузить регистр непоср. c пост инкрем | Rd<(Z),Z<Z+1 | None | 2 |
| LD | Rd,-Z | Загрузить регистр непоср. с предв.декрем | Z<Z-1,Rd<(Z) | None | 2 |
| LDD | Rd,Z+q | Загрузить регистр непоср. 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 непоср.c пред.декрем. | X<X-1,(X)<Rr |  |  |
| ST | Y,Rr | Записать регистр непосредственно | (Y)<Rr |  |  |
| ST | Y+,Rr | Записать регистр непоср.c пост инкр | (X)<Rr,X<X+1 |  |  |
| ST | -Y,Rr | Записать регистр непоср.c пред.декрем | Y<Y-1,(Y)<Rr | None | 2 |
| STD | Y+q,Rr | Запис.рег.непоср.сосмещением |  (Y+q)<Rr | None | 2 |
| ST | Z,Rr | Записать регистр непосредственно | (Y)<Rr | None | 2 |
| ST | Z+,Rr | Записать регистр непоср.c пост инкр | (Y)<Rr,Y<Y+1 | None | 2 |
| ST | -Z,Rr | Записать регистр непоср.c пред.декрем | Z<Z-1,(Z)<Rr | None | 2 |
| STD | Y+q,Rr | Запис.рег.непоср.сосмещением | (Y+q)<Rr | None | 2 |
| STS | k,Rr | Записать в ОЗУ | (k)<Rr | None | 3 |
| LPM |  | Загр.из памяти программ | 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 регистре ввода/вывода | I/O(P,b)<1 | None | 2 |
| CBI | P,b | Сбросить бит в регистре 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 | Логичский 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 тетрад | 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 |
| 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 |

**Подготовил Матвеев Дмитрий**

**2003**