ГЛАВА 4

МИКРОКОНТРОЛЛЕРЫ ФИРМЫ ATMEL С АРХИТЕКТУРОЙ AVR

4.1. Особенности семейства AVR

В этой главе мы рассмотрим фирменную RISC-архитектуру AVR компании Atmel на примере 8-разрядного микроконтроллера ATmega103.

Микроконтроллер ATmega103 является старшей моделью семейства AVR фирмы Atmel. Семейство AVR (AT90) удачно воплощает современные тенденции архитектуры RISC-микроконтроллеров, что в сочетании с достижениями фирмы Atmel в области создания flash-памяти сделало его весьма популярным на мировом рынке 8-разрядных микроконтроллеров.

Высокие характеристики семейства AVR обеспечиваются следующими особенностями архитектуры:

- В качестве памяти программ используется внутренняя flash-память. Она организована в виде матрицы 16-разрядных ячеек и может загружаться программатором, либо через порт SPI;
- Система команд включает 120 инструкций. Все инструкции 16 и 32-разрядные;
- 16-разрядные память программ и шина команд вместе с одноуровневым конвейером позволяют выполнить большинство инструкций за один такт синхрогенератора (50 нс при частоте Fosc = 20 МГц);
- память данных имеет 8-разрядную организацию. Младшие 32 адреса пространства занимают регистры общего назначения, далее следуют 64 адреса регистров ввода-вывода, затем внутреннее ОЗУ данных

объемом до 4000 ячеек. Возможно применение внешнего ОЗУ данных объемом до 60 Кбайт;

- внутренняя энергонезависимая память типа EEPROM объемом до 4 Кбайт представляет собой самостоятельную матрицу, обращение к которой осуществляется через специальные регистры ввода/вывода;

Важным преимуществом микроконтроллеров *classic* AVR является их совместимость по функциям выводов с микроконтроллерами архитектуры MCS-51. Это позволяет во многих случаях увеличить производительность имеющейся системы управления посредством замены микроконтроллера, разработки и отладки рабочей программы.

Семейство AVR включает около двух десятков типов 8-разрядных микроконтроллеров трех основных линий:

- ***Tiny AVR*** представляют собой низкостоимостные микроконтроллеры, как правило, в 8-выводном корпусе. Их особенностью является встроенная схема контроля напряжения питания;
- ***Classic AVR*** является основной линией семейства. Быстродействие некоторых моделей достигает 16 MIPS, flash-память программ 2 – 8 Кбайт, EEPROM данных 64 – 512 байт, ОЗУ данных 128 – 512 байт;
- ***Mega AVR*** представляет собой старшую модель, ориентированную на высокопроизводительную работу со сложными задачами, требующими больших ресурсов памяти. Flash ROM программ составляет 16 – 128 Кбайт, EEPROM данных 512 байт, ОЗУ данных 1 – 4 Кбайт. Имеются 10-разрядный АЦП и аналоговый компаратор.

Благодаря универсальности, широкому набору функциональных возможностей, высоким техническим характеристикам микроконтроллеры семейства AVR находят все более широкое применение в системах управления различными объектами. В составе семейства существуют модификации с низковольтным питанием (3 В).

Ниже в таблице приведены параметры некоторых микроконтроллеров семейства AVR. В верхней части таблицы указаны поставляемые микросхемы, а в нижней перспективные.

Все микроконтроллеры семейства AVR имеют общие принципы функционирования, единую систему команд, используют одинаковые методы адресации.

Микроконтроллер ATmega103 является типичным представителем семейства AVR. Знакомство с ним позволит освоить основные методы проектирования, программирования и отладки систем управления на базе современных RISC микроконтроллеров.

Микроконтроллеры семейства AVR

Тип	Flash память программ, Кбайт	ОЗУ, байт	EEPROM, байт	Таймеры	Посл. порты	АЦП, компаратор	Частота, МГц	Корпус
ATiny11 (L)	1	0	0	1	–	комп.	0...6	DIP8 SOIC8
ATiny12 (L)	1	0	64	1	–	комп.	0...8	DIP8 SOIC8
ATiny15 (L)	1	0	64	2 PWM	–	АЦП комп.	1,6	DIP8 SOIC8
ATiny22 (L)	2	128	128	1	–	–	0...8	DIP8 SOIC8
ATiny28 (L)	2	0	0	1	–	–	0...4	DIP8 TQFP32
AT90S1200	1	0	64	1	–	комп.	0...12	DIP20 SOIC20 SSOP20
AT90S2313	2	128	128	2 PWM	UART	комп.	0...10	DIP20 SOIC20
AT90S2323 AT90S2343	2	128	128	1	–	–	0...10	DIP8 SOIC8
AT90S2333	2	128	128	2 2*PWM	SPI UART	комп. АЦП	0...8	DIP28 TQFP32
AT90S4414	4	256	256	2 2*PWM	SPI UART	комп.	0...8	DIP40 PLCC44 TQFP44
AT90S4433	4	128	256	2 2*PWM	SPI UART	АЦП комп.	0...8	DIP28 TQFP32
AT90S4434	4	256	256	3 3*PWM	SPI UART	АЦП комп.	0...8	DIP40 PLCC44 TQFP44
AT90S8515	8	512	512	2 2*PWM	SPI UART	комп.	0...8	DIP40 PLCC44 TQFP44
AT90S8535	8	512	512	3 3*PWM	SPI UART	АЦП комп.	0...8	DIP40 PLCC44 TQFP44
ATmega163(L)	16	1K	512	3 4*PWM	SPI UART	АЦП комп.	0...6	DIP40 PLCC44 TQFP44
ATmega103(L)	128	4K	4K	3 4*PWM	SPI UART	АЦП комп.	0...6	TQFP64
ATmega161(L)	16	1K	512	3 4*PWM	SPI 2 UART	комп.	0...6	DIP40 PLCC44 TQFP44
ATmega83 (L)	8	512	512	3 4*PWM	SPI UART	АЦП, комп.	0...6	DIP40 PLCC44 TQFP44
ATmega128	128	4K	4K	3 PWM	SPI UART	АЦП комп.	0...16	TQFP64

4.2. Структура и функционирование микроконтроллера ATmega103

Структура микроконтроллера включает следующие функциональные блоки:

- 8-разрядное арифметическо-логическое устройство (АЛУ);
- внутреннюю flash-память программ объемом 128 Кбайт с возможностью внутрисистемного программирования через последовательный интерфейс;
- 32 регистра общего назначения;
- внутреннюю EEPROM память данных объемом 4 Кбайт;
- внутреннее ОЗУ данных объемом 4 Кбайт;
- 6 параллельных 8-разрядных портов;
- 3 программируемых таймера-счетчика;
- 10-разрядный 8-канальный АЦП и аналоговый компаратор;
- последовательные интерфейсы UART и SPI;
- блоки прерывания и управления (включая сторожевой таймер).

На рис. 4.1 изображен корпус и приведено назначение выводов микроконтроллера. В скобках указана альтернативная функция вывода, если она существует.

Port A (PA7..PA0). 8-разрядный двунаправленный порт. К выводам порта могут быть подключены встроенные нагрузочные резисторы (отдельно к каждому разряду). Выходные буферы обеспечивают ток 20 мА и способны прямо управлять светодиодным индикатором. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт А при наличии внешней памяти данных используется для организации мультиплексируемой шины адреса/данных.

Port B (PB7..PB0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт В используется также при реализации специальных функций.

Port C (PC7..PC0). Порт С является 8-разрядным выходным портом. Выходные буферы обеспечивают ток 20 мА. Порт С при наличии внешней памяти данных используется для организации шины адреса.

Port D (PD7..PD0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при

подключенных встроенных нагрузочных резисторах. Порт D используется также при реализации специальных функций.

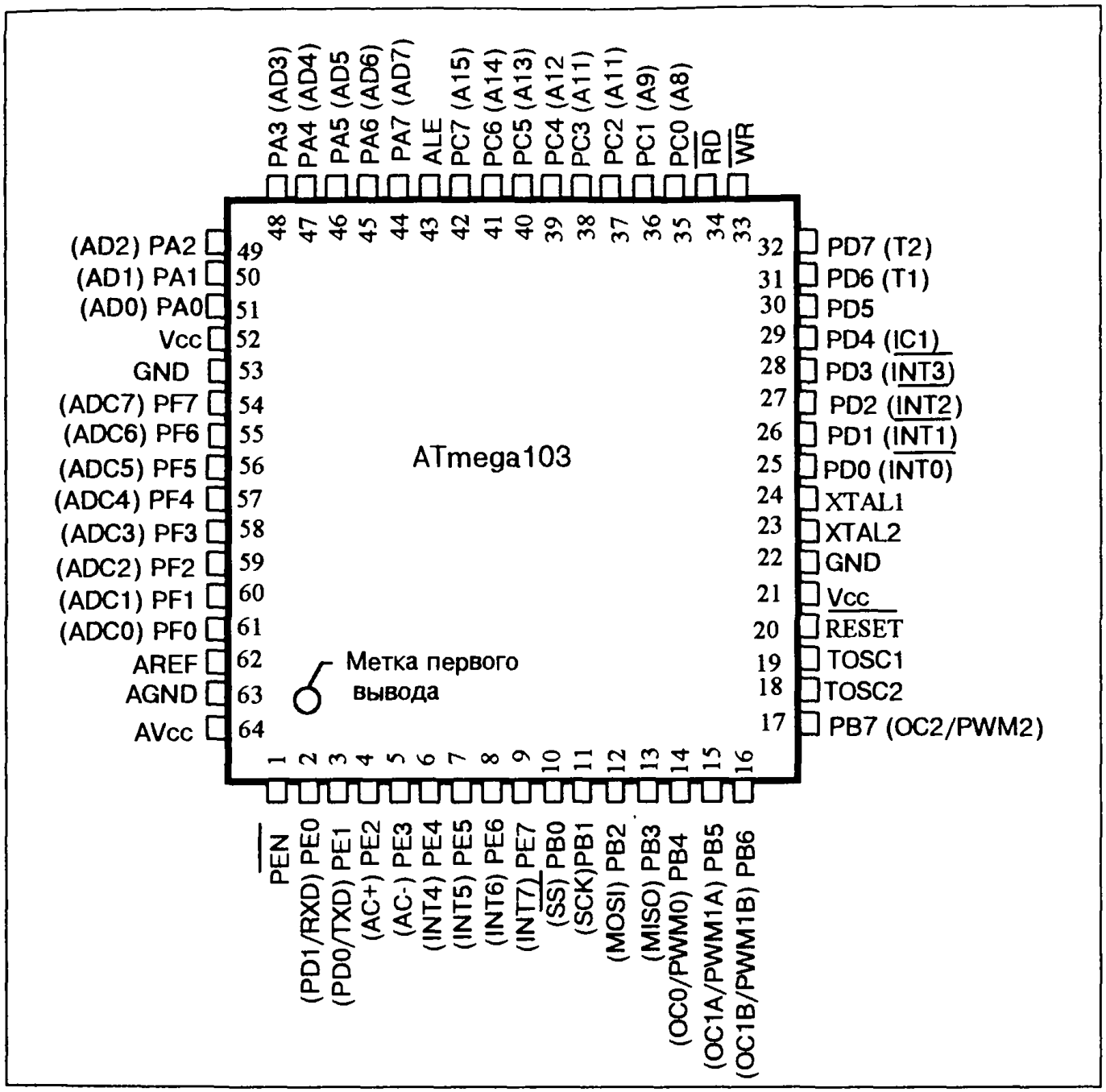


Рис. 4.1. Вид корпуса и назначение выводов микроконтроллера ATmega103

Port E (PE7..PE0). *8-разрядный двунаправленный порт* со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, вытекающий через них ток обеспечивается только при подключенных встроенных нагрузочных резисторах. Порт E используется также при реализации специальных функций.

Port F (PF7..PF0). 8-разрядный входной порт. Входы порта используются также как аналоговые входы аналого-цифрового преобразователя.

RESET. Вход сброса. Для выполнения сброса необходимо удерживать низкий уровень на входе более 50 нс.

XTAL1, XTAL2. Вход и выход инвертирующего усилителя генератора тактовой частоты.

TOSC1, TOSC2. Вход и выход инвертирующего усилителя генератора таймера/счетчика.

WR#, RD#. Стробы записи и чтения внешней памяти данных.

ALE. Строб разрешения фиксации адреса внешней памяти. Строб ALE используется для фиксации младшего байта адреса с выводов AD0 – AD7 в защелке адреса в течение первого цикла обращения. В течение второго цикла обращения выводы AD0 – AD7 используются для передачи данных.

AVCC. Напряжение питания аналого-цифрового преобразователя. Вывод подсоединяется к V_{CC} через низкочастотный фильтр.

AREF. Вход опорного напряжения для аналого-цифрового преобразователя. На этот вывод подается напряжение в диапазоне между AGND и AVCC.

AGND. Это вывод должен быть подсоединен к отдельной аналоговой земле, если она есть на плате. В ином случае вывод подсоединяется к общей земле.

PEN#. Вывод разрешения программирования через последовательный интерфейс. При удержании сигнала на этом выводе на низком уровне после включения питания, прибор переходит в режим программирования по последовательному каналу.

V_{CC}, GND. Напряжение питания и земля.

Микроконтроллеры AVR имеют раздельные пространства адресов памяти программ и данных (гарвардская архитектура). Организация памяти показана на рис. 4.2.

Как видно из рис. 4.2, 32 регистра общего назначения включены в сквозное адресное пространство ОЗУ данных и занимают младшие адреса. Эти регистры на самом деле находятся вне памяти данных, но их включение в единое адресное пространство обеспечивает гибкость при программировании. Файл регистров общего назначения прямо связан с АЛУ, каждый из регистров способен работать как аккумулятор.

Большинство команд выполняются за один такт, при этом из регистров файла могут быть выбраны два операнда, выполнена операция и результат возвращен в регистровый файл. Старшие шесть регистров файла могут использоваться как три 16-разрядных регистра, и выполнять роль, например, указателей при косвенной адресации (рис. 4.3).

Следующие 64 адреса за регистрами общего назначения занимают регистры ввода-вывода. В этой области сгруппированы все регистры дан-

ных, управления и статуса внутренних программируемых блоков ввода-вывода.

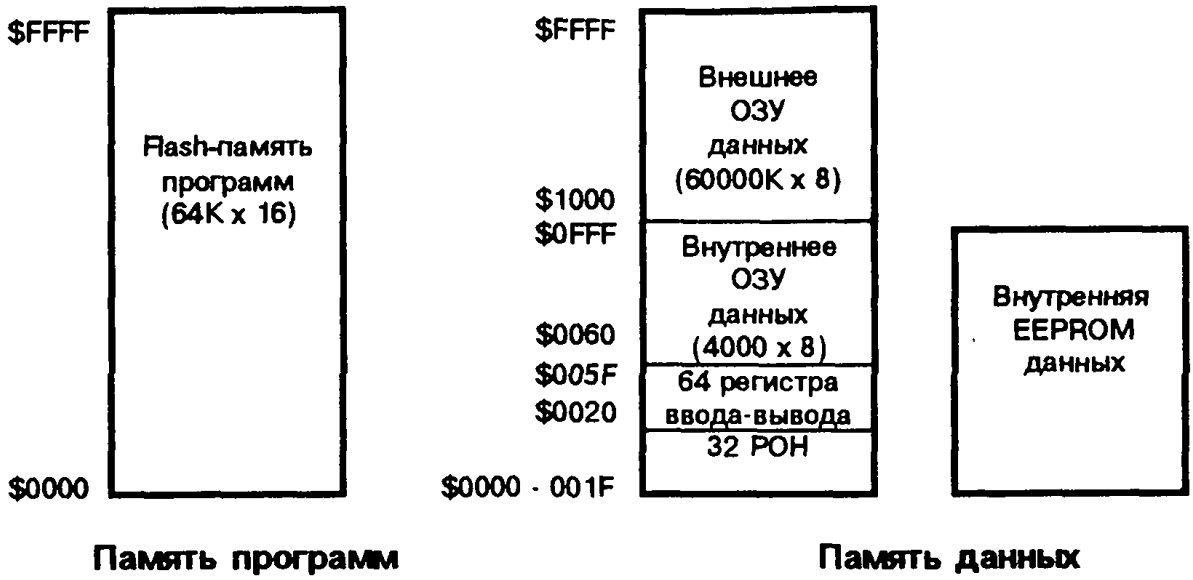


Рис. 4.2. Организация памяти микроконтроллера ATmega103

При использовании команд IN и OUT используются адреса ввода-вывода с $00 по $3F. Но к регистрам ввода-вывода можно обращаться и как к ячейкам внутреннего ОЗУ. При этом к непосредственному адресу ввода-вывода прибавляется $20.

7 0	Адрес	
R31	$1F	Старший байт регистра Z
R30	$1E	Младший байт регистра Z
R29	$1D	Старший байт регистра Y
R28	$1C	Младший байт регистра Y
R27	$1B	Старший байт регистра X
R26	$1A	Младший байт регистра X
...		
R17	$11	
R16	$10	
R15	$0F	
R14	$0E	
R13	$0D	
...		
R2	$02	
R1	$01	
R0	$00	

Рис. 4.3. Регистры общего назначения микроконтроллера ATmega103

Адрес регистра как ячейки ОЗУ приводится далее в круглых скобках. Регистры ввода-вывода с $00 ($20) по $1F ($3F) имеют программно доступные биты. Обращение к ним осуществляется командами SBI и

CBI, а проверка состояния – командами SBIS и SBIC. В следующей таблице приведен список регистров ввода-вывода.

Регистры ввода-вывода микроконтроллера ATmega103

Адрес I/O (адрес SRAM)	Обозначение	Функция
$3F($5F)	SREG	Регистр статуса
$3E($5E)	SPH	Старший байт указателя стека
$3D($5D)	SPL	Младший байт указателя стека
$3C($5C)	XDIV	Регистр управления делением тактовой частоты
$3B($5B)	RAMPZ	Регистр Zвыбора страницы
$3A($5A)	IECR	Регистр управления внешними прерываниями
$39($59)	EIMSK	Регистр масок внешних прерываний
$38($58)	EIFR	Регистр флагов внешних прерываний
$37($57)	TIMSK	Регистр масок прерываний таймеровм/счетчиков
$36($56)	TIFR	Регистр флагов прерывания таймеров/счетчиков
$35($55)	MCUCR	Регистр управления процессора
$34($54)	MCUSR	Регистр статуса процессора
$33($53)	TCCR0	Регистр управления таймером/счетчиком 0
$32($52)	TCNT0	Таймер/счетчик 0
$31($51)	OCR0	Регистр сравнения выхода таймера/счетчика 0
$30($50)	ASSR	Регистр статуса асинхронного режима
$2F($4F)	TCCR1A	Регистр управления A таймера/счетчика 1
$2E($4E)	TCCR1B	Регистр управления B таймера/счетчика 1
$2D($4D)	TCNT1H	Старший байт таймера/счетчика 1
$2C(44C)	TCNT1L	Младший байт таймера/счетчика 1
$2B($4B)	OCR1AH	Старший байт регистра A сравнения выхода таймера/счетчика 1
$2A($4A)	OCR1AL	Младший байт регистра A сравнения выхода таймера/счетчика 1
$29($49)	OCR1BH	Старший байт регистра B сравнения выхода таймера/счетчика 1
$28($48)	OCR1BL	Младший байт регистра B сравнения выхода таймера/счетчика 1
$27($47)	ICR1H	Старший байт регистра захвата таймера/счетчика 1
$26($46)	ICR1L	Младший байт регистра захвата таймера/счетчика 1
$25($45)	TCCR2	Регистр управления таймера/счетчика 2
$24($44)	TCNT2	Таймер/счетчик 2
$23($43)	OCR2	Регистр сравнения выхода таймера/счетчика 2
$21($41)	WDTCR	Регистр управления сторожевого таймера
$1F($3F)	EEARH	Старший байт регистра адреса EEPROM
$1E($3E)	EEARL	Младший байт регистра адреса EEPROM
$1D($3D)	EEDR	Регистр данных EEPROM
$1C($3C)	EECR	Регистр управления EEPROM
$1B($3B)	PORTA	Регистр данных порта A
$1A($3A)	DDRA	Регистр направления данных порта A
$19($39)	PINA	Выводы входов порта A
$18($38)	PORTB	Регистр данных порта B
$17($37)	DDRB	Регистр направления данных порта B
$16($36)	PINB	Выводы входов порта B
$15($35)	PORTC	Регистр данных порта C
$12($32)	PORTD	Регистр данных порта D
$11($31)	DDRD	Регистр направления данных порта D
$10($30)	PIND	Выводы входов порта D

$0F($2F)	SPDR	Регистр данных порта SPI
$0E($2E)	SPSR	Регистр статуса порта SPI
$0D($2D)	SPCR	Регистр управления порта SPI
$0C($2C)	UDR	Регистр данных порта UART
$0B($2B)	USR	Регистр статуса порта UART
$0A($2A)	UCR	Регистр управления порта UART
$09($29)	UBRR	Регистр управления скоростью порта UART
$08($28)	ACSR	Регистр управления и статуса аналогового компаратора
$07($27)	ADMUX	Регистр мультиплексора АЦП
$06($26)	ADCSR	Регистр управления и статуса АЦП
$05($25)	ADCH	Старший байт регистра данных АЦП
$04($24)	ADCL	Младший байт регистра данных АЦП
$03($23)	PORTE	Регистр данных порта E
$02($22)	DDRE	Регистр направления данных порта E
$01($21)	PINE	Выводы входов порта E
$00($20)	PINF	Выводы входов порта F

В пространстве регистров ввода-вывода находятся и регистры управления процессором микроконтроллера: регистр состояния, указатель стека, регистр выбора страницы, регистр управления процессором, регистр управления коэффициентом деления частот. Формат этих регистров следующий:

Регистр состояния SREG

	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	**SREG**
Исх.код	0	0	0	0	0	0	0	0	

SREG.7 – I: Бит разрешения всех прерываний. Для разрешения прерываний этот бит должен быть установлен (=1). Разрешение конкретного прерывания выполняется регистрами маски прерывания EIMSK и TIMSK. Если этот бит очищен (=0), то ни одно из прерываний не обрабатывается. Бит аппаратно очищается после возникновения прерывания и устанавливается (разрешая последующие прерывания) командой RETI.

SREG.6 – T: Бит сохранения копии. Команды копирования бита BLD и BST используют этот бит как источник и приемник при операциях с битами. Командой BST бит регистра общего назначения копируется в бит T, командой BLD бит T копируется в бит регистра общего назначения.

SREG.5 – H: Флаг полупереноса. Флаг полупереноса указывает на перенос между тетрадами при выполнении ряда арифметических операций. Подробности приведены в описании системы команд.

SREG.4 – S: Бит знака. Бит S имеет значение результата операции исключающее ИЛИ (N⊕V) над флагами отрицательного значения (N) и дополнения до двух флага переполнения (V) . Подробности приведены в описании системы команд.

SREG.3 – V: Флаг переполнения. Этот бит поддерживает арифметику дополнения до двух.

SREG.2 – N: Флаг отрицательного значения. Этот флаг указывает на отрицательный результат арифметических и логических операций.

SREG.1 – Z: Флаг нулевого значения. Этот флаг указывает на нулевой результат ряда арифметических и логических операций.

SREG.0 – C: Флаг переноса. Этот флаг указывает на перенос при арифметических и логических операциях.

Указатель стека – SP

	7	6	5	4	3	2	1	0	
$3E ($5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	**SPH**
$3D ($5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	**SPL**
Исх.код	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Микроконтроллеры AVR имеют 16-разрядный указатель стека, размещенный в двух регистрах ввода-вывода. Поскольку микроконтроллеры поддерживают объем ОЗУ до 64 Кбайт, то используются все 16 разрядов указателя стека.

Указатель стека указывает на область в ОЗУ данных, в которой размещается стек подпрограмм и процедур прерываний. Начальный адрес указателя должен задаваться программно перед вызовом подпрограмм и разрешением прерываний. Начальное значение должно быть больше $60. Указатель стека декрементируется на единицу при каждом занесении командой PUSH данных в стек, и на две единицы при занесении в стек адреса при вызове подпрограммы или процедуры прерывания.

Указатель стека инкрементируется на единицу при извлечении данных из стека командой POP, и на две единицы при извлечении адреса из стека при возврате из подпрограммы (RET) или возврате из процедуры прерывания (RETI).

Регистр выбора страницы ОЗУ – RAMPZ

	7	6	5	4	3	2	1	0	
$3B ($5B)	-	-	-	-	-	-	-	RAMPZ0	**RAMPZ**
Исх.код	0	0	0	0	0	0	0	0	

Регистр RAMPZ используется обычно для определения страницы ОЗУ данных, к которой возможно обращение посредством указателя Z. Поскольку микроконтроллеры ATmega103 не поддерживают ОЗУ объемом более 64 К, этот регистр используется только для выбора страницы в памяти программ при использовании команды ELPM. Имеются следующие варианты использования единственного значащего бита RAMPZ0:

- RAMPZ0 = 0: Команде ELPM доступна память программ с адресами от $0000 до $7FFF (младшие 64 Кбайт)
- RAMPZ0 = 1: Команде ELPM доступна память программ с адресами от $8000 до $FFFF (старшие 64 Кбайт).

На команду LPM установки регистра RAMPZ не воздействуют.

Микроконтроллер ATmega603 не содержит регистра RAMPZ и не имеет команды ELPM. Команда LPM способна перекрыть все пространства памяти программ микроконтроллера Atmega603.

Регистр управления процессора – MCUCR

	7	6	5	4	3	2	1	0	
$35 ($55)	SRE	SRW	SE	SM1	SM0	–	–	–	MCUCR
Ис х.код	0	0	0	0	0	0	0	0	

Биты регистра управления MCUCR управляют выполнением основных функций процессора.

MCUCR.7 – SRE: Разрешение внешнего ОЗУ. Установленный (=1) бит SRE разрешает обращение к внешней памяти данных и переводит работу выводов AD0 – 7 (Порт A), A8 – 15 (Порт C), WR и RD на выполнение альтернативной функции. Бит SRE также перенастраивает установки направлений в соответствующих регистрах направления данных. Очистка бита SRE (=0) запрещает обращение к внешней памяти данных и восстанавливает нормальные установки направлений выводов и данных.

MCUCR.6 – SRW: Режим состояния ожидания. При установленном (=1) бите SRW к циклу обращений к внешней памяти данных добавляется один цикл ожидания. При сброшенном (0) бите SRW обращение к внешней памяти выполняется по трехцикловой схеме.

MCUCR.5 – SE: Разрешение режима энергосбережения. Установленный в 1 бит SE разрешает перевод MCU в режим sleep по команде SLEEP. Чтобы исключить перевод MCU в незапланированный режим sleep, рекомендуется устанавливать бит SE непосредственно перед выполнением команды SLEEP.

MCUCR.4,3 – SM1, SM0: Биты выбора режима энергосбережения. Эти биты позволяют выбрать один из трех возможных режимов энергосбережения следующим образом.

SM1	SM0	Режим
0	0	Режим Idle
0	1	Зарезервирован
1	0	Режим Power Down
1	1	Режим Power Save

MCUCR.2..0 – Зарезервированные биты

Регистр управления делением частоты кварцевого генератора – XDIV

	7	6	5	4	3	2	1	0	
$3C ($5C)	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	XDIV
Исх.код	0	0	0	0	0	0	0	0	

Регистр XDIV используется для установки коэффициента деления частоты кварцевого генератора. Эта возможность может быть использована для уменьшения энергопотребления.

XDIV.7 – XDIVEN: Разрешение деления частоты XTAL. При установленном (=1) бите XDIVEN тактовая частота для процессора и всей периферии получается в результате деления частоты Fosc на величину, установленную битами XDIV6 – XDIV0.

XDIV.6..0 – Биты выбора коэффициента деления. Эти биты устанавливают коэффициент деления тактовой частоты Fosc (при установленном бите XDIVEN). Если десятичное значение этих семи битов обозначить через d, то результирующая тактовая частота процессора будет определяться по формуле:

$$F_{CLK} = XTAL/(129 - d)$$

Состояния этих битов можно изменить только при сброшенном бите XDIVEN. При установленном бите XDIVEN, записанное единовременно в биты XDIV6..XDIV0 значение будет определять коэффициент деления. При сбросе бита XDIVEN записанные в биты XDIV6..XDIV0 значения игнорируются.

За регистрами ввода-вывода следуют 4000 адресов внутреннего ОЗУ данных. При использовании внешней памяти данных адресуются оставшиеся 60 Кбайт. Таким образом, при использовании микросхем памяти объемом 64 К будут потеряны 4 Кбайта. При обращении к внешней и внутренней памяти данных используются одни и те же команды, но во втором случае сигналы RD# и WR# не активизируются.

Работа с внешней памятью данных разрешается установкой бита SRE регистра MCUCR. По сравнению с обращением к внутренней памяти, обращение к внешней требует дополнительно одного цикла на каждый байт. Это означает, что для выполнения команд LD, ST, LDS, STS, PUSH и POP требуется дополнительно машинный цикл. Если стек размещен во внешней памяти данных, то прерывания, вызов процедур и возвраты требуют по два дополнительных цикла, поскольку в стеке перемещается содержимое двухбайтного счетчика команд. Если при обмене с внешней памятью данных используется состояние ожидания, то на каждый байт необходимо еще два цикла. Поэтому командам пересылки данных необходимы два дополнительных цикла, тогда как при обработке прерываний, вызове процедур и возвратах требуется на четыре цикла больше, чем указано в описании системы команд.

4.3. Методы адресации и система команд

При создании программы для микроконтроллера на языке Ассемблер разработчик оперирует программно доступными ресурсами микропроцессорной системы. У микроконтроллера ATmega103 эти ресурсы включают: программно доступные регистры микроконтроллера, внутреннюю память данных, внешнюю память данных. Перечисленные ресурсы изображены на рис. 4.2, 4.3.

Каждая команда языка Ассемблер сообщает процессору выполняемую операцию и методы доступа к операндам. Командная строка Ассемблера включает метку (символический адрес), мнемонику (символическое имя) команды, поле операндов, комментарий. Имя команды однозначно связано с выполняемой ею операцией.

Методы адресации представляют собой набор механизмов доступа к операндам. Одни из них просты и поэтому приводят к компактному формату команды и быстрому доступу к операнду, но объем доступных с их помощью ресурсов ограничен. Другие методы адресации позволяют оперировать со всеми имеющимися в системе ресурсами, но команда получается длинной, на ее ввод и выполнение тратится много времени. Набор методов адресации в каждой системе команд является компромиссным сочетанием известных механизмов адресации, выбранных проектировщиками архитектуры исходя из набора решаемых задач.

Заметим, что при двух операндах и приемнике результата имеет место трехадресная команда и каждый адрес формируется с использованием собственного метода адресации. Если адрес операнда или приемника результата не указан в команде, а подразумевается, то имеет место неявная адресация.

На рис. 4.4 – 4.15 приведены схемы методов адресации микроконтроллера ATmega103.

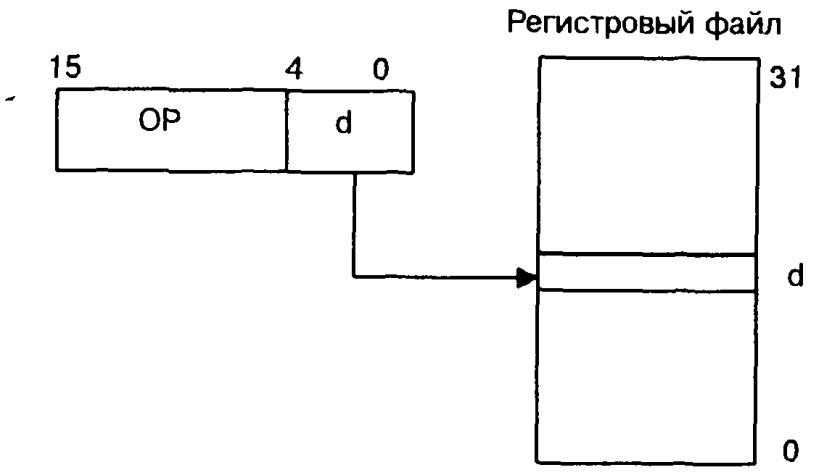


Рис. 4.4. Регистровая адресация (один регистр общего назначения)

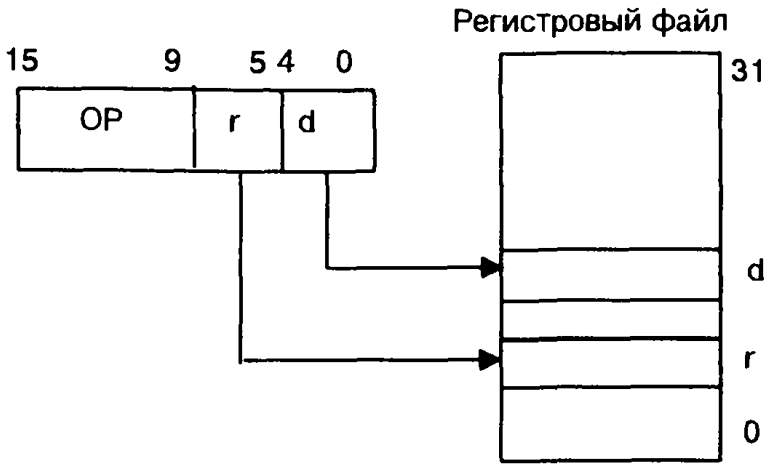


Рис. 4.5. Регистровая адресация (два регистра общего назначения)

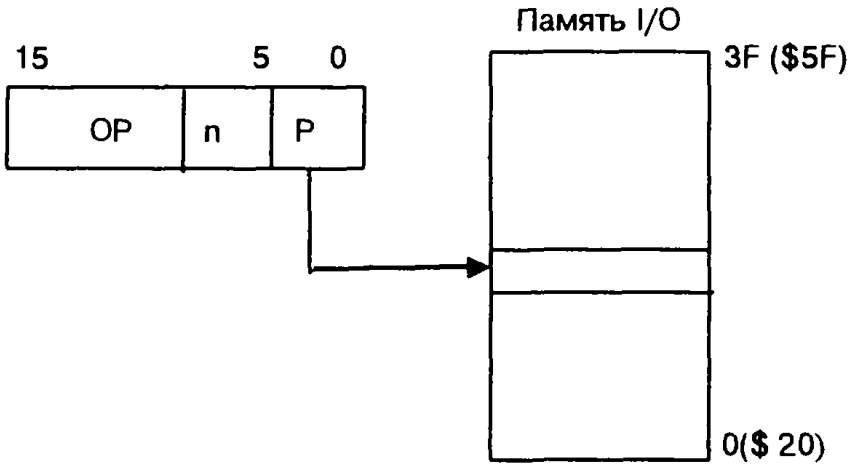


Рис. 4.6. Регистровая адресация (регистры ввода-вывода)

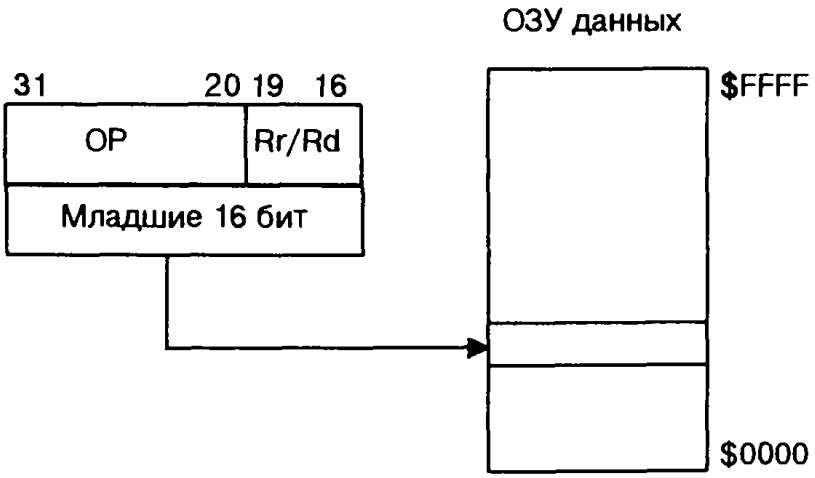


Рис. 4.7. Прямая адресация данных

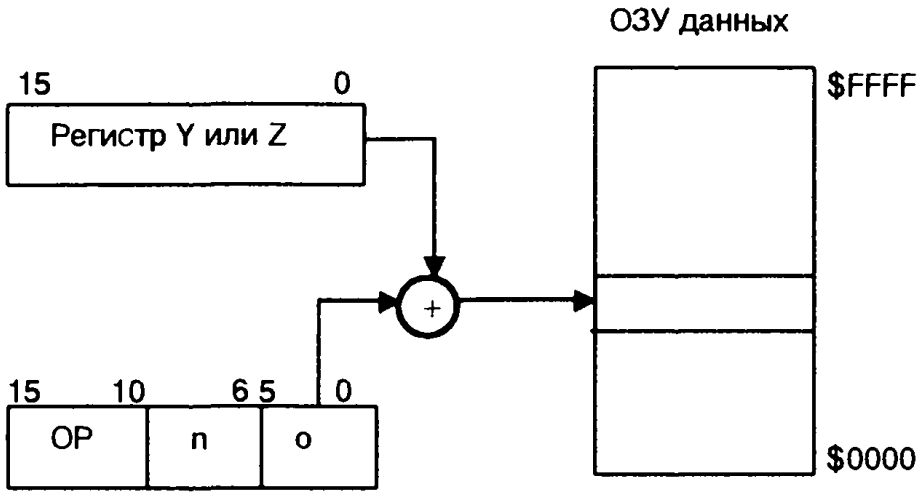


Рис. 4.8. Косвенная адресация данных со смещением

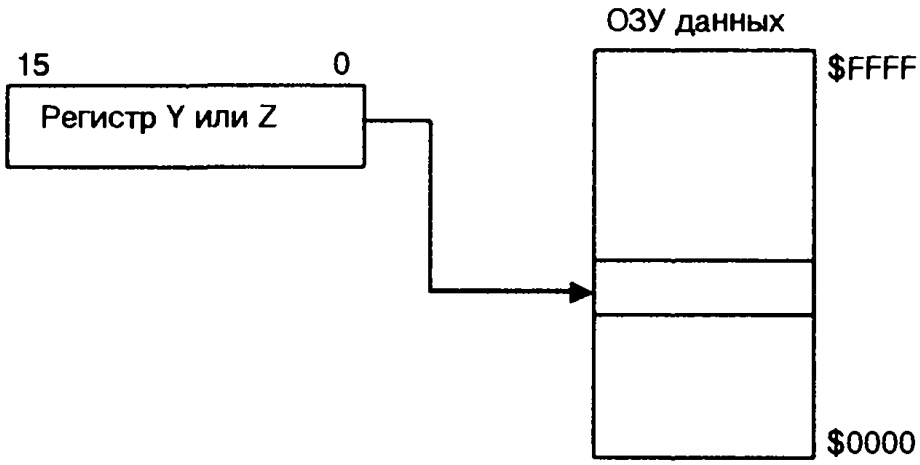


Рис. 4.9. Косвенная адресация данных

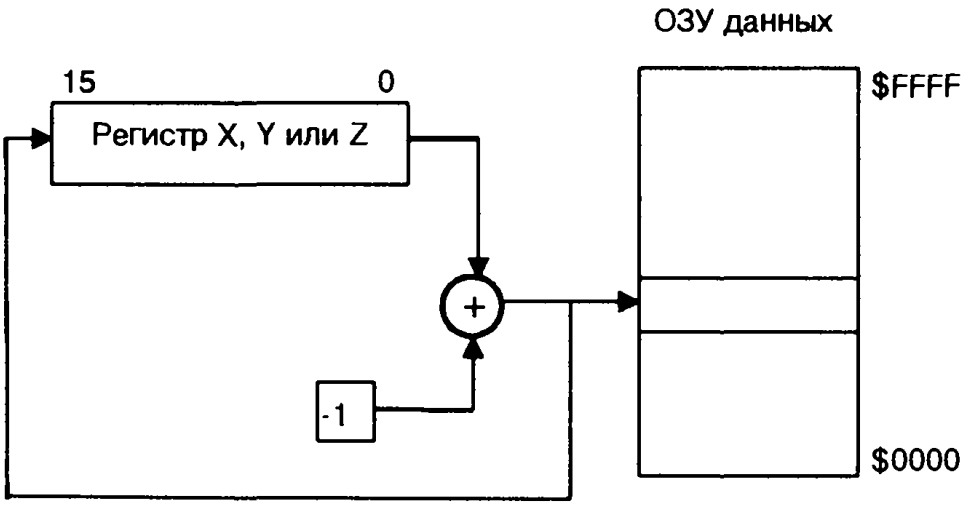


Рис. 4.10. Косвенная адресация данных с преддекрементом

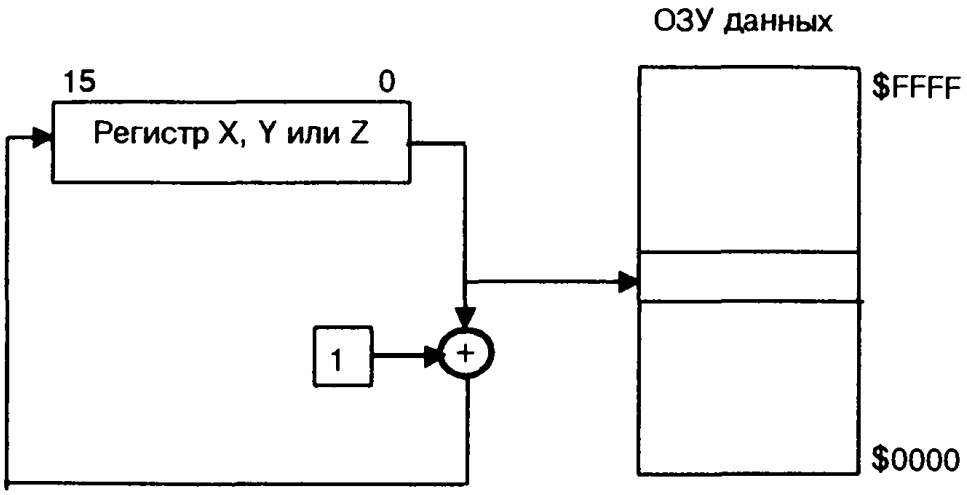


Рис. 4.11. Косвенная адресация данных с постинкрементом

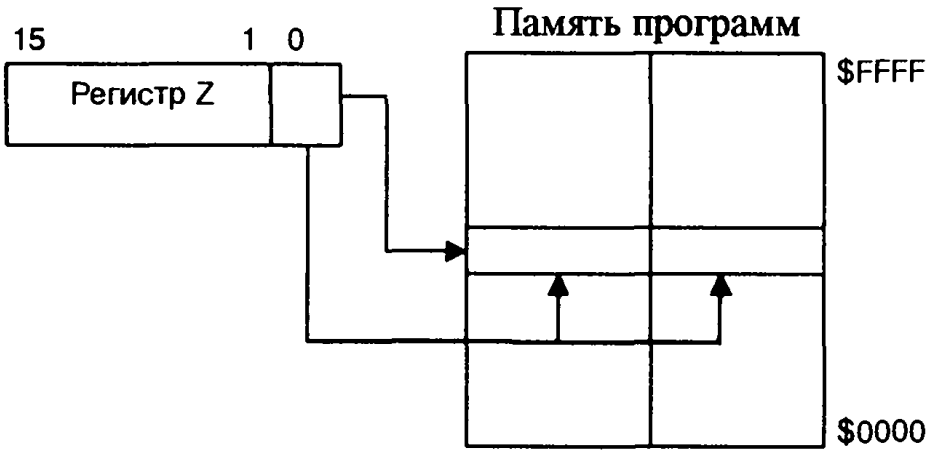


Рис. 4.12. Адресация константы в памяти программ в командах LPM и ELPM

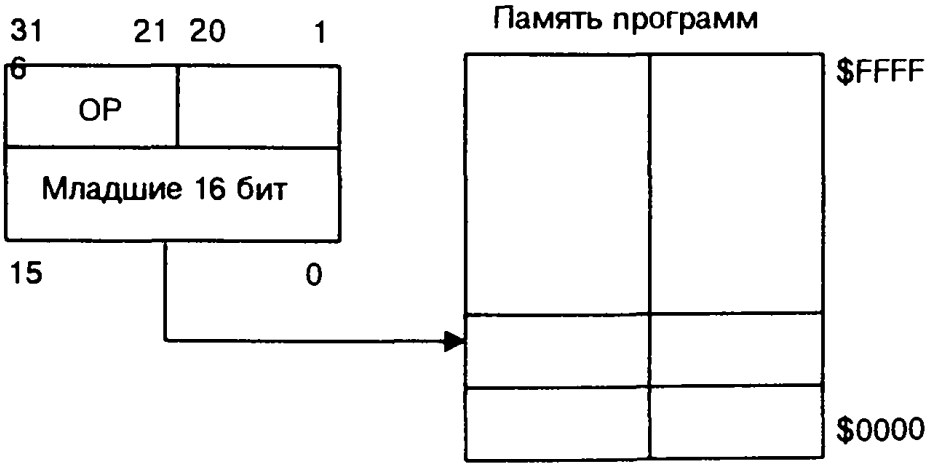


Рис. 4.13. Прямая адресация памяти программ в командах JMP и CALL

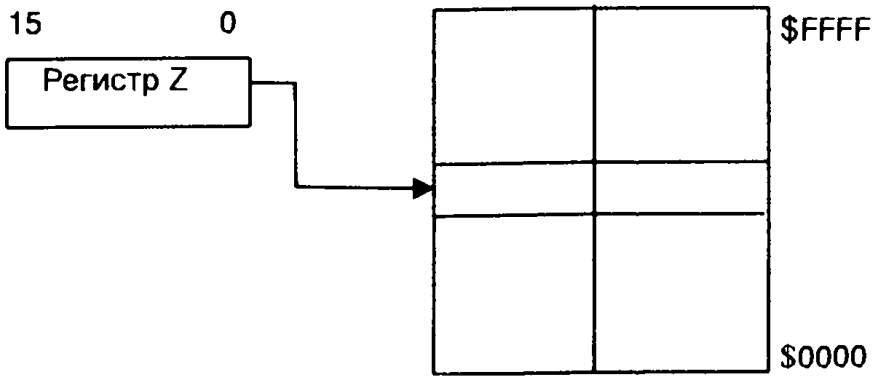


Рис. 4.14. Косвенная адресация памяти программ в командах IJMP и ICALL

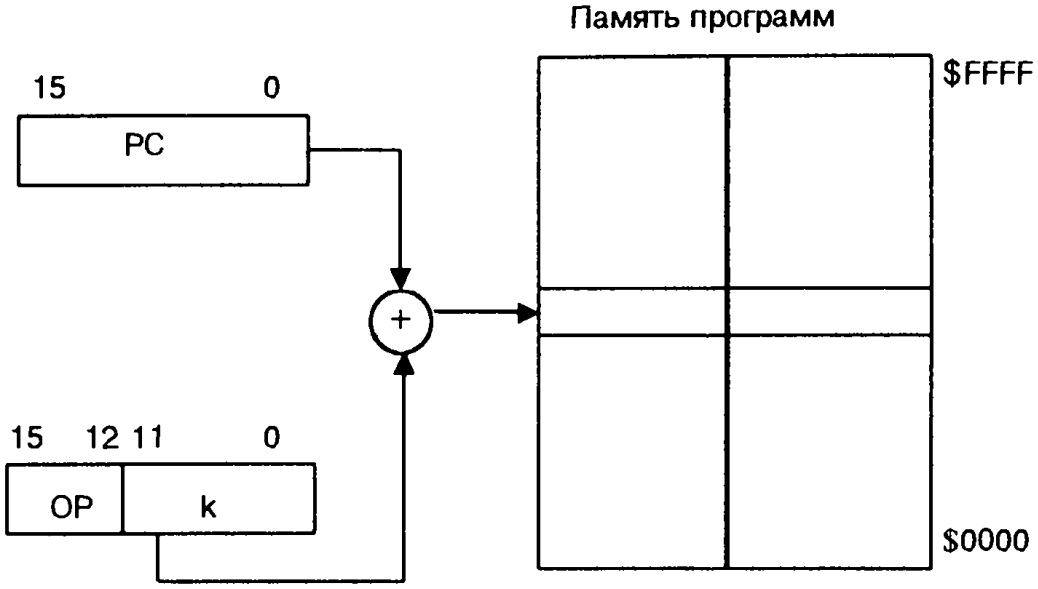


Рис. 4.15. Относительная адресация памяти программ в командах RJMP и RCALL

Команды передачи данных приведены в следующей таблице. Из нее видно, что набор этих команд представляет собой сочетание восьми операций с различными методами адресации.

Команды передачи данных

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
ELPM		Расширенная загрузка из памяти программ в регистр R0	R0 ← (Z+RAMPZ)	Нет	3
MOV	Rd,Rr 0≤d≤31 0≤r≤31	Копировать регистр	Rd ← Rr	Нет	1
LDI	Rd,K 16≤d≤31 0≤k≤255	Загрузить непосредственное значение	Rd ← K	Нет	1
LDS	Rd,k 0≤d≤31 0≤k≤65535	Загрузить из ОЗУ	Rd ← (k)	Нет	3

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
LD	Rd,X 0≤d≤31	Загрузить косвенно	Rd ← (X)	Нет	2
LD	Rd,X+ 0≤d≤31	Загрузить косвенно с постинкрементом	Rd ← (X), X ← X+1	Нет	2
LD	Rd,-X 0≤d≤31	Загрузить косвенно с преддекрементом	X ← X - 1, Rd ← (X)	Нет	2
LD	Rd,Y 0≤d≤31	Загрузить косвенно	Rd ← (Y),	Нет	2
LD	Rd,Y+ 0≤d≤31	Загрузить косвенно с постинкрементом	Rd ← (Y), Y ← Y+1	Нет	2
LD	Rd,-Y 0≤d≤31	Загрузить косвенно с преддекрементом	Y ← Y - 1, Rd ← (Y)	Нет	2
LDD	Rd,Y+q 0≤d≤31 0≤q≤63	Загрузить косвенно со смещением	Rd ← (Y+q)	Нет	2
LD	Rd,Z 0≤d≤31	Загрузить косвенно	Rd ← (Z)	Нет	2
LD	Rd,Z+ 0≤d≤31	Загрузить косвенно с постинкрементом	Rd ← (Z), Z ← Z+1	Нет	2
LD	Rd,-Z 0≤d≤31	Загрузить косвенно с преддекрементом	Z ← Z - 1, Rd ← (Z)	Нет	2
LDD	Rd,Z+q 0≤d≤31 0≤q≤31	Загрузить косвенно со смещением	Rd ← (Z+q)	Нет	2
STS	k,Rr 0≤r≤31 0≤k≤65535	Записать непосредственно в ОЗУ	(k) ← Rr	Нет	3
ST	X,Rr 0≤r≤31	Записать косвенно	(X) ← Rr	Нет	2
ST	X+,Rr 0≤r≤31	Записать косвенно с постинкрементом	(X) ← Rr, X ← X + 1	Нет	2
ST	-X,Rr 0≤r≤31	Записать косвенно с преддекрементом	X ← X - 1, (X) ← Rr	Нет	2
ST	Y,Rr 0≤r≤31	Записать косвенно	(Y) ← Rr	Нет	2
ST	Y+,Rr 0≤r≤31	Записать косвенно с постинкрементом	(Y) ← Rr, Y ← Y + 1	Нет	2
ST	-Y,Rr 0≤r≤31	Записать косвенно с преддекрементом	Y ← Y - 1, (Y) ← Rr	Нет	2
STD	Y+q,Rr 0≤r≤31 0≤q≤63	Записать косвенно со смещением	(Y+q) ← Rr	Нет	2
ST	Z,Rr 0≤r≤31	Записать косвенно	(Z) ← Rr	Нет	2
ST	Z+,Rr 0≤r≤31	Записать косвенно с постинкрементом	(Z) ← Rr, Z ← Z + 1	Нет	2
ST	-Z,Rr 0≤r≤31	Записать косвенно с преддекрементом	Z ← Z - 1, (Z) ← Rr	Нет	2

Мнемо-ника	Операн-ды	Описание	Операция	Флаги	Кол-во циклов
STD	Z+q,Rr 0≤r≤31 0≤q≤63	Записать косвенно со смещением	(Z+q) ← Rr	Нет	2
LPM		Загрузить байт из памяти программ	R0 ← (Z)	Нет	3
IN	Rd,P 0≤d≤31 0≤P≤63	Загрузить данные из порта I/O в регистр	Rd ← P	Нет	1
OUT	P,Rr 0≤r≤31 0≤P≤63	Записать данные из регистра в порт I/O	P ← Rr	Нет	1
PUSH	Rr 0≤r≤31	Сохранить регистр в стеке	STACK ← Rr	Нет	2
POP	Rd 0≤d≤31	Выгрузить регистр из стека	Rd ← STACK	Нет	2

Команды ELPM и LPM выполняют загрузку памяти программ.

Команда MOV копирует содержимое одного регистра общего назначения в другой.

Команда LDI загружает регистр общего назначения непосредственно константой (только R16 – R32), а команды LD выполняют загрузку регистра из ячейки ОЗУ при косвенной адресации, используя в качестве источника адреса регистры X, Y, Z, а также варианты с постинкрементом и преддекрементом источника. Команда LDD выполняет эту же операцию при помощи косвенной адресации со смещением. Команда LDS загружает в регистр общего назначения содержимое прямо адресуемой ячейки памяти данных.

Команда ST выполняют обратную операцию относительно LD – заносит в косвенно адресуемую ячейку памяти данных содержимое регистра общего назначения, используя также варианты с постинкрементом и преддекрементом адреса. Команда STD выполняет эту же операцию при помощи косвенной адресации со смещением, а команда STS прямо адресует ячейку ОЗУ.

Команда IN заносит содержимое регистра ввода-вывода в регистр общего назначения, а команда OUT выполняет обратную операцию.

Команда PUSH сохраняет содержимое регистра в стеке, а команда POP выполняет обратную операцию. При выполнении этих команд кроме программного счетчика изменяется и значение указателя стека SP.

Обратите внимание, что команды пересылки данных, как и рассматриваемые далее команды переходов значения флагов регистра SREG не изменяют.

Команды передачи управления (переходов) приведены ниже.

Команды переходов

Мне-моника	Операн-ды	Описание	Операция	Флаги	Кол-во циклов
RJMP	k -2K≤k≤2K	Перейти относительно	PC ← PC + k + 1	Нет	2
LJMP		Перейти косвенно	PC ← Z	Нет	2
JMP	k 0≤k≤4M	Перейти	PC ← k	Нет	3
RCALL	K -2K≤k≤2K	Вызвать подпрограмму относительно	PC ← PC + k + 1	Нет	3
ICALL		Вызвать подпрограмму косвенно	PC ← Z	Нет	3
CALL	K 0≤k≤64K	Выполнить длинный вызов подпрограммы	PC ← k	Нет	4
RET		Вернуться из подпрограммы	PC ← STACK	Нет	4
RETI		Вернуться из прерывания	PC ← STACK	1	4
CPSE	Rd,Rr 0≤d≤31, 0≤r≤31	Сравнить и пропустить, если равно	if Rd=Rr then PC ← PC + 2 (or 3)	Нет	1/2/3
SBRC	Rr,b 0≤r≤31 0≤b≤7	Пропустить, если бит в регистре очищен	if Rr(b)=0 then PC ← PC + 2 (or 3)	Нет	1/2/3
SBRS	Rr,b 0≤r≤31 0≤b≤7	Пропустить, если бит в регистре установлен	if Rr(b)=1 then PC ← PC + 2 (or 3)	Нет	1/2/3
SBIC	P,b 0≤P≤31 0≤b≤7	Пропустить, если бит в регистре I/O очищен	if I/OP(b)=0 then PC ← PC + 2 (or 3)	Нет	1/2/3
SBIS	P,b 0≤r≤31 0≤b≤7	Пропустить, если бит в регистре I/O установлен	if I/OP(b)=1 then PC ← PC + 2 (or 3)	Нет	1/2/3
BRBS	s,k 0≤s≤7 -64≤k≤+63	Перейти, если бит в регистре статуса установлен	if SREG(s)=1 then PC ← PC + k + 1	Нет	1/2
BRBC	s,k 0≤s≤7 -64≤k≤+63	Перейти, если бит в регистре статуса очищен	if SREG(s)=0 then PC ← PC + k + 1	Нет	1/2
BREQ	k -64≤k≤+63	Перейти, если равно	if Rd=Rr(Z=1) then PC ← PC + k + 1	Нет	1/2
BRNE	k -64≤k≤+63	Перейти, если не равно	if Rd≠Rr(Z=0) then PC ← PC + k + 1	Нет	1/2
BRCS	k -64≤k≤+63	Перейти, если флаг переноса установлен	if C=1 then PC ← PC + k + 1	Нет	1/2
BRCC	k -64≤k≤+63	Перейти, если флаг переноса очищен	if C=0 then PC ← PC + k + 1	Нет	1/2
BRSH	k -64≤k≤+63	Перейти, если равно или больше (без знака)	if Rd≥Rr (C=0) then PC ← PC + k + 1	Нет	1/2

Мне-моника	Операн-ды	Описание	Операция	Флаги	Кол-во циклов
BRLO	k -64≤k≤+63	Перейти, если меньше (без знака)	if Rd<Rr (C=1) then PC ← PC + k + 1	Нет	1/2
BRMI	k -64≤k≤+63	Перейти, если минус	if N=1 then PC ← PC + k + 1	Нет	1/2
BRPL	k -64≤k≤+63	Перейти, если плюс	if N=0 then PC ← PC + k + 1	Нет	1/2
BRGE	k -64≤k≤+63	Перейти, если больше или равно (с учетом знака)	if Rd≥Rr (N⊕V=0) then PC ← PC + k + 1	Нет	1/2
BRLT	k -64≤k≤+63	Перейти, если меньше чем (со знаком)	if Rd<Rr (N⊕V=1) then PC ← PC + k + 1	Нет	1/2
BRHS	k -64≤k≤+63	Перейти, если флаг полупереноса установлен	if H=1 then PC ← PC + k + 1	Нет	1/2
BRHC	k -64≤k≤+63	Перейти, если флаг полупереноса очищен	if H=0 then PC ← PC + k + 1	Нет	1/2
BRTS	k -64≤k≤+63	Перейти, если флаг T установлен	if T=1 then PC ← PC + k + 1	Нет	1/2
BRTC	k -64≤k≤+63	Перейти, если флаг T очищен	if T=0 then PC ← PC + k + 1	Нет	1/2
BRVS	k -64≤k≤+63	Перейти, если флаг переполнения установлен	if V=1 then PC ← PC + k + 1	Нет	1/2
BRVC	k -64≤k≤+63	Перейти, если флаг переполнения очищен	if V=0 then PC ← PC + k + 1	Нет	1/2
BRIE	k -64≤k≤+63	Перейти, если глобальное прерывание разрешено	if I=1 then PC ← PC + k + 1	Нет	1/2
BRID	k -64≤k≤+63	Перейти, если глобальное прерывание запрещено	if I=0 then PC ← PC + k + 1	Нет	1/2

Арифметические и логические команды, команды сдвигов и операций с битами. Для обработки данных микроконтроллер ATmega103 использует группу команд, реализующих арифметические и логические операции, а также сдвиги. Важную и достаточно большую группу у каждого микроконтроллера образуют команды, выполняющие операции над отдельными битами.

Арифметические операции являются операциями над 8-разрядными целыми числами. Для выполнения целочисленных операций над длинными словами служат команды сложения и вычитания с учетом флага переноса. Для выполнения операций над числами с плавающей точкой, реализации тригонометрических функций и т.д. служат библиотеки периода выполнения, входящие в состав интегрированной системы программирования «ТурбоАссемблер-AVR».

Арифметические и логические команды, а также команды сдвигов и операций с битами приведены в следующих двух таблицах.

Арифметические и логические команды

Мнемоника	Операнды	Описание	Операция	Флаги	К-во циклов
ADD	Rd,Rr 0≤d≤31 0≤r≤31	Сложить без переноса	Rd ← Rd + Rr	Z, C, N, V, H	1
ADC	Rd,Rr 0≤d≤31 0≤r≤31	Сложить с переносом	Rd ← Rd + Rr + C	Z, C, N, V, H	1
ADIW	Rd,K d∈{24,26,28,30} 0≤K≤63	Сложить непосредственное значение со словом	Rdh:Rdl ← Rdh:Rdl + K	Z, C, N, V	2
SUB	Rd,Rr 0≤d≤31 0≤r≤31	Вычесть без заема	Rd ← Rd - Rr	Z, C, N, V, H	1
SUBI	Rd, K 16≤d≤31 0≤K≤255	Вычесть непосредственное значение	Rd ← Rd - K	Z, C, N, V, H	1
SBC	Rd, Rr 0≤d≤31 0≤r≤31	Вычесть с заемом	Rd ← Rd - Rr - C	Z, C, N, V, H	1
SBCI	Rd, K 16≤d≤31 0≤K≤255	Вычесть непосредственное значение с заемом	Rd ← Rd - K - C	Z, C, N, V, H	1
SBIW	Rd, K d∈{24,26,28,30} 0≤K≤63	Вычесть непосредственное значение из слова	Rdh:Rdl ← Rdh:Rdl - K	Z, C, N, V	2
AND	Rd, Rr 0≤d≤31 0≤r≤31	Выполнить логическое AND	Rd ← Rd • Rr	Z, N, V	1
ANDI	Rd, K 16≤d≤31 0≤K≤255	Логическое AND	Rd ← Rd • K	Z, N, V	1
OR	Rd, Rr 0≤d≤31 0≤r≤31	Логическое OR	Rd ← Rd ∨ Rr	Z, N, V	1
ORI	Rd, K 16≤d≤31 0≤K≤255	Логическое OR с непосредственным значением	Rd ← Rd ∨ K	Z, N, V	1
EOR	Rd, Rr 0≤d≤31 0≤r≤31	Исключающее OR	Rd ← Rd ⊕ Rr	Z, N, V	1
COM	Rd 0≤d≤31	Дополнение до единицы	Rd ← SFF - Rd	Z, C, N, V	1
NEG	Rd 0≤d≤31	Дополнение до двух	Rd ← S00 - Rd	Z, C, N, V, H	1
SBR	Rd, K 16≤d≤31 0≤K≤255	Установить биты в регистре	Rd ← Rd ∨ K	Z, N, V	1

Мне-моника	Операнды	Описание	Операция	Флаги	К-во циклов
CBR	Rd, K 16≤d≤31 0≤K≤255	Очистить биты в регистре	Rd ← Rd • (SFF - K)	Z, N, V	1
INC	Rd 0≤d≤31	Инкрементировать	Rd ← Rd + 1	Z, N, V	1
DEC	Rd 0≤d≤31	Декрементировать	Rd ← Rd - 1	Z, N, V	1
TST	Rd 0≤d≤31	Проверить на ноль или минус	Rd ← Rd • Rd	Z, N, V	1
CLR	Rd 0≤d≤31	Очистить регистр	Rd ← Rd ⊕ Rd	Z, N, V	1
SER	Rd 16≤d≤31	Установить все биты регистра	Rd ← SFF	нет	1
CP	Rd, Rr 0≤d≤31 0≤r≤31	Сравнить	Rd - Rr	Z, C, N, V, H	1
CPC	Rd, Rr 0≤d≤31 0≤r≤31	Сравнить с учетом переноса	Rd - Rr - C	Z, C, N, V, H	1
CPI	Rd, K 16≤d≤31 0≤K≤255	Сравнить с константой	Rd - K	Z, C, N, V, H	1

Команды сдвигов и операций с битами

Мне-моника	Операнды	Описание	Операция	Флаги	Кол-во циклов
LSL	Rd 0≤d≤31	Логически сдвинуть влево	Rd(n+1) ← Rd(n), Rd(0) ←0, C ← Rd(7)	Z,C,N,V,H	1
LSR	Rd 0≤d≤31	Логически сдвинуть вправо	Rd(n) ← Rd(n+1), Rd(7) ←0, C ← Rd(0)	Z,C,N,V	1
ROL	Rd 0≤d≤31	Сдвинуть влево через перенос	Rd(0) ←C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V,H	1
ROR	Rd 0≤d≤31	Сдвинуть вправо через перенос	Rd(7) ←C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd 0≤d≤31	Арифметически сдвинуть вправо	Rd(n) ← Rd(n+1), C ← Rd(0), n=0...6	Z,C,N,V	1
SWAP	Rd 0≤d≤31	Поменять нибблы местами	Rd(3...0) ↔ Rd(7...4)	Нет	1
BSET	s 0≤s≤7	Установить флаг	SREG(s)← 1	SREG(s)	1
BCLR	s 0≤s≤7	Очистить флаг	SREG(s)← 0	SREG(s)	1
SBI	P,b 0≤P≤31 0≤b≤7	Установить бит в регистре I/O	I/O(P,b)← 1	Нет	2

Мне-моника	Операнды	Описание	Операция	Флаги	Кол-во циклов
CBI	P,b 0≤P≤31 0≤b≤7	Очистить бит в регистре I/O	I/O(P,b)← 0	Нет	2
BST	Rd,b 0≤d≤31 0≤b≤7	Переписать бит из регистра во флаг T	T← Rd(b)	T	1
BLD	Rd,b 0≤d≤31 0≤b≤7	Загрузить T флаг в бит регистра	Rd(b) ← T	Нет	1
SEC		Установить флаг переноса	C← 1	C	1
CLC		Очистить флаг переноса	C← 0	C	1
SEN		Установить флаг отрицательного значения	N← 1	N	1
CLN		Очистить флаг отрицательного значения	N← 0	N	1
SEZ		Установить флаг нулевого значения	Z← 1	Z	1
CLZ		Очистить флаг нулевого значения	Z← 0	Z	1
SEI		Установить флаг глобального прерывания	I← 1	I	1
CLI		Очистить флаг глобального прерывания	I← 0	I	1
SES		Установить флаг знака	S← 1	S	1
CLS		Очистить флаг знака	S← 0	S	1
SEV		Установить флаг переполнения	V← 1	V	1
CLV		Очистить флаг переполнения	V← 0	V	1
SET		Установить флаг T	T← 1	T	1
CLT		Очистить флаг T	T← 0	T	1
SEH		Установить флаг полупереноса	H← 1	H	1
CLH		Очистить флаг полупереноса	H← 0	H	1
NOP		Выполнить холостую команду		Нет	1
SLEEP		Установить режим SLEEP	См. описание команды в Приложении 2	Нет	1
WDR		Сбросить сторожевой таймер	См. описание команды в Приложении 2	Нет	1

4.4. Параллельные порты

У микроконтроллера ATmega103 имеется 6 параллельных портов ввода/вывода.

4.4.1. Порт A

Порт A является 8-разрядным двунаправленным портом ввода/вывода, оснащен встроенными нагрузочными резисторами.

Взаимодействие с портом A осуществляется через три расположенных в пространстве ввода/вывода (памяти данных) регистра: регистр порта данных PORTA, регистр направления данных DDRA и регистр входных данных PINA. Регистр входных данных обеспечивает только возможность чтения, регистры порта данных и направления данных обеспечивают возможность и чтения и записи.

Все выводы порта A оснащены индивидуально подключаемыми встроенными нагрузочными резисторами. Выходные буферы порта A обеспечивают ток до 20 мА, что достаточно для прямого управления светодиодными индикаторами. Если выводы PA0 – PA7 используются в качестве входов и внешним сигналом удерживаются на низком уровне, то вытекающий ток обеспечивается подключением внутренних нагрузочных резисторов.

Выводы порта A могут выполнять альтернативную функцию. При взаимодействии с внешней памятью данных они могут формировать младшие разряды мультиплексированной шины адреса/данных. Альтернативная функция включается установкой бита SRE регистра MCUCR, при этом установки регистра направления данных игнорируются.

Регистр данных порта A – PORTA

	7	6	5	4	3	2	1	0	
$1B ($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Исх.код	0	0	0	0	0	0	0	0	

Регистр направления данных порта A – DDRA

	7	6	5	4	3	2	1	0	
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Исх.код	0	0	0	0	0	0	0	0	

Регистр входных данных порта A – PINA

	7	6	5	4	3	2	1	0	
$19 ($39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Исх.код	0	0	0	0	0	0	0	0	

PINA не является регистром, при обращении по этому адресу читаются значения на выводах порта. При чтении регистра PORTA читается состояние защелок данных порта А.

Все восемь линий порта А при его использовании в качестве цифрового устройства ввода/вывода работают одинаково. Соответствующий бит DDAn определяет направление вывода, а бит PORTAn = 1 при работе вывода в качестве входа подключает нагрузочный резистор. После сброса выводы портов находятся в третьем состоянии.

Воздействие битов DDAn на работу выводов порта А

DDAn	PORTAn	I/O	Нагрузочный резистор	Описание
0	0	Вход	Не подключен	Третье состояние
0	1	Вход	Подключен	При входном низком уровне PAn обеспечивает вытекающий ток
1	0	Выход	Не подключен	Низкий уровень, двухтактный выход
1	1	Выход	Не подключен	Высокий уровень, двухтактный выход

4.4.2. Порт B

Порт В является 8-разрядным двунаправленным портом ввода/вывода, оснащен встроенными нагрузочными резисторами.

Взаимодействие с портом В осуществляется через три расположенных в пространстве ввода/вывода (памяти данных) регистра: регистр порта данных PORTB, регистр направления данных DDRB и регистр входных данных PINB. Регистр входных данных обеспечивает только возможность чтения, регистры порта данных и направления данных обеспечивают возможность и чтения и записи.

Все выводы порта В оснащены индивидуально подключаемыми встроенными нагрузочными резисторами. Выходные буферы порта В обеспечивают ток до 20 мА, что достаточно для прямого управления светодиодными индикаторами. Если выводы PB0 – PB7 используются в качестве входов и внешним сигналом удерживаются на низком уровне, то вытекающий ток обеспечивается подключением внутренних нагрузочных резисторов. Выводы порта В могут выполнять следующие альтернативные функции:

Альтернативные фукции выводов порта В

Вывод порта	Альтернативная функция
PB0	SS - вход выбора ведомого устройства SPI
PB1	SCK - тактовый сигнал порта SPI
PB2	MOSI - линия выход ведущего / вход ведомого порта SPI
PB3	MISO - линия вход ведущего / выход ведомого порта SPI
PB4	OC0A/PWM0A - выход сравнения и PWM таймера/счетчика0
PB5	OC1A/PWM1A - выход А сравнения и PWM таймера/счетчика1
PB6	OC1B/PWM1B - выход В сравнения и PWM таймера/счетчика1
PB7	OC2/PWM2 - выход сравнения и PWM таймера/счетчика2

При использовании выводов для выполнения альтернативных функций регистры DDRB и PORTB должны быть установлены соответствующим образом.

Регистр данных порта B – PORTB

	7	6	5	4	3	2	1	0	
$18 ($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	**PORTB**
Исх.код	0	0	0	0	0	0	0	0	

Регистр направления данных порта B – DDRB

	7	6	5	4	3	2	1	0	
$17 ($37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	**DDRB**
Исх.код	0	0	0	0	0	0	0	0	

Регистр выводов входа порта B – PINB

	7	6	5	4	3	2	1	0	
$16 ($36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	**PINB**
Исх.код	0	0	0	0	0	0	0	0	

PINB не является регистром, обращение по этому адресу обеспечивает чтение логического состояния каждого вывода порта. При чтении регистра PORTB читается состояние защелок данных порта B.

Все восемь выводов порта B при его использовании в качестве цифрового устройства ввода/вывода работают одинаково. Соответствующий бит DDBn определяет направление вывода, а бит PORTBn = 1 при работе вывода в качестве входа подключает нагрузочный резистор. После сброса выводы портов находятся в третьем состоянии.

Воздействие битов DDBn на работу выводов порта B

DDBn	PORTBn	I/O	Нагрузочный резистор	Описание
0	0	Вход	Не подключен	Третье состояние
0	1	Вход	Подключен	При входном низком уровне PBn обеспечивает вытекающий ток
1	0	Выход	Не подключен	Низкий уровень, двухтактный выход
1	1	Выход	Не подключен	Высокий уровень, двухтактный выход

4.4.3. Порт C

Порт C представляет собой 8-разрядный выходной порт. Кроме основной функции выводы порта C выполняют альтернативную функцию при работе с внешней памятью данных. В этом режиме через линии порта C выводится старший байт адреса внешней памяти.

Регистр данных порта C – PORTC

	7	6	5	4	3	2	1	0	
$15 ($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	**PORTC**
Исх.код	0	0	0	0	0	0	0	0	

4.4.4. Порт D

Порт D является 8-разрядным двунаправленным портом ввода/вывода, оснащен встроенными нагрузочными резисторами.

Взаимодействие с портом D осуществляется через три расположенных в пространстве ввода/вывода (памяти данных) регистра: регистр порта данных PORTD, регистр направления данных DDRD и регистр входных данных PIND. Регистр входных данных обеспечивает возможность только чтения, регистры порта данных и направления данных обеспечивают возможность и чтения и записи.

Выходные буферы выводов порта D обеспечивают втекающий ток до 20 мА. Если выводы PD0 – PD7 используются в качестве входов и внешним сигналом удерживаются на низком уровне, то вытекающий ток обеспечивается подключением нагрузочных резисторов. Выводы порта D могут выполнять альтернативные функции, представленные в таблице.

Альтернативные функции выводов порта D

Вывод порта	Дополнительная функция
PD0	INT0 - Вход внешнего прерывания 0
PD1	INT1 - Вход внешнего прерывания 1
PD2	INT2 - Вход внешнего прерывания 2
PD3	INT3 - Вход внешнего прерывания 3
PD4	IC1 - Вход триггера захвата Таймера/счетчика 1
PD6	T1 - Вход тактового сигнала Таймера/счетчика 1
PD7	T2 - Вход тактового сигнала Таймера/счетчика 2

При использовании выводов порта для выполнения альтернативных функций, их функционирование определяется установками регистров DDRD и PORTD.

Регистр данных порта D – PORTD

	7	6	5	4	3	2	1	0	
$12	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	**PORTD**
Исх.код	0	0	0	0	0	0	0	0	

Регистр направления данных порта D – DDRD

	7	6	5	4	3	2	1	0	
$11	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	**DDRD**
Исх.код	0	0	0	0	0	0	0	0	

Регистр выводов входа порта D – PIND

	7	6	5	4	3	2	1	0	
$10	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	**PIND**
Исх.код	0	0	0	0	0	0	0	0	

PIND не является регистром, обращение по этому адресу обеспечивает чтение логического состояния на каждом выводе порта. При чтении регистра PORTD читаются состояния защелок данных порта D.

Все восемь выводов порта D при его использовании в качестве цифрового устройства ввода/вывода работают одинаково. Соответствующий бит DDDn определяет направление вывода, а бит PORTDn = 1 при работе вывода в качестве входа подключает нагрузочный резистор. После сброса выводы портов находятся в третьем состоянии.

Воздействие битов DDDn на работу выводов порта D

DDDn	PORTDn	I/O	Нагрузочный резистор	Описание
0	0	Вход	Не подключен	Третье состояние
0	1	Вход	Подключен	При входном низком уровне PDn обеспечивает вытекающий ток
1	0	Выход	Не подключен	Низкий уровень, двухтактный выход
1	1	Выход	Не подключен	Высокий уровень, двухтактный выход

4.4.5. Порт E

Порт E является 8-разрядным двунаправленным портом ввода/вывода, оснащен встроенными нагрузочными резисторами.

Взаимодействие с портом E осуществляется через три расположенных в пространстве ввода/вывода (памяти данных) регистра: регистр порта данных PORTE, регистр направления данных DDRE и регистр входных данных PINE. Регистр входных данных обеспечивает только возможность чтения, регистры порта данных и направления данных обеспечивают возможность и чтения и записи.

Все выводы порта E оснащены индивидуально подключаемыми встроенными нагрузочными резисторами. Выходные буферы порта E обеспечивают ток до 20 мА. Если выводы PE0 – PE7 используются в качестве входов и внешним сигналом удерживаются на низком уровне, то вытекающий ток обеспечивается подключением внутренних нагрузочных резисторов. Выводы порта E могут выполнять альтернативные функции в соответствии с следующей таблицей.

Альтернативные функции выводов порта E

Вывод порта	Альтернативная функция
PE0	PDI/RXD – Вход данных при программировании flash-памяти или вход приемника UART.
PE1	PDO/TXD – Выход данных при программировании или выход передатчика UART.
PE2	AC+ – Положительный вход аналогового компаратора.
PE3	AC - – Отрицательный вход аналогового компаратора.
PE4	INT4 – Вход внешнего прерывания 4.
PE5	INT5 – Вход внешнего прерывания 5.
PE6	INT6 – Вход внешнего прерывания 6.
PE7	INT7 – Вход внешнего прерывания 7.

При использовании выводов порта для выполнения альтернативных функций регистры DDRE и PORTE должны быть запрограммированы соответствующим образом.

Регистр данных порта E – PORTE

	7	6	5	4	3	2	1	0	
$03 ($23)	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	**PORTE**
Исх.код	0	0	0	0	0	0	0	0	

Регистр направления данных порта E – DDRE

	7	6	5	4	3	2	1	0	
$02 ($22)	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	**DDRE**
Исх.код	0	0	0	0	0	0	0	0	

Регистр входных данных порта E – PINE

	7	6	5	4	3	2	1	0	
$01 ($21)	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	**PINE**
Исх.код	0	0	0	0	0	0	0	0	

PINE - не является регистром, обращение по этому адресу обеспечивает чтение логического состояния каждого вывода порта. При чтении регистра PORTE читается состояние защелок данных порта E.

Все восемь выводов порта E при его использовании в качестве цифрового устройства ввода/вывода работают одинаково. Соответствующий бит DDEn определяет направление вывода, а бит PORTEn = 1 при работе вывода в качестве входа подключает нагрузочный резистор. После сброса выводы портов находятся в третьем состоянии.

Воздействие битов DDEn на работу выводов порта E

DDEn	PORTEn	I/O	Нагрузочный резистор	Описание
0	0	Вход	Не подключен	Третье состояние
0	1	Вход	Подключен	При входном низком уровне PEn обеспечивает вытекающий ток
1	0	Выход	Не подключен	Низкий уровень, двухтактный выход
1	1	Выход	Не подключен	Высокий уровень, двухтактный выход

4.4.6. Порт F

Порт F является 8-разрядным входным портом. В пространстве ввода/вывода этому порту соответствует только регистр PINF.

Выводы порта F подключены через аналоговый мультиплексор к аналого-цифровому преобразователю. Выводы порта F, кроме выполнения функций аналоговых входов, могут быть использованы и в качестве цифровых входов. Это позволяет пользователю одновременно использовать часть выводов порта F в качестве цифровых входов, а оставшуюся часть – в качестве аналоговых входов.

Регистр входных данных порта F – PINF

	7	6	5	4	3	2	1	0	
$00 ($20)	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	**PINF**
Исх.код	0	0	0	0	0	0	0	0	

PINF не является регистром, обращение по этому адресу обеспечивает чтение логического состояния каждого вывода порта.

4.5. Система прерываний

Микроконтроллеры ATmega103 обслуживают 23 источника прерывания. Все прерывания и механизм сброса имеют отдельные векторы, которые образуют таблицу прерываний. Эта таблица располагается по младшим адресам пространства памяти программ. У каждого прерывания существует свой бит разрешения, который должен быть установлен совместно с битом I регистра SREG, чтобы прерывание могло быть обслужено.

Перечень векторов прерываний приведен в таблице. Перечень отражает также уровень приоритета каждого прерывания. Прерывания с младшими адресами имеют больший уровень приоритета: RESET имеет наивысший уровень приоритета, следующим является запрос внешнего прерывания INT0 и т.д.

Микроконтроллеры содержат два 8-разрядных регистра масок прерываний: регистр масок внешних прерываний EIMSK (External Interrupt

Mask) и регистр масок прерываний таймеров/счетчиков TIMSK (Timer/Counter Interrupt Mask). Кроме того, в регистрах управления блоков ввода/вывода имеются и другие биты разрешения и масок.

При возникновении прерывания бит I (разрешения всех прерываний) в регистре SREG очищается и все прерывания запрещаются. Процедура прерывания может установить бит I, чтобы разрешить вложенные прерывания. Команда RETI в конце процедуры обслуживания прерывания устанавливает бит I (=1).

Векторы сброса и прерываний

Вектор №	Адрес	Источник	Условия возникновения прерывания
1	$0000	RESET	Сброс по сигналу Reset, включению питания и сторожевому таймеру
2	$0002	INT0	Запрос внешнего прерывания 0
3	$0004	INT1	Запрос внешнего прерывания 1
4	$0006	INT2	Запрос внешнего прерывания 2
5	$0008	INT3	Запрос внешнего прерывания 3
6	$000A	INT4	Запрос внешнего прерывания 4
7	$000C	INT5	Запрос внешнего прерывания 5
8	$0006	INT6	Запрос внешнего прерывания 6
9	$0010	INT7	Запрос внешнего прерывания 7
10	$0012	TIMER2 COMP	Совпадение при сравнении таймера/счетчика 2
11	$0014	TIMER2 OVF	Переполнение таймера/счетчика 2
12	$0016	TIMER1 CAPT	Захват значения в таймере/счетчике 1
13	$0018	TIMER1 COMPA	Совпадение регистра A и содержимого таймера/счетчика 1
14	$001A	TIMER1 COMPB	Совпадение регистра B и содержимого таймера/счетчика 1
15	$001C	TIMER1 OVF	Переполнение таймера/счетчика 1
16	$001E	TIMER0 COMP	Совпадение при сравнении таймера/счетчика 0
17	$0020	TIMER0 OVF	Переполнение таймера/счетчика 0
18	$0022	SPI, STC	Конец сдвига данных порта SPI
19	$0024	UART, RX	Завершение приема UART
20	$0026	UART, UDRE	Регистр данных UART пуст
21	$0028	UART, TX	Завершение передачи UART
22	$002A	ADC	Завершение АЦП-преобразования
23	$002C	EE READY	Готовность EEPROM
24	$002E	ANALOG COMP	Срабатывание аналогового компаратора

Когда в счетчик команд загружен вектор прерывания, соответствующий флаг, вызвавший прерывание, аппаратно очищается. Некоторые флаги прерываний можно очистить программно записью логической единицы по адресу бита флага.

Если запрос прерывания возник, когда соответствующий бит разрешения очищен, флаг прерывания будет сохранен в установленном состоянии, пока прерывание не будет разрешено или флаг не будет очищен программно.

Если запросы прерываний возникают при сброшенном бите разрешения всех прерываний, флаги прерываний будут сохранены в установленном состоянии, пока все прерывания не будут разрешены и обработаны в порядке приоритетов.

Внешние прерывания по уровню сигнала флага не имеют и условия прерывания имеют место, пока активен внешний сигнал.

В целом, система прерываний микроконтроллеров весьма удобна для использования, флаги прерываний автоматически сбрасываются при принятии запроса на обслуживание, но программно установить флаг прерывания невозможно.

Регистр масок внешних прерываний – EIMSK

	7	6	5	4	3	2	1	0	
$39 ($59)	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Исх.код	0	0	0	0	0	0	0	0	

EIMSK.7..4 – Разрешение внешних прерываний INT7..INT4. При установленных битах INT7-INT4 и установленном бите I регистра SREG разрешаются запросы по соответствующим входам внешних прерываний. Биты управления идентификацией регистра EICR (External Interrupt Control Register) определяет условия возникновения запроса: по нарастающему/спадающему фронту сигнала или по логическому уровню. Активность сигнала по любому из этих выводов вызовет запрос прерывания, даже если вывод определен как выход.

EIMSK.3..0 – Разрешение внешних прерываний INT3..INT0. При установленных битах INT3-INT0 и установленном бите I регистра SREG разрешаются прерывания по соответствующим входам внешних прерываний. Активным всегда является низкий уровень сигнала. Активность сигнала по любому из этих выводов вызовет запрос прерывания, даже если вывод определен как выход. Запрос прерывания по логическому уровню, если он разрешен, будет существовать до тех пор, пока на входе будет низкий уровень сигнала.

Регистр флагов внешних прерываний – EIFR

	7	6	5	4	3	2	1	0	
$38 ($58)	INTF7	INTF6	INTF5	INTF4	–	–	–	–	EIFR
Исх.код	0	0	0	0	0	0	0	0	

EIFR.7..4 – Флаги внешних прерываний INTF7 - INTF4. При идентификации активного сигнала на входах INT7 - INT4 соответствующий флаг прерывания INTF7 - INTF4 устанавливается (=1). Если бит I регистра SREG и соответствующий бит разрешения (INT7 - INT4) регистра EIMSK установлены, то выполняется переход по вектору прерывания.

При этом переходе флаг очищается. Кроме того, флаг можно очистить, записав в него логическую 1.

Внимание! Программно установить флаг внешнего прерывания нельзя!

Регистр управления внешними прерываниями – EICR

	7	6	5	4	3	2	1	0	
$3A ($5A)	ISC71	ISC70	ISC61	ISC60	ISC51-	ISC50	ISC41	ISC40	EICR
Исх.код	0	0	0	0	0	0	0	0	

EICR.7..0 – ISCX1, ISCX0: Биты управления идентификацией внешних прерываний INT7-INT4. Запросы внешних прерываний на выводах INT7 – INT4 идентифицируются по значениям ISCx1/ISCx0 в соответствии со следующей таблицей:

Управление идентификацией прерываний

ISCx1	ISCx0	Описание
0	0	Запрос прерывания идентифицируется по низкому уровню на INTx
0	1	Зарезервирован
1	0	Запрос прерывания идентифицируется по спадающему фронту на INTx
1	1	Запрос прерывания идентифицируется по нарастающему фронту на INTx

x = 7 .. 4

При изменении значений битов ISCx1/ISCx0 прерывание должно быть запрещено очисткой бита разрешения в регистре GIMSK. Иначе может произойти прерывание в момент изменения значения битов.

Входы прерываний INTx периодически опрашиваются на наличие запроса. Если внешний запрос прерывания фиксируется по фронту, то для надежной работы длительность импульса должна быть больше, чем период частоты синхронизации процессора. Заметим, что частота процессора может быть меньше частоты XTAL из-за наличия делителя. Запрос прерывания по низкому логическому уровню должен продолжаться, пока выполняется текущая инструкция, после которой он будет зафиксирован. Зарос прерывания по логическому уровню, если он разрешен, будет генерировать прерывания до тех пор, пока на входе удерживается низкий уровень.

Регистр масок прерываний таймеров/счетчиков – TIMSK

	7	6	5	4	3	2	1	0	
$37 ($57)	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Исх.код	0	0	0	0	0	0	0	0	

TIMSK.7 – OCIE2: Разрешение прерывания по совпадению Таймера 2. При установленном бите OCIE2 и установленном бите I регистра SREG разрешается прерывание по совпадению содержимого регистра сравнения и рабочего регистра Таймера 2.

TIMSK.6 – TOIE2: Разрешение прерывания по переполнению Таймера 2. При установленном бите TOIE2 и установленном бите I регистра SREG разрешается прерывание по переполнению Таймера 2

TIMSK.5 – TICIE1: Разрешение прерывания по захвату Таймера 1. При установленном бите TICIE1 и установленном бите I регистра SREG разрешается прерывание по захвату Таймера 1.

TIMSK.4 – OCIE1A: Разрешение прерывания по совпадению регистра OCR1A с Таймером 1. При установленном бите OCIE1A и установленном бите I регистра SREG разрешается прерывание по совпадению значения в регистре OCR1A со значением в рабочем регистре Таймера1.

TIMSK.3 – OCIE1B: Разрешение прерывания по совпадению регистра OCR1B с Таймером 1. При установленном бите OCIE1B и установленном бите I регистра SREG разрешается прерывание по совпадению значения в регистре OCR1B со значением в рабочем регистре Таймера 1.

TIMSK.2 – TOIE1: Разрешение прерывания по переполнению Таймера 1. При установленном бите TOIE1 и установленном бите I регистра SREG разрешается прерывание по переполнению Таймера 1.

TIMSK.1 – OCIE0: Разрешение прерывания по совпадению Таймера 0. При установленном бите OCIE0 и установленном бите I регистра SREG разрешается прерывание по совпадению содержимого регистра сравнения OCR0 и рабочего регистра Таймера 0.

TIMSK.0 – TOIE0: Разрешение прерывания по переполнению Таймера 0. При установленном бите TOIE0 и установленном бите I регистра SREG разрешается прерывание по переполнению Таймера 0.

Регистр флагов прерываний таймеров/счетчиков – TIFR

	7	6	5	4	3	2	1	0	
$36 ($56)	OCF2	TOV2	ICF1	OCF1A	OCF1B-	TOV1	OCF0	TOV0	TIFR
Исх.код	0	0	0	0	0	0	0	0	

TIFR.7 – OCF2: Флаг совпадения значения Таймера 2. Бит OCF2 устанавливается при совпадении значения Таймера 2 и содержимого регистра OCR2. Бит OCF2 аппаратно очищается при переходе на вектор прерывания. Возможна очистка флага записью в бит логической 1.

TIFR.6 – TOV2: Флаг переполнения Таймера 2. Бит TOV2 устанавливается при переполнении Таймера 2. Он аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1. В режиме PWM этот бит устанавливается при переходе через $00.

TIFR.5 – ICF1: Флаг захвата входа Таймера 1. Бит ICF1 устанавливается в случае захвата по входу и показывает, что значение Таймера 1 послано в регистр захвата ICR1. Бит очищается аппаратно при переходе

на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

TIFR.4 – OCF1A: Флаг A совпадения выхода Таймера 1. Бит OCF1A устанавливается при совпадении значения в Таймере 1 и содержимого регистра OCR1A. Бит OCF1A аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

TIFR.3 – OCF1B: Флаг B совпадения выхода Таймера 1. Бит OCF1B устанавливается при совпадении значения в Таймере 1 с содержимым регистра OCR1B. Бит OCF1B аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

TIFR.2 – TOV1: Флаг переполнения Таймера 1. Бит TOV1 устанавливается при переполнении Таймера 1. Он аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

TIFR.1 – OCF0: Флаг совпадения выхода Таймера 0. Бит OCF0 устанавливается при совпадении значения в Таймере 0 с содержимым регистра OCR0 . Бит OCF0 аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

TIFR.0 – TOV0: Флаг переполнения Таймера 0. Бит TOV0 устанавливается при переполнении Таймера 0. Он аппаратно очищается при переходе на вектор прерывания. Возможна очистка бита записью во флаг логической 1.

4.6. Таймеры-счетчики

Микроконтроллер ATmega103 имеет три счетчика-таймера: два 8-разрядных – T0, T2 и один 16-разрядный – T1. Таймер 0, в дополнение к обычному режиму, может работать в режиме генератора, синхронизированного кварцем с частотой 32768 кГц. Это позволяет использовать его как часы реального времени (Real Time Clock – RTC).

Таймер 0 имеет свой собственный предделитель. Таймеры 1 и 2 используют индивидуальные коэффициенты деления от общего 10-разрядного предварительного делителя.

Предварительные делители таймеров/счетчиков. Предварительный делитель Таймеров/счетчиков 1 и 2 содержит четыре ступени деления: СК/8, СК/64, СК/256 и СК/1024, где СК - тактовый сигнал процессора, (который может быть ниже XTAL из-за делителя). Кроме СК, в качестве источников тактовых сигналов для Таймеров/счетчиков 1 и 2 могут использоваться сигналы от внешних источников, а также нулевой тактовый сигнал (stop).

Исходный сигнал для предделителя Таймера/счетчика 0 обозначен PCK0. Его цепь по умолчанию подключена к основному тактовому сигналу СК. При установке бита AS0 в регистре ASSR Таймер/счетчик 0 будет тактироваться сигналом с вывода TOSC1, что позволяет использовать его в качестве часов реального времени.

8-разрядные таймеры/счетчики T/C0 и T/C2. 8-разрядный Таймер/счетчик 0 получает непосредственно тактовый сигнал PCK0 или его производное значение после прохождения через предварительный делитель.

8-разрядный таймер/счетчик 2 получает тактовый сигнал СК, либо его производное значение после прохождения через предварительный делитель, либо сигнал с внешнего вывода.

Оба таймера/счетчика могут быть остановлены, как это указано в описании регистров управления TCCR0 и TCCR2.

В регистре TIFR хранятся различные флаги состояния (переполнения, совпадения при сравнении и захвата события) таймеров/счетчиков. Управление таймерами/счетчиками осуществляется через регистры TCCR0 и TCCR2. Разрешение и запрещение прерываний производится посредством битов регистра масок прерываний таймеров/счетчиков TIMSK.

При работе Таймера/счетчика 2 с внешним сигналом необходимо, чтобы минимальное время между двумя входящими импульсами было не меньше одного цикла тактового сигнала процессора.

Точность и разрешение 8-разрядных таймеров/счетчиков растет с уменьшением коэффициента предварительного деления. Высокий коэффициент предварительного деления удобно использовать при реализации медленных операций или синхронизации редко происходящих событий.

Оба таймера/счетчика поддерживают две функции сравнения выхода и используют регистры сравнения OCR0 и OCR2 как источники данных, сравниваемых с содержимым регистров таймеров/счетчиков. В качестве дополнительных функции сравнения по выходу можно установить очистку счетчика при совпадении, а также формирование активных сигналов на выводах PB4(OC0/PWM0) и PB7(OC2/PWM2).

Таймеры/счетчики 0 и 2 можно использовать как 8-разрядные широтно-импульсные модуляторы (PWM). В этом режиме таймер/счетчик, совместно с регистром совпадения выхода, работает как автономный ШИМ с центрированными импульсами без ложных выбросов.

Регистры управления таймерами/счетчикамм – TCCR0, TCCR2

	7	6	5	4	3	2	1	0	
$33 ($53)	–	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	**TCCR0**
Исх.код	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
$25 ($45)	–	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	**TCCR2**
Исх.код	0	0	0	0	0	0	0	0	

Bit 7 – Зарезервированный бит

Bit 6 – Разрешение режима широтно-импульсного модулятора. Установленный (=1) бит разрешает режим ШИМ для таймеров/счетчиков 0 или 2.

Bits 5,4 – Режим сравнения выхода, биты 1 и 0. Биты COMn1 и COMn0 управляют состоянием выводов PB4(OC0/PWM0) или PB7(OC2/PWM2) после совпадения по Таймеру 2. Поскольку это альтернативная функция выводов порта, то соответствующий бит направления вывода должен быть установлен в состояние 1. Коды управления приведены в следующей таблице.

Выбор режима сравнения таймеров/счетчиков

COMn1	COMn0	Описание
0	0	Таймер/счетчик отсоединен от выходного вывода OCn/PWMn
0	1	Переключение выходной линии OCn/PWMn
1	0	Очистка выходной линии Ocn/PWMn (установка в состояние 0)
1	1	Установка выходной линии OCn/PWMn (установка в состояние 1)

n = 0 или 2

В режиме ШИМ функции этих битов отличаются, функционирование описано в таблице 4.19.

Bit 3 – Очистить таймер/счетчик при совпадении. При установленном бите CTC0 или CTC2 таймер/счетчик сбрасывается в состояние $00 в течение одного цикла процессора после наступления совпадения. Если бит управления сброшен, то таймер продолжает считать и не используется в процедуре сравнения. Поскольку факт совпадения определяется в цикле процессора, следующем за совпадением, эта функция работает несколько по другому, если коэффициент предварительного деления больше 1. Если коэффициент деления равен 1 и в регистр сравнения занесено значение C, таймер считает при установленном бите CTC0/2 следующим образом:

...| C-2 | C-1 | C| 0 | 1 |

Если установлен коэффициент деления 8, таймер будет считать в следующем порядке:

...|C-2,C-2,C-2,C-2,C-2,C-2,C-2,C-2|C-1,C-1,C-1,C-1,
C-1,C-1,C-1,C-1|C,0,0,0,0,0,0,0,0|1, 1, 1,...

В режиме ШИМ состояние этого бита значения не имеет.

Bits 2,1,0 – Биты выбора тактовой частоты. Эти биты подключают выход определенной ступени предварительного делителя.

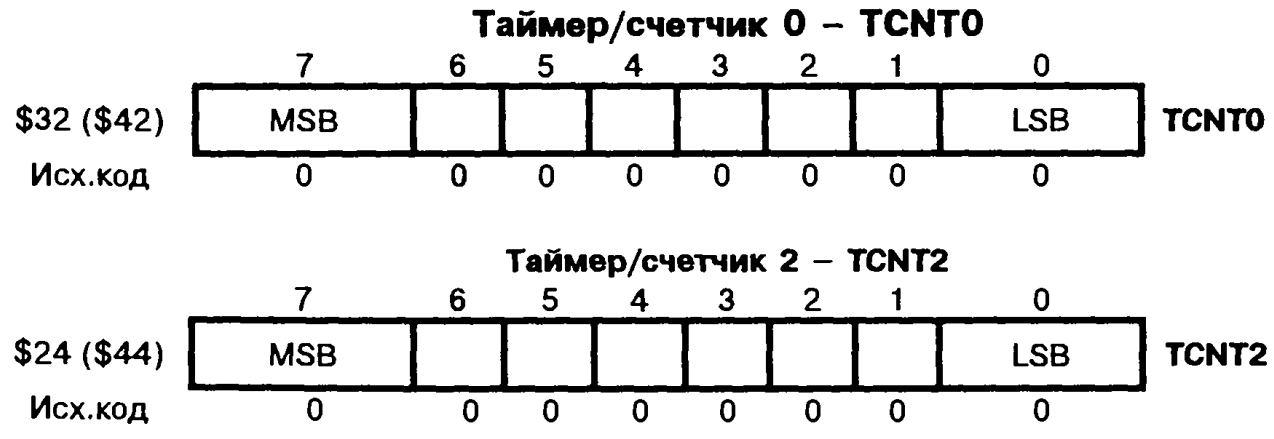
Выбор коэффициента деления частоты для Таймера 0

CS02	CS01	CS00	Описание
0	0	0	Таймер/счетчик 0 остановлен
0	0	1	PCK0
0	1	0	PCK0 / 8
0	1	1	PCK0 / 32
1	0	0	PCK0 / 64
1	0	1	PCK0 / 128
1	1	0	PCK0 / 256
1	1	1	PCK0 / 1024

Выбор коэффициента деления частоты для Таймера 2

CS22	CS21	CS20	Описание
0	0	0	Таймер/счетчик 2 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 PD7(T2), спадающий фронт
1	1	1	Внешний вывод PD7(T2), нарастающий фронт

Условие Stop обеспечивает реализацию функции разрешения/запрещения таймера. Режим деления синхросигнала реализуется непосредственным делением тактовой частоты процессора СК. Если для тактирования Таймера/счетчика 2 используется внешний источник, то переключения на выводе PD7/(T2) будут воздействовать на счетчик, даже если этот вывод определен как выход.



Эти два 8-разрядных регистра являются регистрами счета таймеров/счетчиков. Оба таймера/счетчика работают как инкрементирующие,

либо как реверсивные (в ШИМ режиме) счетчики с возможностью чтения/записи.

Регистр сравнения Таймера/счетчика 0 – OCR0

	7	6	5	4	3	2	1	0	
$31 ($51)	MSB							LSB	OCR0
Исх.код	0	0	0	0	0	0	0	0	

Регистр сравнения Таймера/счетчика 2 – OCR2

	7	6	5	4	3	2	1	0	
$23 ($43)	**MSB**							**LSB**	OCR2
Исх.код	0	0	0	0	0	0	0	0	

Регистры сравнения выходов OCR0 и OCR2 являются 8-разрядными регистрами с возможностью чтения/записи. Выбор функции сравнения определяется битами регистров TCCR0 и TCCR2. Совпадение при сравнении произойдет тогда, когда таймер/счетчик досчитает до значения, содержащегося в OCRx. Программная запись одного и того же значения в таймер/счетчик и в регистр сравнения не приведет к совпадению при сравнении.

Совпадение при сравнении устанавливает флаг прерывания в течение процессорного цикла, следующего за событием.

Таймеры/счетчики 0 и 2 в ШИМ режиме. При установленном режиме ШИМ таймер/счетчик и регистр сравнения выхода (OCR0 или OCR2) формируют 8-разрядный ШИМ-сигнал на выводах PB4(OC0/PWM0) или PB7(OC2/PWM2). Таймер/счетчик работает как прямой/реверсивный счетчик, считающий от $00 до $FF, затем обратно до нуля. После этого начинается новый цикл. Когда состояние счетчика совпадает с содержимым регистра сравнения, значения на выводах PB4(OC0/PWM0) или PB7(OC2/PWM2) устанавливаются в 1 или 0. Значение зависит от установленных в регистрах управления TCCR0 и TCCR2 значений битов COM01/COM00 или COM21/COM20

Выбор режима сравнения в ШИМ режиме

COMn1	COMn0	Значение в регистре и на выводе
0	0	Не подсоединен.
0	1	Не подсоединен.
1	0	Очистка регистра при совпадении, счет по нарастанию. Установка при совпадении, счет по убыванию (неинвертирующий ШИМ).
1	1	Очистка регистра при совпадении, счет по убыванию. Установка при совпадении, счет по нарастанию (инвертирующий ШИМ).

n =0 или 2

Если в регистр OCR заносится $00 или $FF, значение на выходе ШИМ изменится на низкое или высокое, в зависимости от установок битов COM21/COM20 или COM11/COM10.

Состояния ШИМ выходов при OCRn = $00 или $FF

COMn1	COMn0	OCRn	Выход PWMn
1	0	$00	L – низкий уровень
1	0	$FF	H – высокий уровень
1	1	$00	H – высокий уровень
1	1	$FF	L – низкий уровень

n =0 или 2

В режиме ШИМ флаг переполнения таймера (TOV0 или TOV2) устанавливается при смене направления счета при значении $00. Частота ШИМ будет соответствовать тактовой частоте таймера, деленной на 510.

Работа Таймера/счетчика 0 в режиме часов реального времени. Работа в этом режиме (который в фирменной документации называется асинхронным) имеет некоторые особенности.

При переключении между асинхронным и синхронным режимами Таймера/счетчика 0 значения в регистрах TCNT0, OCR0 и TCCR0 могут измениться. Безопасное переключение выполняется следующим образом:

- Запрещаются прерывания OCIE0 и TOIE0 Таймера 0.
- Установкой бита ASO выбирается источник тактового сигнала.
- В регистры TCNT0, OCR0 и TCCR0 записываются новые значения.
- После переключения в асинхронный режим нужно подождать, пока биты TCN0UB, OCR0UB и TCR0UB очистятся.
- Разрешаются прерывания, если необходимо.

Внутренний усилитель/генератор оптимизирован для использования часового кварца (с частотой 32,768 кГц). Внешний тактовый сигнал с вывода TOSC1 проходит через этот же усилитель с полосой пропускания 256 кГц. Таким образом, внешний тактовый сигнал должен иметь частоту в диапазоне от 0 до 256 кГц. Частота внешнего тактового сигнала на выводе TOSC1, не должна превышать одной четверти от тактовой частоты процессора. Тактовая частота процессора может быть ниже частоты XTAL, если разрешено деление частоты.

При записи в один из регистров TCNT0, OCR0 или TCCR0 записываемая величина пересылается в регистр временного хранения и фиксируется после двух положительных фронтов сигнала на TOSC1. Пользователь не должен записывать новое значение прежде, чем содержимое регистра временного хранения не будет передано по назначению. Каждый из указанных регистров имеет свой собственный регистр временного хранения, это означает, к примеру, что содержимое регистра TCNT0 не исказится при записи в регистр OCR0. Для слежения за операцией пересылки

используется регистр статуса асинхронного режима (Asynchronous Status Register – ASSR).

При переходе в режим энергосбережения после записи в регистры TCNT0, OCR0 или TCCR0, пользователь должен ждать обновления записываемого регистра, если для активизации микроконтроллера используется Таймер/счетчик 0. Иначе процессор перейдет в режим Sleep прежде, чем проявятся изменения. Это особенно важно, если для активизации микроконтроллера используется прерывание по сравнению выхода, поскольку сравнение выхода запрещается во время записи в регистры OCR0 и TCNT0. Если цикл записи не завершен (т.е. режим Sleep наступил прежде, чем бит OCR0UB сброшен в 0), прибор никогда не получит совпадения при сравнении и процессор не будет активизирован.

Если Таймер/счетчик 0 используется для активизации микроконтроллера из режима энергосбережения и пользователь намеревается возобновить режим Power Save, то необходимо предпринимать меры предосторожности – для сброса логики прерывания необходим один цикл TOSC1. Если время между активизацией и восстановлением режима Power Save меньше одного цикла TOSC1, прерывания не будет и микроконтроллер не будет активизирован. Если пользователь сомневается в том, что промежуток времени перед восстановлением режима Power Save достаточен, необходимо использовать следующий алгоритм:

- Записать значение в регистр TCCR0, TCNT0 или OCR0.
- Подождать пока соответствующий флаг в регистре ASSR не будет сброшен в 0.
- Ввести режим Power Save.

Генератор частоты 32 кГц Таймера/счетчика 0 работает всегда, за исключением режима Power Down. При восстановления питания или активизации из режима Power Down пользователь должен помнить о том, что генератору для стабилизации необходимо время порядка одной секунды.

При активизации микроконтроллера из режима Power Save при асинхронном режиме Таймера 0, когда условия прерывания выполнены, процесс активизации начинается в следующем цикле тактовой частоты таймера. Состояние таймера должно увеличиться как минимум на единицу, прежде, чем процессор сможет прочитать его состояние. Флаги прерываний обновляются через три тактовых цикла процессора после запуска тактовой частоты процессора. В течение этих циклов процессор выполняет команды, но условия прерывания еще не идентифицируются и процедура обработки прерывания не может быть выполнена.

В асинхронном режиме обработка флагов прерываний асинхронного таймера занимает три тактовых цикла процессора плюс один цикл таймера. Таким образом, содержимое таймера увеличивается минимум на еди-

ницу, прежде чем процессор сможет прочитать содержимое таймера вызвавшее установку флага прерывания. Значение на выходе сравнения меняется в соответствии с тактовым сигналом таймера и не синхронизировано с тактовой частотой процессора.

Регистр статуса асинхронного режима – ASSR

	7	6	5	4	3	2	1	0	
$30 ($50)	–	–	–	–	AS0	TCV0UB	OCR0UB	TCR0UB	**ASSR**
Исх.код	0	0	0	0	0	0	0	0	

ASSR.7..4: – Зарезервированные биты

ASSR.3 – AS0: Асинхронный режим Таймера/счетчика0. При установленном бите AS0 Таймер/счетчик 0 тактируется сигналом с вывода TOSC1. При очищенном бите Таймер/счетчик 0 тактируется внутренним тактовым сигналом СК. При изменении состояния этого бита содержимое TCNT0 может быть повреждено.

ASSR.2 – TCN0UB: Таймер/счетчик0 занят. Бит устанавливается при работе Таймера/счетчика 0 в асинхронном режиме и записи в регистр TCNT0. После обновления значения TCNT0 этот бит аппаратно очищается. Значение TCN0UB = 0 означает, что регистр TCNT0 готов к записи нового значения.

Bit 1 – OCR0UB: Регистр выхода занят. Бит устанавливается при работе Таймера/счетчика0 в асинхронном режиме после записи в регистр OCR0. После обновления значения OCR0 этот бит аппаратно очищается. Значение OCR0UB=0 означает, что регистр OCR0 готов к записи нового значения.

Bit 0 – TCROUB: Регистр управления Таймера/счетчика 0 занят. Бит устанавливается при работе Таймера/счетчика 0 в асинхронном режиме и после записи в регистр TCCR0. После обновления значения TCCR0 этот бит аппаратно очищается. Значение TCROUB = 0 означает, что регистр TCCR0 готов к записи нового значения.

Запись в любой из трех регистров Таймера/счетчика 0 при установленном его флаге занятости может привести к нежелательному прерыванию.

Чтение регистров TCNT0, OCR0 и TCCR0 отличается. При чтении регистра TCNT0 считывается его действительное содержимое, а при чтении регистров OCR0 и TCCR0 читывается содержимое соответствующего регистра временного хранения.

16-разрядный Таймер/счетчик 1. Таймер/счетчик 1 может тактироваться непосредственно сигналом СК, либо СК после делителя, либо сигналом с внешнего вывода. Кроме того, его можно остановить, как указано в описании регистра управления TCCR1B. В регистрах управления TCCR1A и TCCR1B находятся флаги, указывающие на переполнение,

совпадение при сравнении и захват событий. В регистре TIMSK находятся биты масок прерываний Таймера/счетчика 1. При внешнем тактировании Таймера/счетчика 1 внешний сигнал привязывается к частоте процессора, стробирование осуществляется нарастающим фронтом тактового сигнала процессора. Для правильной работы Таймера/счетчика 1 от внешнего тактового сигнала минимальное время между двумя переключения этого сигнала должно быть не менее одного периода тактового сигнала процессора.

Наилучшие точность и разрешение Таймер/счетчик 1 обеспечивает при наименьшем коэффициенте предварительного деления. С другой стороны, высокий коэффициент предварительного деления удобен при реализации низкоскоростных функций или синхронизации редко происходящих событий. Таймер/счетчик 1 поддерживает две функции сравнения, используя регистры сравнения OCR1A и OCR1B для хранения значений, которые сравниваются с содержимым Таймера/счетчика 1. Функции сравнения включают опции очистки счетчика по совпадению сравнения A и изменения состояния на внешних выводах при совпадениях сравнения A и B.

Таймер/счетчик 1 может быть использован в качестве 8-, 9- или 10-разрядного широтно-импульсного модулятора. В этом режиме счетчик и регистры OCR1A/OCR1B работают как сдвоенный самостоятельный ШИМ с центрированными импульсами.

Функция захвата по входу обеспечивает захват содержимого Таймера/счетчика1, запускаемый внешним событием на выводе PD4/(IC1). Условия захвата определяются регистром управления TCCR1B. Кроме того, аналоговый компаратор может быть использован для формирования сигнала захвата.

Регистр управления A Таймера/счетчика1 – TCCR1A

	7	6	5	4	3	2	1	0	
$2F ($4F)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	PWM11	PWM10	TCCR1A
Исх.код	0	0	0	0	0	0	0	0	

TCCR1A.7,6 – COM1A1, COM1A0: Функция выхода сравнения A. Биты COM1A1 и COM1A0 определяют характер сигнала на выходе после совпадения при сравнении. Выходной сигнал поступает на вывод OC1A, соответствующий бит управления направлением должен быть установлен в 1. Коды управления представлены в следующей таблице.

Выбор режима сравнения Таймера/счетчика1

COM1X1	COM1X0	Описание
0	0	Таймер/счетчик1 отключен от вывода выхода OC1x
0	1	Переключение выходной линии OC1x
1	0	Низкий уровень на выводе OC1x
1	1	Высокий уровень на выводе OC1x

x=A или B

TCCR1A.5,4 – COM1B1, COM1B0: Функция выхода сравнения B. Биты COM1B1 и COM1B0 определяют характер сигнала на выходе после совпадения при сравнении. Выходной сигнал поступает на вывод OC1B. Поскольку это альтернативная функция порта, то соответствующий бит управления направлением должен быть установлен в 1. Коды управления представлены в предыдущей таблице. В режиме ШИМ функции этих битов отличаются.

При изменении значений битов COM1X1/COM1X0 прерывания по сравнению Таймера/счетчика 1 должны быть запрещены очисткой битов регистре TIMSK. В противном случае при изменении битов может произойти прерывание.

TCCR1A.3..2 – Зарезервированные биты.

TCCR1A.1..0 – PWM11, PWM10: Биты выбора режима ШИМ

Эти биты определяют установку режима ШИМ в соответствии со следующей таблицей.

Выбор режима ШИМ Таймера/счетчика 1

PWM11	PWM10	Описание
0	0	Работа таймера/счетчика1 в режиме ШИМ запрещена
0	1	Работа таймера/счетчика1 в 8-разрядном ШИМ режиме
1	0	Работа таймера/счетчика1 в 9-разрядном ШИМ режиме
1	1	Работа таймера/счетчика1 в 10-разрядном ШИМ режиме

Регистр управления B Таймера/счетчика1 – TCCR1B

	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	**TCCR1B**
Исх.код	0	0	0	0	0	0	0	0	

TCCR1B.7 – ICNC1: Бит режима подавления шума на входе захвата. При очищенном (=0) бите ICNC1 функция подавления шума запрещена. Захват происходит по первому нарастающему/падающему фронту, поступившему на вывод PD4(IC1). При установленном бите ICNC1 выполняются четыре последовательных опроса состояния вывода PD4(IC1) и все четыре выборки должны иметь одинаковый уровень, в соответствии со значением бита ICES1. Частота опроса равна частоте XTAL.

TCCR1B.6 – ICES1: Выбор фронта срабатывания на входе захвата 1. При сброшенном бите ICES1 содержимое Таймера/счетчика 1 захватывается по спадающему фронту сигнала на выводе PD4(IC1). При установленном бите ICES1 захват происходит по нарастающему фронту.

TCCR1B.5,4 – Зарезервированные биты.

TCCR1B.3 - CTC1: Очистка Таймера/счетчика 1 после совпадения. При установленном бите CTC1 Таймер/счетчик 1 сбрасывается в состоя-

ние $0000 в течение тактового цикла, следующего за совпадением при сравнении A. Если бит CTC1 очищен, Таймер/счетчик1 продолжает отсчет и не реагирует на совпадение при сравнении. Поскольку совпадение при сравнении детектируется в течение тактового цикла процессора, следующего за совпадением, то поведение функции будет отличаться при установке коэффициента предварительного деления Таймера/счетчика 1, больше 1. При коэффициенте предварительного деления 1 и значении C в регистре сравнения таймер будет считать в соответствии с установкой CTC1:

..| C-2 | C-1 | C | 0 | 1 |..

При коэффициенте предварительного деления 8 таймер будет считать следующим образом:

..| C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2| C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1| C, 0 , 0, 0, 0, 0, 0, 0|...

В режиме ШИМ состояние бита CTC1 значения не имеет.

*TCCR1B.*2,1,0 – CS12, CS11, CS10: Биты выбора источника тактовой частоты и коэффициента деления.

Условие Stop выполняет функцию разрешения/запрещения Таймера/счетчика1. В режимах с предварительным делением делится частота тактового генератора CK. При использовании внешнего тактирования необходимо выполнить соответствующие установки в регистре управления направлением (очистка бита порта переводит вывод в режим входа).

Выбор источника тактового сигнала Таймера/счетчика 1

CS12	CS11	CS10	Источник
0	0	0	Stop - Таймер/счетчик1 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 сигнал на выводе T1, нарастающий фронт
1	1	1	Внешний сигнал на выводе T1, спадающий фронт

Таймер/счетчик 1 – TCNTH и TCNT1L

	7	6	5	4	3	2	1	0	
$2D ($4D)	MSB								TCNT1H
$CE ($4C)								LSB	TCNT1L
Исх.код	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Этот 16-разрядный регистр содержит текущее значение Таймера/счетчика 1. Для того, чтобы процессор мог читать/записывать стар-

ший и младший байты этого регистра одновременно, доступ выполняется с использованием 8-разрядного регистра временного хранения (TEMP). Этот регистр используется также при обращении в OCR1A, OCR1B и ICR1. Если основная программа и процедуры обработки прерываний используют обращение к регистрам через TEMP, прерывания должны быть запрещены на время обращений из основной программы (и процедур прерываний, если прерывания вложенные).

Запись в TCNT1. Когда процессор производит запись в старший байт (TCNT1H), записываемые данные размещаются в регистре TEMP. Затем, когда процессор производит запись в младший байт (TCNT1L), данные младшего байта объединяются с байтом данных регистра TEMP и все 16 битов одновременно переписываются в регистр TCNT1. Поэтому при 16-разрядных операциях записи обращение к старшему байту (TCNT1H) должно выполняться первым. При использовании Таймера/счетчика 1 в качестве 8-разрядного таймера достаточно производить запись только младшего байта.

Чтение TCNT1. Когда процессор читает младший байт (TCNT1L), то содержимое TCNT1L выбирается непосредственно, а содержимое старшего байта (TCNT1H) размещается в регистре TEMP. Следовательно, при 16-разрядных операциях чтения первым должно выполняться обращение к младшему байту (TCNT1L). При использовании Таймера/счетчика 1 в качестве 8-разрядного таймера достаточно производить чтение только младшего байта.

Таймер/счетчик 1 работает, как счетчик с нарастанием или реверсивный счетчик (в ШИМ режиме), имеет возможность чтения/записи. Если в Таймер/счетчик 1 занесено некоторое значение и выбран источник тактового сигнала, то Таймер/счетчик 1 начнет счет в следующем тактовом цикле после установки в нем значения.

Регистры сравнения A Таймера/счетчика 1 – OCR1AH и OCR1AL

	7	6	5	4	3	2	1	0	
$2B	MSB								OCR1AH
$2A								LSB	OCR1AL
Исх.коде	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Регистры сравнения B Таймера/счетчика 1 – OCR1BH и OCR1BL

	7	6	5	4	3	2	1	0	
$29	MSB								OCR1BH
$28								LSB	OCR1BL
Исх.код	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

16-разрядные регистры сравнения выхода обеспечивают и чтение, и запись.

Регистры сравнения выхода Таймера/счетчика 1 хранят эталон, постоянно сравниваемый с состоянием Таймера/счетчика 1. Действие, запускаемое совпадением при сравнении, определяется содержимым регистра управления Таймера/счетчика 1. Совпадение при сравнении происходит, когда Таймер/счетчик 1 досчитает до значения, хранящегося в OCR. Если в TCNT1, OCR1A или OCR1B записаны одинаковые значения, то совпадение при сравнении сформировано не будет.

Совпадение при сравнении устанавливает флаг прерывания в цикле процессора, следующем за самим совпадением.

Поскольку регистры OCR1A и OCR1B являются 16-разрядными, то для обеспечения одновременного занесения старшего и младшего байтов данных используется регистр временного хранения TEMP. Когда процессор записывает старший байт, данные временно сохраняются в регистре TEMP. Содержимое регистра TEMP переписывается в OCR1AH или OCR1BH, когда процессор записывает младший байт OCR1AL или OCR1BL. Поэтому, при 16-разрядных операциях старшие байты регистров OCR1A/B должны записываться первыми.

Регистр TEMP используется, также, при обращении к регистрам TCNT1 и ICR1. Если основная программа и процедуры обработки прерываний используют обращение к регистрам посредством TEMP, то прерывания должны быть запрещены на время обращений.

Регистр захвата Таймера/счетчика 1 – ICR1H и ICR1L

	7	6	5	4	3	2	1	0	
$27 ($37)	MSB								ICR1H
$26 ($36)								LSB	ICR1L
Исх.код	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

16-разрядный регистр захвата обеспечивает только чтение содержимого.

При обнаружении на выводе PD4(IC1) нарастающего или спадающего фронта сигнала (зависит от установки ICES1) текущее состояние Таймера/счетчика 1 пересылается в регистр ICR1. Одновременно устанавливается флаг ICF1.

Поскольку регистр захвата является 16-разрядным, то для обеспечения одновременного чтения старшего и младшего байтов регистра ICR1 используется регистр временного хранения TEMP. При чтении процессором младшего байта значение ICR1L пересылается непосредственно, а значение старшего байта ICR1H размещается в регистре TEMP. Поэтому при работе с 16-разрядным регистром операцию чтения необходимо начинать с младшего байта ICR1L.

Таймер/счетчик 1 в режиме ШИМ. При установленном режиме ШИМ Таймер/счетчик 1 и регистры сравнения (OCR1A/ OCR1B), образуют сдвоенный 8-, 9- или 10-разрядный автономный генератор ШИМ с выходами на выводах PB5(OC1A) и PB6(OC1B). Таймер/счетчик 1 работает как реверсивный счетчик, считающий от $0000 до верхней границы. Когда значение в счетном регистре совпадает с содержимым 10 младших битов регистров OCR1A или OCR1B, сигнал на выводах PB5(OC1A)/PB6(OC1B) принимает значение 0 или 1, в соответствии с установками битов COM1A1/COM1A0 или COM1B1/COM1B0 регистра управления TCCR1A (таблица).

Наибольшие значения таймера и частота ШИМ

Разрядность ШИМ	TOP значения таймера	Частота ШИМ
8-разрядов	$00FF(255)	F_{TC1} / 510
9-разрядов	$01FF(511)	F_{TC1} / 1022
10-разрядов	$03FF(1023)	F_{TC1} /2046

Выбор режима сравнения Таймера/счетчика 1 в режиме ШИМ

COM1x1	COM1x0	Выходной сигнал на OCX1
0	0	Не подключен
0	1	Не подключен
1	0	Очищается (=0) по совпадению при счете вверх. Устанавливается (=1) по совпадению при счете вниз (не инвертированный ШИМ).
1	1	Очищается (=0) по совпадению при счете вниз. Устанавливается (=1) по совпадению при счете вверх (инвертированный ШИМ).

x=A или B

Когда OCR1 содержит $0000 или значение верхней границы, вывод OC1A/OC1B принимает значения 0 или 1, соответственно установкам COM1A1/COM1A0 или COM1B1/COM1B0.

Состояние выходов в режиме ШИМ при OCR1x = $0000 или TOP

COM1X1	COM1X0	OCR1X	Состояние выводов OC1X
1	0	$0000	0
1	0	TOP	1
1	1	$0000	1
1	1	TOP	0

x=A или B

В режиме ШИМ флаг переполнения (TOV1) устанавливается по достижении значения $0000. Прерывание по переполнению таймера 1 работает так же, как в обычном режиме таймера/счетчика. Это относится и к прерыванию по сравнению таймера 1.

Сторожевой таймер (Watchdog Timer). Сторожевой таймер тактируется отдельным встроенным генератором с частотой 1 Мгц. Установкой коэффициента деления тактовой частоты можно изменять длительность интервала до сброса по сторожевому таймеру от 16 тыс. до 2048 тыс. циклов (от 16 до 2048 мс). Программно сторожевой таймер сбрасывается командой WDR (Watchdog Reset).

С момента сброса сторожевого таймера до завершения внутреннего сброса микроконтроллера длится период времени, который зависит от коэффициента деления тактовой частоты таймера. Если этот период завершился и другой сигнал сброса сторожевого таймера не поступил, микроконтроллер начинает работу с вектора сброса.

Прежде, чем разрешать сторожевой таймер, необходимо выполнить команду WDR и загрузить значение в счетчик.

Регистр управления сторожевым таймером – WDTCR

	7	6	5	4	3	2	1	0	
$21($41)	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Исх.код	0	0	0	0	0	0	0	0	

WDTCR.7..5 – Зарезервированные биты.

WDTCR.4 – WDTOE: Бит разрешения отключения сторожевого таймера. Этот бит должны быть установлены (=1) при очистке бита WDE. Иначе сторожевой таймер не будет запрещен. Если этот бит установлен, то он аппаратно очищается через четыре тактовых цикла.

WDTCR.3 – WDE: Бит разрешения сторожевого таймера. Если бит WDE установлен (=1), то сторожевой таймер разрешен. Если бит WDE очищен (=0), то функционирование сторожевого таймера запрещено. Бит WDE может быть очищен только если установлен бит WDTOE. Для запрещения работы сторожевого таймера необходимо выполнить следующую процедуру:

- Одной командой записать 1 в WDTOE и WDE. Логическая 1 должна быть записана в WDE, даже если этот бит был установлен перед началом операции запрета сторожевого таймера.
- За время последующих четырех циклов записать логический 0 в WDE. Сторожевой таймер будет запрещен.

WDTCR.2..0 – WDP2, WDP1, WDP0: Биты установки коэффициента предварительного деления сторожевого таймера. Состояние битов WDP2, WDP1 и WDP0 определяет коэффициент деления тактовой частоты сторожевого таймера.

Коэффициенты деления частоты сторожевого таймера

WD2	WDP1	WDP0	Длительность цикла сторожевого таймера	Длительность периода сброса (Vcc = 5,0В), мс
0	0	0	16 тыс. циклов	15
0	0	1	32 тыс. циклов	30
0	1	0	64 тыс. циклов	60
0	1	1	128 тыс. циклов	120
1	0	0	256 тыс. циклов	240
1	0	1	512 тыс. циклов	490
1	1	0	1024 тыс. циклов	970
1	1	1	2048 тыс. циклов	1900

4.7. Последовательные интерфейсы – SPI и UART

Микроконтроллер ATmega103 имеет два последовательных порта: SPI (Serial Peripheral Interface) и UART (Universal Asynchronous Receiver and Transmitter).

4.7.1 Последовательный периферийный интерфейс – SPI

Последовательный порт SPI обеспечивает высокоскоростной синхронный обмен данными. Основные характеристики интерфейса:

- Дуплексный 3-проводный синхронный обмен данными;
- Режимы работы: ведущий или ведомый;
- Старший или младший бит в начале посылки;
- Четыре скорости обмена данными;
- Установка флага прерывания по окончании передачи;
- Активизация из режима Idle (только в режиме ведомого).

Соединение между ведущим и ведомым устройствами с использованием SPI-интерфейса показано на рис. 4.16. Вывод SCK (PB1) является выходом тактового сигнала ведущего устройства и входом тактового сигнала ведомого. После записи ведущим устройством данных в регистр SPI начинает работать тактовый генератор интерфейса SPI и записанные данные сдвигаются через вывод MOSI (PB2) ведущего устройства на вывод MOSI ведомого. После сдвига одного байта тактовый генератор останавливается, устанавливается флаг окончания передачи SPIF. Если в регистре SPCR установлен бит разрешения прерывания SPI (SPIE), то возникнет запрос прерывания.

Активный низкий уровень сигнала на входе SS# (PB0) определяет устройство как ведомое. При высоком уровне на входе SS# вывод MOSI может быть использован в качестве входа. Режим веду-

щий/ведомый может быть установлен и программным способом посредством установки или очистки бита MSTR в регистре управления SPI.

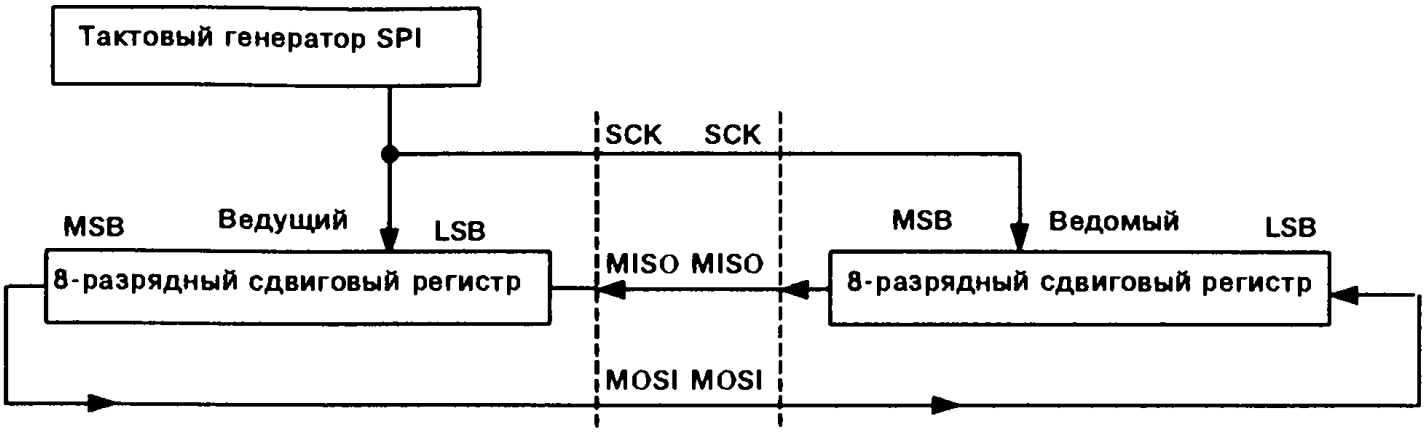


Рис. 4.16. Соединение между ведущим и ведомым устройствами с использованием интерфейса SPI

Два сдвиговых регистра ведущего и ведомого устройств (микроконтроллеров) можно рассматривать как один разнесенный 16-разрядный циклический сдвиговый регистр. При сдвиге данных из ведущего устройства в ведомое одновременно происходит сдвиг данных из ведомого устройства ведущее, т.е. в течение одного цикла сдвига происходит обмен данными между ведущим и ведомым устройствами.

В системе организована одиночная буферизация передающей стороны и двойная буферизация приемной стороны. Это означает, что передаваемые символы не могут быть записаны в регистр данных SPI прежде, чем будет полностью завершен цикл сдвига. С другой стороны, при приеме данных принимаемый символ должен быть считан из регистра данных SPI прежде, чем будет завершен прием следующего символа, в противном случае предшествовавший символ будет потерян.

Направления данных на линиях MOSI, MISO, SCK и SS# настраиваются в соответствии со следующей таблицей:

Настройка линий интерфейса SPI

Вывод	Направление ведущего	Направление ведомого
MOSI	Определяется пользователем	Вход
MISO	Вход	Определяется пользователем
SCK	Определяется пользователем	Вход
SS	Определяется пользователем	Вход

Функционирование входа SS#. Когда SPI-устройство определено как ведущее (бит MSTR регистра SPCR установлен), пользователь имеет возможность определить направление линии SS#. Если вывод SS# определен как выход, то он является выводом общего назначения и не участ-

вует в работе интерфейса SPI. Если же вывод SS# определен как вход, то для обеспечения работы ведущего устройства SPI он должен удерживаться на высоком уровне. Если в режиме ведущего вывод SS# является входом и внешней схемой на него подан низкий уровень, то интерфейс SPI воспримет его как обращение другого ведущего SPI к себе, как к ведомому. Чтобы избежать конфликтной ситуации на шине, система SPI выполняет следующие действия:

- бит MSTR в регистре SPCR очищается и устройство SPI становится ведомым. В результате выводы MOSI и SCK становятся входами;
- устанавливается флаг SPIF регистра SPSR и, если разрешено прерывание SPI, начинается выполнение процедуры обработки прерывания.

Таким образом, когда управляемое прерыванием передающее устройство SPI используется в режиме ведущего и существует вероятность подачи на вывод SS# активного сигнала низкого уровня, процедура прерывания должна проверять установку бита MSTR. Если бит MSTR был очищен выбором режима ведомого, то он должен быть установлен пользователем для переназначения устройства ведущим.

Если порт SPI является ведомым, то вывод SS# постоянно работает как вход. При подаче на вывод SS# низкого уровня устройство SPI активируется и вывод MISO становится выходом, если это определено пользователем. Все остальные выводы являются входами. Если вывод SS# удерживается на высоком уровне, то все выводы являются входами, устройство SPI пассивно и не будет получать входных данных.

Существует четыре комбинации фазы и полярности сигнала SCK относительно данных, которые задаются битами CPHA и CPOL. Форматы передачи данных SPI показаны на рис. 4.17.

Регистр управления SPI – SPCR

	7	6	5	4	3	2	1	0	
$0D ($2D)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	**SPCR**
Исх.код	0	0	0	0	0	0	0	0	

SPCR.7 – SPIE: Разрешение прерывания SPI. Установка этого бита разрешает установку бита SPIF регистра SPSR и, при разрешении всех прерываний битом I регистра SREG, обслуживание прерывания порта SPI.

SPCR.6 – SPE: Разрешение работы порта SPI. Установка этого бита разрешает подключение линий SS, MOSI, MISO и SCK к выводам PB0, PB1, PB2 и PB3.

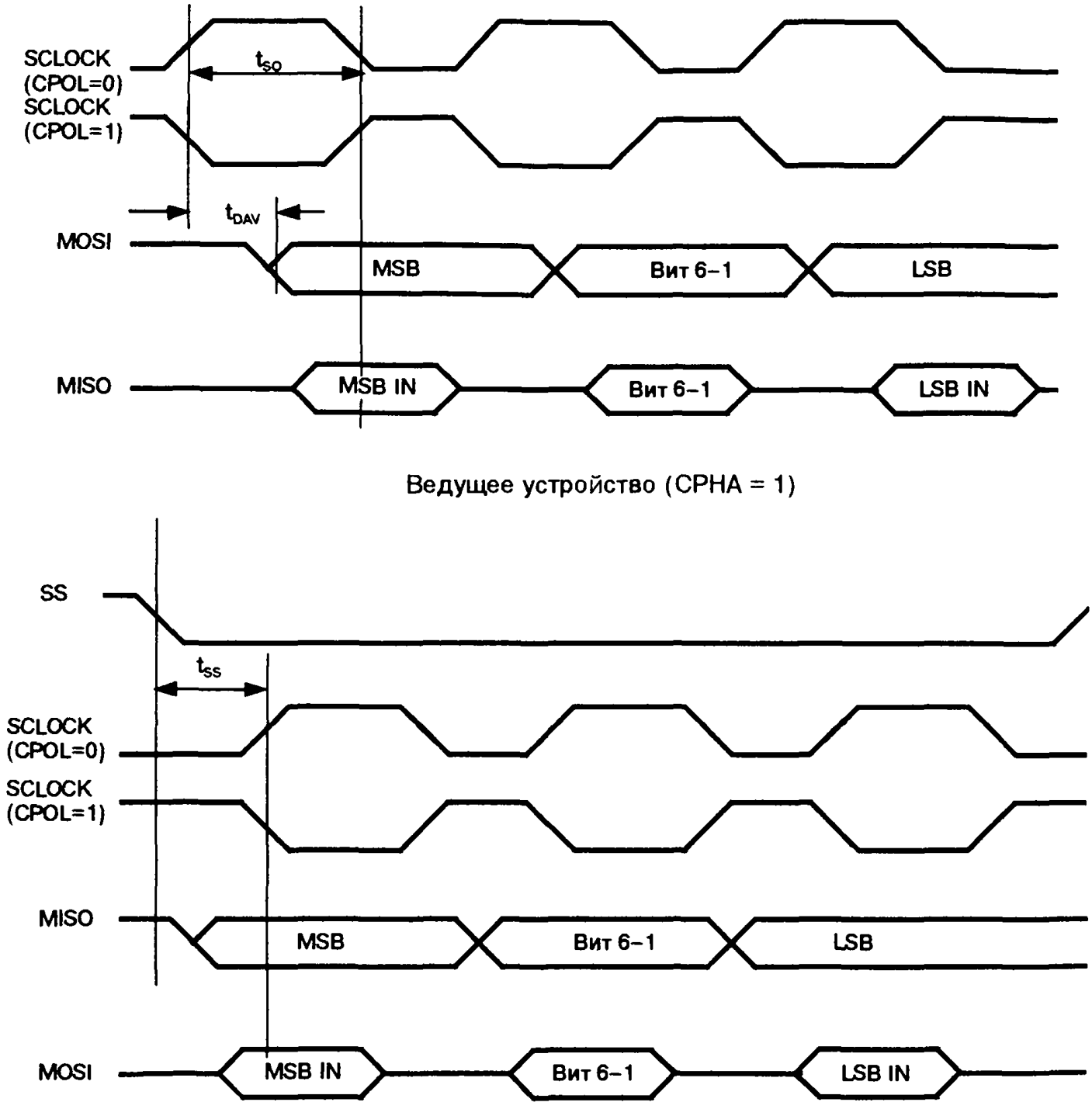


Рис. 4.17. Временные диаграммы передачи данных по SPI-интерфейсу

SPCR.5 – DORD: Порядок пересылки данных. Установка этого бита приводит к передаче посылки данных вперед младшим битом. При очищенном бите первым передается старший бит данных.

SPCR.4 – MSTR: Выбор режима ведущий/ведомый. При установленном бите MSTR порт SPI работает в режиме ведущего, а при очищенном бите – в режиме ведомого. Если SS# определен как вход и на него подан низкий уровень при установленном бите MSTR, бит MSTR будет

сброшен, а бит SPIF регистра SPSR установлен. Чтобы вновь установить режим ведущего устройства, пользователь должен установить бит MSTR.

SPCR.3 – CPOL: Полярность тактового сигнала. При установленном бите CPOL сигнал SCK в режиме Idle имеет высокий уровень, при сброшенном бите CPOL - низкий уровень.

SPCR.2 – CPHA: Фаза тактового сигнала. Действие этого бита отображено на рис.4.17.

SPCR.1,0 – SPR1, SPR0: Выбор частоты тактового сигнала. Эти два бита управляют частотой тактового сигнала устройства, работающего в режиме ведущего. В режиме ведомого значения битов влияния не оказывают. Состояния битов и устанавливаемый коэффициент деления частоты f_{CL} указаны в таблице:

SPR1	SPR0	Тактовая частота SCK
0	0	F_{CL} / 4
0	1	F_{CL} / 16
1	0	F_{CL} / 64
1	1	F_{CL} / 128

Регистр состояния SPI – SPSR

	7	6	5	4	3	2	1	0	
$0E	SPIF	WCOL	–	–	–	–	–	–	SPSR
Исх.код	0	0	0	0	0	0	0	0	

SPSR.7 – SPIF: Флаг прерывания порта SPI. После завершения обмена бит SPIF устанавливается и, если установлены биты SPIE в регистре SPCR и I в регистре SREG, прерывание порта SPI обрабатывается. Бит SPIF очищается аппаратно при переходе на вектор прерывания. Бит SPIF может быть очищен также при первом чтении регистра состояния SPSR и последующем обращении к регистру данных SPDR.

SPSR.6 – WCOL: Флаг ошибки при записи. Бит WCOL устанавливается, если в процессе передачи данных выполнялась запись в регистр данных SPDR. Бит WCOL (и бит SPIF) аппаратно очищаются при первом чтении регистра SPSR и последующем обращении к регистру данных SPDR.

*SPSR.*5..0 – Res: Зарезервированные биты.

Регистр данных SPI – SPDR

	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Исх.код	0	0	0	0	0	0	0	0	

Регистр данных SPDR представляет собой регистр с возможностью чтения/записи и предназначен для пересылки данных между регистровым файлом и сдвиговым регистром SPI. Запись в регистр SPDR инициирует передачу данных, считывание регистра приводит к чтению сдвигового регистра приемника.

4.7.2. UART - универсальный последовательный порт

Микроконтроллеры ATtmega103 оснащены универсальным дуплексным последовательным портом (UART). Его основные возможности:

- Широкий диапазон скоростей обмена данными;
- Высокая скорость передачи при низкой частоте XTAL;
- 8- и 9-разрядный форматы данных;
- Обнаружение ошибок потери данных при приеме;
- Обнаружение ошибок формата кадров;
- Обнаружение ложного стартового бита;
- Три отдельных прерывания: по завершению передачи, по пустому регистру передатчика и по завершению приема.

Передача данных. Передача данных инициируется записью передаваемых данных в регистр UDR. Из этого регистра байт данных пересылается в сдвиговый регистр передатчика в следующих случаях:

- Новый символ записан в регистр UDR после того, как был выдвинут стоповый бит предыдущего символа. Сдвиговый регистр загружается немедленно;
- Новый символ записан в регистр UDR прежде, чем был выведен стоповый бит предшествовавшего символа. Сдвиговый регистр загружается после выхода стопового бита передаваемого символа, находящегося в сдвиговом регистре.

Если 10 (11)-разрядный сдвиговый регистр передатчика пуст, данные из регистра UDR передаются в сдвиговый регистр. При этом устанавливается бит UDRE регистра USR. При установленном бите UDRE порт UART готов принять следующий символ. Запись в регистр UDR очищает бит UDRE. Когда символ передан из регистра UDR в сдвиговый регистр, к его коду вначале добавляется стартовый бит (состояние 0 – стартовый бит), а в конце – стоповый бит (состояние 1 – стоповый бит). Если в регистре управления UCR установлен бит CHR9 (т.е. выбран режим 9-разрядного слова данных), то бит TxB8 регистра UCR пересылается в бит 9 сдвигового регистра передатчика.

Сразу после пересылки данных в сдвиговый регистр импульсом синхросигнала стартовый бит выдвигается на вывод TxD. За ним следуют

биты данных, младший бит первым. После выдвижения стопового бита сдвиговый регистр загружается новыми данными, если символ был записан в регистр UDR во время передачи. В процессе загрузки бит UDRE находится в установленном состоянии. Если новые данные не будут загружены в UDR до выдачи стопового бита, флаг UDRE останется установленным. В этом случае в регистре статуса порта UART (USR) устанавливается флаг TxC завершения передачи.

Установленный бит TxEN регистра UCR разрешает передачу. При очищенном бите TxEN вывод PE1 может быть использован в качестве вывода общего назначения. При установленном бите TxEN передатчик UART подключается к выводу PE1 и использует его выхода, независимо от установки бита DDE1 в регистре DDRE.

Прием данных. Приемник проверяет состояние линии RxD с частотой в 16 раз большей, чем частота передачи. При пассивном состоянии линии перепад 1– 0 воспринимается как начало стартового бита и запускается последовательность его проверок. Вслед за переходом 1– 0 на выборках 8, 9 и 10 приемник вновь проверяет линию RxD на изменение логического состояния. Если две или более из этих трех выборок обнаружат логическую 1, то стартовый бит отвергается как шумовой всплеск и приемник начинает выявлять и анализировать следующие переходы из 1 в 0.

Если был обнаружен действительный стартовый бит, то принимаются следующие за стартовым битом информационные биты. Эти биты также тестируются на выборках 8, 9 и 10. Логическое состояние бита определяется по двум и более (из трех) одинаковым состояниям выборок. Все биты вводятся в сдвиговый регистр приемника с тем значением.

При поступлении стопового бита необходимо, чтобы не менее двух выборок из трех показали уровень 1. Если две или более выборок покажут состояние 0, то при пересылке принятого байта в регистр UDR, в регистре статуса USR устанавливается бит ошибки кадра FE. Пользователь перед чтением регистра UDR должен проверять состояние бита FE. Флаг FE очищается при чтении содержимого регистра данных UDR. Вне зависимости от того, принят правильный стоповый бит или нет, данные пересылаются в регистр UDR и устанавливается флаг RxC в регистре статуса USR. Регистр UDR фактически объединяет два физически отдельных регистра: передатчика и приемника. При чтении регистра UDR читается регистр приемника, а при записи обращение осуществляется к регистру передатчика. Если выбран режим обмена 9-разрядными словами данных (установлен бит CHR9 регистра UCR), то при пересылке данных в регистр UDR бит RxB8 регистра UCR загружается в бит 9 сдвигового регистра передачи. Если после получения символа к регистру UDR не было обращения, начиная с последнего приема, в регистре UCR устанавливается флаг OR. Это означает, что данные, принятые в сдвиговый ре-

гистр, потеряны. Бит OR доступен тогда, кода из регистра UDR прочитан байт данных. Пользователю, для обнаружения потери данных, необходимо всегда проверять флаг OR после чтения регистра UDR.

При очищенном бите RxEN регистра UCR прием запрещен и вывод PE0 может использоваться в качестве вывода общего назначения. При установленном бите RxEN приемник UART подключается к выводу PE0, который работает как вход, вне зависимости от установки бита DDE0 в регистре DDRE. При этом бит PORTE0 может использоваться для управления подтягивающим резистором вывода.

Регистр данных UART – UDR

	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR
Исх.код	0	0	0	0	0	0	0	0	

В действительности регистр UDR представляет собой сдвоенный регистр данных - регистр передатчика и регистр приемника используют один и то же адрес. При записи в регистр обращение производится к регистру передатчика, а при чтении – к регистру приемника.

Регистр статуса UART – USR

	7	6	5	4	3	2	1	0	
$0B ($2B)	RxC	TxC	UDRE	FE	OR	–	–	–	USR
Исх.код	0	0	0	0	0	0	0	0	

Регистр USR обеспечивает чтение информации о состоянии порта UART.

USR.7 – RxC: Прием завершен. Этот бит устанавливается при пересылке принятого символа из сдвигового регистра приемника в регистр UDR. Бит устанавливается вне зависимости от отсутствия или наличия ошибки приема кадра. При установленном в регистре UCR бите RxCIE и установленном бите RxC выполняется прерывание по завершению приема UART. Бит RxC очищается при чтении регистра UDR. При приеме данных по прерыванию, процедура обработки прочитать UDR для очистки бита RxC, иначе после ее завершения произойдет новое прерывание.

USR.6 – TxC: Передача завершена. Этот бит устанавливается, когда весь кадр (включая стоповый бит) выведен из сдвигового регистра передатчика, а в регистр UDR не записаны новые данные. Этот флаг используется при полудуплексном протоколе обмена, когда нужно освободить коммуникационную линию сразу после завершения передачи.

При установленном в регистре UCR бите TxCIE установка TxC приведет к выполнению прерывания по завершению передачи UART. Флаг TxC очищается аппаратно при переходе по соответствующему вектору прерывания. Очистить бит TxC можно записью в бит логической «1».

USR.5 – UDRE: Регистр данных пуст. Этот бит устанавливается, когда весь символ, записанный в UDR, пересылается в сдвиговый регистр передатчика. Установка этого бита означает, что передатчик готов к получению нового символа.

Если бит UDRIE в UCR установлен, при установленном бите UDRE существует запрос прерывания по завершению передачи UART. Бит UDRE очищается при записи в регистр UDR. При приеме данных по прерыванию, процедура обработки прерывания должна прочитать регистр UDR, чтобы очистить бит UDRE, иначе после завершения процедуры произойдет новое прерывание.

USR.4 – FE: Ошибка кадра. Этот бит устанавливается, если при приеме стопового бита обнаружено состояние 0. Бит FE очищается при приеме стопового бита с логическим уровнем 1.

USR.3 – DOR: Потеря данных. Бит DOR устанавливается при обнаружении потери данных, т.е. затирании непрочитанного символа в регистре UDR новым символом из сдвигового регистра приемника. Бит DOR очищается, когда данные приняты правильно и посланы в UDR.

USR.2..0 – Res: Зарезервированные биты.

Регистр управления UART – UCR

	7	6	5	4	3	2	1	0	
$0A ($2A)	RxCIE	TxCIE	UDRIE	FxEN	TxEN	CHR9	RxB8	TxB8	**UCR**
Исх.код	0	0	0	0	0	0	0	0	

UCR.7 – RxCIE: Разрешение прерывания по завершению приема. При установленном (=1) бите RxCIE и общем разрешении прерываний установка бита RxC в регистре USR приведет к прерыванию по завершению приема.

UCR.6 – TxCIE: Разрешение прерывания по завершению передачи. При установленном бите TxCIE и общем разрешении всех прерываний установка бита TxC в регистре USR приведет к прерыванию по завершению передачи.

UCR.5 – UDRIE: Разрешение прерывания по пустому регистру данных. При установленном бите UDRIE и общем разрешении прерываний установка бита UDRE в регистре USR приведет к прерыванию по пустому регистру данных UART.

UCR.4 – RxEN: Разрешение приемника. Установленный бит RxEN разрешает приемник UART. Если приемник запрещен, то флаги статуса TXC, DOR и FE установить невозможно. Если эти флаги установлены, то очистка бита RxEN не приведет к очистке этих флагов.

UCR.3 – TxEN: Разрешение передатчика. Установленный бит TxEN разрешает передатчик UART. При запрещении передатчика текущий сим-

вол в сдвиговом регистре и символ в регистре данных UDR будут посланы.

UCR.2 – CHR9: Режим 9-разрядных символов. При установленном бите CHR9 передаются и принимаются 9-разрядные символы плюс стартовый и стоповый биты. Девятые биты читаются и записываются с использованием битов RxB8 и TxB8 регистра UCR. Девятый бит данных может использоваться как дополнительный стоповый бит или бит контроля четности.

UCR.1 – RxB8: Прием 8-разрядных данных. При установленном бите CHR9 бит RxB8 является девятым битом данных принятого символа.

UCR.0 – TxB8: Передача 8-разрядных данных. При установленном бите CHR9 бит TxB8 является девятым битом данных передаваемого символа.

Формирователь синхросигнала последовательного обмена. Формирователь представляет собой делитель, на выходе которого образуется синхросигнал с частотой, определяемой выражением.

$$BAUD = F_{CK} / 16(UBRR+1)$$

где: BAUD – частота в бодах;

F_{CK} – частота процессора;

UBRR = содержимое регистра UBRR (0 – 255).

Регистр формирователя синхросигнала UART – UBRR

	7	6	5	4	3	2	1	0	
$09 ($29)	MSB							LSB	UBRR
Исх.код	0	0	0	0	0	0	0	0	

Регистр UBRR является 8-разрядным регистром с возможностью чтения/записи. Значение в нем определяет скорость обмена порта UART.

4.8. Аналоговый компаратор и АЦП

Микроконтроллер ATmega103 имеет встроенные аналоговый компаратор и аналого-цифровой преобразователь (АЦП).

4.8.1. Аналоговый компаратор

Аналоговый компаратор сравнивает уровни на положительном выводе PE2 (AC+) и отрицательном выводе PE3 (AC–). Если напряжение на положительном выводе AC+ больше, чем напряжение на отрицательном выводе AC–, выход аналогового компаратора ACO устанавливается в со-

стояние 1. Выход компаратора может быть использован для управления входом захвата Таймера/счетчика 1. Кроме того, компаратор может формировать свой запрос прерывания. Пользователь может задать условием формирования запроса на прерывание наличие на выходе компаратора нарастающему или спадающего фронта, а также переключения.

Регистр состояния и управления аналогового компаратора – ACSR

	7	6	5	4	3	2	1	0	
$08 ($28)	ACD	–	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Исх.код	0	0	0	0	0	0	0	0	

ACSR.7 – ACD: Запрет аналогового компаратора. При установленном бите ACD аналоговый компаратор запрещен. Отключение аналогового компаратора позволяет снизить потребление в активном и Idle режимах,установку данного бита можно производить в любое время. При изменении состояния бита ACD необходимо запрещать прерывание по аналоговому компаратору очисткой бита ACIE в регистре ACSR. В противном случае при изменении состояния бита ACD может произойти прерывание.

ACSR.6 – Res: Зарезервированный бит.

ACSR.5 – ACO: Выход аналогового компаратора. Значение бита ACO соответствует значению на выходе компаратора.

ACSR.4 – ACI: Флаг прерывания аналогового компаратора. Бит устанавливается в случае запроса компаратором прерывания, тип которого определяется битами ACIS1 и ACIS0. Подпрограмма обработки прерывания от аналогового компаратора будет выполняться при установленном бите ACIE и установленном бите I в регистре SREG. Бит ACI очищается аппаратно при переходе по соответствующему вектору прерывания. Его можно очистить также записью во флаг логической 1. При модификации других битов регистра ACSR командами SBI или CBI бит ACI будет очищен, если он был перед этими установлен.

ACSR.3 – ACIE: Разрешение прерывания аналогового компаратора. При установленном бите ACIE и установленном бите I в регистре SREG разрешается прерывание по аналоговому компаратору. При сброшенном бите ACIE прерывание запрещено.

ACSR.2 – ACIC: Разрешение захвата по сигналу аналогового компаратора. Установленный бит ACIC разрешается срабатывание функции захвата Таймера/счетчика 1 по переключению аналогового компаратора. В этом случае выход аналогового компаратора подсоединяется ко входной цепи логики захвата, что обеспечивает подавления шума и выбор типа прерывания. При очищенном бите ACIC соединения нет. Для запуска прерывания по захвату Таймера/счетчика 1 бит TICIE1 в регистре TIMSK должен быть установлен.

ACSR.1,0 – ACIS1, ACIS0: Выбор типа прерывания аналогового компаратора. Эти биты определяют тип события компаратора, при котором возникает запрос прерывания. Варианты установок показаны в следующей таблице.

ACIS1	ACIS0	События, вызывающие прерывание
0	0	Переключение выхода компаратора
0	1	Зарезервировано
1	0	Спадающий фронт на выходе компаратора
1	1	Нарастающий фронт на выходе компаратора

При изменении состояния битов ACIS1/ACIS0 прерывание аналогового компаратора должно быть запрещено. В противном случае при изменении состояния битов может произойти прерывание.

4.8.2. Аналого-цифровой преобразователь

Микроконтроллер оснащен 10-разрядным АЦП последовательного приближения. Блок АЦП включает 8-канальный аналоговый мультиплексор, позволяющий использовать любой вывод порта F в качестве входа АЦП во время преобразования. Основные характеристики встроенного блока АЦП:

- Разрешение 10 разрядов
- Точность ±1/2 LSB
- Время преобразования 70...280 мксек
- 8 мультиплексируемых входных каналов
- Режим однократного преобразования
- Прерывание по завершению ADC преобразования
- Схема подавления шумов в режиме Sleep

Для питания АЦП используются два отдельных вывода: AV_{CC} и AGND. Вывод AGND должен быть присоединен к выводу GND, а напряжение AV_{CC} не должно отличаться от напряжения V_{CC} более чем на 0,4 В. Внешнее опорное напряжение подается на вывод AREF и должно быть в диапазоне от AGND до AV_{CC}.

АЦП работает в режиме однократного преобразования, каждое преобразование должно инициировать пользователем. Работа АЦП разрешается установкой бита ADEN в регистре ADCSR. Первому преобразованию после разрешения АЦП предшествует пустое стартовое преобразование. Для пользователя это означает, что первое преобразование будет занимать на 13 циклов больше, чем обычно.

Преобразование инициируется записью логической единицы в бит ADSC. Этот бит остается установленным все время преобразования и

сбрасывается аппаратно, когда преобразование завершено. Если во время преобразования производится смена выбранного канала, АЦП завершит текущее преобразование перед переходом.

АЦП формирует 10-разрядный результат в двух регистрах – ADCH и ADCL. Для чтения правильного результата из этих регистров существует специальный механизм. При чтении результата регистр ADCL должен читаться первым. После этого доступ АЦП к регистрам данных блокируется. Это означает, что если следующее преобразование завершено между чтением ADCL и ADCH, его результат будет потерян. После чтения регистра ADCH доступ АЦП к регистрам данных восстанавливается.

АЦП имеет свои флаг и вектор прерывания. Флаг запроса ADIF устанавливается после завершения прерывания. Флаг устанавливается, даже если результат преобразования потерян из-за запаздывания с чтением регистра ADCH.

Делитель частоты для АЦП. Блок АЦП включает делитель, который формирует для него тактовый сигнал из синхросигнала процессора. АЦП работает с тактовой частотой в диапазоне от 50 до 200 кГц. При большей частоте ухудшается точность.

Биты ADPS0 – ADPS2 регистра управления ADCSR используются для формирования тактовой сигнала АЦП из сигнала XTAL с частотой выше 100 кГц. Делитель частоты работает, когда установлен бит ADEN.

При запуске АЦП установкой бита ADSC преобразование начинается по заднему фронту импульса синхросигнала АЦП. Один такт синхросигнала требуется на выборку-сохранение значения аналогового сигнала, после чего 13 тактов затрачивается на собственно преобразование и запись результата в регистры ADCL, ADCH. Далее перед новым преобразованием блоку АЦП необходим еще интервал в 2 такта, но если бит ADSC установлен, преобразование начнется немедленно.

Подавление шума АЦП. Блок АЦП включает схему подавления шума, которая разрешает преобразование в режиме Idle, что позволяет снизить шумы, создаваемые процессором.

Для реализации этой функции необходимо выполнить следующее:

1. Отключить АЦП очисткой бита ADEN.
2. Включить АЦП и одновременно запустить преобразование установкой битов ADEN и ADSC. Таким образом запускается пустое преобразование, за которым последует рабочее преобразование.
3. В течение 14 тактов АЦП-преобразования ввести Idle режим.
4. Если до завершения рабочего преобразования не произойдет другого прерывания, то прерывание АЦП активирует процессор и будет выполнена его процедура обслуживания.

Регистр выбора каналов АЦП – ADMUX

	7	6	5	4	3	2	1	0	
$07 ($27)	–	–	–	–	–	MUX2	MUX1	MUX0	ADMUX
Исх.код	0	0	0	0	0	0	0	0	

ADMUX.7..3: – Зарезервированные биты

ADMUX.2..0 – MUX2..MUX0: – Биты выбора аналогового канала. Состояние этих битов определяет, какой из восьми аналоговых каналов (0–7) будет подключен к АЦП.

Регистр управления и состояния АЦП – ADCSR

	7	6	5	4	3	2	1	0	
$06 ($26)	ADEN	ADSC	-	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSR
Исх.код	0	0	0	0	0	0	0	0	

ADCSR.7 – ADEN: ADC Enable – Разрешение ADC. Установка этого бита разрешает работу блока АЦП. Очистка бита запрещает АЦП. Запрещение АЦП в процессе преобразования прекращает преобразование.

ADCSR.6 – ADSC: Запуск АЦП преобразования. Для запуска каждого цикла преобразования необходимо устанавливать бит ADSC в состояние 1. Каждый раз после первой установки бита ADSC, выполненной после разрешения ADC или одновременно с разрешением ADC, будет выполняться пустое преобразование, предшествующее активному преобразованию. Это пустое преобразование инициализирует АЦП.

Бит ADSC остается установленным в течение всего цикла преобразования и сбрасывается после завершения преобразования. При выполнении пустого преобразования бит ADSC остается установленным до завершения активного преобразования. Запись 0 в этот бит влияния не оказывает.

ADCSR.5 – зарезервированный бит. **При записи в регистр ADCSR в этот бит должен быть записан 0.**

ADCSR.4 – ADIF: Флаг прерывания ADC. Этот бит устанавливается в состояние 1 после завершения преобразования и обновления регистров данных. Прерывание по завершению АЦП преобразования обслуживается, если в состояние 1 установлены бит ADIE и бит I регистра SREG. Бит ADIF сбрасывается аппаратно при переходе на вектор прерывания. Кроме того, бит ADIF может быть очищен записью во флаг логической 1.

ADCSR.3 – ADIE: Разрешение прерывания от блока АЦП. При установленных в состояние 1 бите ADIE и бите I регистра SREG разрешается прерывание по завершению АЦП-преобразования.

ADCSR.2..0 – ADPS2..ADPS0: Выбор коэффициента предварительного деления. Эти биты определяют коэффициент деления частоты XTAL для получения тактовой частоты АЦП.

ADPS2	ADPS1	ADPS0	Коэффициент деления
0	0	0	Без деления
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Регистры данных АЦП – ADCL и ADCH

	7	6	5	4	3	2	1	0	
$05 ($25)	—	—	—	—	—	—	—	—	ADCH
$04 ($24)	—	—	—	—	—	—	—	—	ADCL
Исх.код	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	

После завершения цикла преобразования результат преобразования размещается в регистрах ADCL и ADCH. Важно, чтобы оба регистра были прочитаны и чтобы регистр ADCL читался перед чтением регистра ADCH.

4.9. Пример реализации широтно-импульсного модулятора

Таймеры-счетчики микроконтроллеров используются для формирования временной сетки (последовательности интервалов времени), отдельных временных интервалов различной длительности, а также ШИМ-сигналов (с широтно-импульсной модуляцией). Кроме того, они являются инструментом анализа событий на основе механизмов захвата-сравнения.

Рассмотренная ниже программа демонстрирует организацию программного таймера реального времени с 24-разрядным счетным регистром, который наращивается по прерыванию таймера 0. Кроме того, программа формирует ШИМ-сигнал с использованием таймера 1 и выдает на линию порта B.6 (PWM1B) импульсы с линейно изменяемой скважностью («гармошка»). Если к этому выводу подключить интегратор, то на его выходе получится «пила» – аналоговый сигнал с линейно изменяющейся амплитудой.

```
; ********* Пример организации режима PWM  микроконтроллера ******
;                  частота резонатора 4.0 МГц
;********* Описание констант ************************
      .EQU  Rld0  = -125                    ; Константа перезагрузки  таймера T0
                                            ; для формирования временного
                                            ;интервала = 4 ms
                                            ; при N=128 (Prescaler, F=4 MHz)
```

```
;********* Описание переменных **********************
     .DEF PWL = r15                    ;переменная перезагрузки
     .DEF A = r16
     .DEF B = r17
     .DEF Flags = r23                  ;регистр флагов, бит 4 –
                                       ;флаг PwmDown
     .DEF TimeL  = r24                 ;программный таймер, цена
                                       ;деления 1 ms
     .DEF TimeM  = r25
     .DEF TimeH  = r26

     .EQU PwmDown = 4                  ;бит 4 - флаг PwmDown

;******* Основная часть программы ******************

     .ORG 0                            ; начало таблицы переходов
     rjmp  Start

     .ORG $2                           ;вектор Int0
                                       reti

       .ORG  $4                        ;вектор Int1
                                       ;(не используется)
       reti

       .ORG $6                         ;вектор Int2
                                       ;(не используется)
       reti

       .ORG $8                         ;вектор Int3
                                       ;(не используется)
       reti

       .ORG $A                         ;вектор Int4
                                       ;(не используется)
       reti

       .ORG $C                        ;вектор Int5
                                       ;(не используется)
       reti

       .ORG $E                         ;вектор Int6
                                       ;(не используется)
       reti

       .ORG $10                        ;вектор Int7
                                       ;(не используется)
       reti

       .ORG $12h                       ;вектор Timer2 Compare
                                       ;(не используется)
     reti

       .ORG $14                        ;вектор Timer2 Overflow
                                       ;(не используется)
       reti
```

```
    .ORG $16                    ;вектор Timer1 Capture
                                ;(не используется)
    reti

    .ORG $18                    ;вектор Timer1 CompareA
                                ;(не используется)
    reti

    .ORG $1A                    ;вектор Timer1 CompareB
                                ;(не используется)
    reti

    .ORG $1C                    ;вектор Timer1 Overflow
    rjmp IntT1

    .ORG $1E                    ;вектор Timer0 Compare
                                ;(не используется)
    reti

    .ORG $20                    ;вектор Timer0 Overflow
    rjmp IntT0

    .ORG $22h                   ;вектор SPI
                                ;(не используется)
    reti

    .ORG $24                    ;вектор приемника UART
                                ;(не используется)
    reti

    .ORG $26                    ;вектор «буфер передатчика
                                ;пуст» UART
                                ;(не используется)
    reti

    .ORG $28h                   ;вектор передатчика UART
                                ;(не используется)
reti

    .ORG $2A                    ;вектор AD преобразователя
                                ;(не используется)
reti

    .ORG $2C                    ;вектор EEPROM
                                ;(не используется)
    reti

    .ORG $2E                    ;вектор компаратора
                                ;(не используется)
    reti

    .ORG $30                    ;начало основной программы
    Start:
    clr  TimeL
    clr  TimeM
```

```
        clr  TimeH
        clr  Flags
        clr  PWL

        ldi  A,$0                       ;указатель стека SP= $DF
        out  SPH,A
        ldi  A,$DF
        out  SPL,A

        ldi  A,$5                       ;запускаем таймер T0, CK/128
        out  TCCR0,A
        ldi  A,$21                      ;таймер T1, PWM 8 bit,
                                        ;очищается по совпадению
                                        ;при счете вверх
        out  TCCR1A,A
        ldi  A,$1                       ;запускаем таймер T1, Ck/1
        out  TCCR1B,A
        ldi  A,$5                       ;разрешим прерывания
                                        ;от T0,T1
        out  TIMSK,A
        sbi  DDRB,6                     ; OC1B – выход

        sei                             ;разрешим все прерывания

        ;Рабочий цикл программы, ничего не делаем.
Work:
        rjmp  Work

;**** Процедура прерывания по переполнению таймера T0 ***
IntT0:
        push  A
        ldi   A,Rld0
        out   TCNT0,A                   ;перезагрузим таймер
        in    A,SREG                    ;сохраненим SREG
        adiw  TimeL,4                   ;Time:=Time+4 ms
        brcc  itN1
        inc   TimeH
    itN1:
        out  SREG,A                     ;восстановление SREG
        pop  A
        reti                            ;

;** Процедура прерывания по переполнению таймера T1 *****
IntT1:
        push  A
        push  B
        in  A,SREG                      ; сохранение  SREG
        ldi B,1                         ; один дискрет
        sbrc  Flags,PwmDown
        rjmp  itDec                     ;если  PwmDown=1,
                                        ;будем уменьшать
        add     PWL,B                   ;если PwmDown=0, то
                                        ;наращиваем PWL на дискрет
        mov   B,PWL
        cpi B, PWmax                    ; достигли верхней границы?
        brcs  itWr                      ;при PWL > PWmax
```

```
        sbr  Flags,(1<<PwmDown)     ;          ; то PwmDown:=1
        ldi  B,Pwmax                           ; PWL := PWmax

        mov  PWL,B
        rjmp itWr
itDec:
        sub  PWL,B
        brcc itWr                              ; если PWL<0
        cbr  Flags,(1<<PwmDown)                ; то PwmDown:=0
        clr  PWL
itWr:
        out  OCR1BL,PWL
        out  SREG,A                            ; восстановление SREG
        pop  B
        pop  A
        reti                                   ;
```

В начале программы описаны константа и переменные, т.е. значению константы и адресам регистров присвоены символические имена. В программе обращение к этой константе и переменным производится по присвоенным именам.

Основная часть программы начинается с таблицы переходов системы сброса и прерываний. Собственно программа начинается с адреса $30 обнулением используемых регистров и загрузкой в указатель стека исходного адреса $DF. Далее загрузкой регистров управления для таймера 0 указывается коэффициент деления тактовой частоты, а для таймера 1 указывается режим 8-разрядного ШИМ с низким уровнем выходного сигнала при совпадении (счетчика и уставки) и счете вверх. Тактовая частота для таймера 1 определена равной CK (системная частота). Загрузка регистра TIMSK кодом $05 разрешает прерывания по переполнению таймеров 0 и 1, а установка бита 6 регистра направления DDRB определяет вывод PORTB.6 как выход. В конце основной программы выводы порта B определяются как выходы и устанавливается общий флаг разрешения прерываний. Далее программа зацикливается и все действия выполняются по запросам прерываний от таймеров 0 и 1.

Процедура обработки прерывания по переполнению Таймера 0 начинается с сохранения в стеке регистра A (R16), затем через регистр A перезагружается счетный регистр таймера 0 (TCNT0) и далее в регистре A сохраняется содержимое SREG. После этого программный счетчик, включающий регистры TimeL, TimeM и TimeH, увеличивается на одно значение (4 мсек). Командой ADIW к 16-разрядному слову в младших двух регистрах прибавляется значение дискрета, при возникновении переноса инкрементируется старший байт в регистре TimeH. При таком подходе программный счетчик, работающий очень долго, является действительно счетчиком реального времени, т.е. в нем всегда находится текущее значение времени в миллисекундах от момента его запуска. Реальное время в миллисекундах обеспечивается соотношением тактовой частоты,

коэффициента ее деления для таймера 0 и константы перезагрузки таймера 0. В завершение процедуры восстанавливаются значения SREG и регистра A.

Процедура обработки прерывания по переполнению Таймера 1 начинается с сохранения в стеке регистров A и B, далее в регистре A сохраняется содержимое SREG, а в регистр B помещается значение дискрета изменения порога (1). Управление ШИМ начинается с анализа флага PwmDown (R23.4). Если флаг установлен («гармошка» сжимается), то осуществляется переход на метку itDec, где из значения PWL вычитается дискрет и при отсутствии заема новое значение заносится в регистр сравнения OCR1B, изменяя порог и ширину импульса. При возникшем заеме флаг PwmDown сбрасывается, PWL обнуляется и заносится в регистр сравнения OCR1B. Если флаг PwmDown сброшен («гармошка» растягивается), то к значению PWL прибавляется дискрет, новое значение сравнивается в верхней границей (если меньше границы, устанавливается флаг C) и если значение меньше границы, новое значение заносится в регистр сравнения OCR1B, изменяя порог и ширину импульса. Если значение вышло за верхнюю границу, флаг PwmDown устанавливается (переход на уменьшение), значение PWL приравнивается к верхней границе (PWL = PWmax) и новое значение заносится в регистр сравнения OCR1B. В конце процедуры восстанавливаются значения SREG и регистров B и A.