

Ассемблер контроллеров семейства ATtiny, ATmega

Инструкция	Описание	Функция	Флаги	#	Ограничение
АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ИНСТРУКЦИИ					
ADD	Rd, Rr	Add	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl, K	Add K to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2 R24, R26, R28, R30 K = 0...63
SUB	Rd, Rr	Subtract	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1 R16 - R31
SBC	Rd, Rr	Subtract w Carry	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract w Carry	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1 R16 - R31
SBIW	Rdl, K	Subtract K from W	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2 R24, R26, R28, R30 K = 0...63
AND	Rd, Rr	Logical AND	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND	$Rd \leftarrow Rd \cdot K$	Z,N,V	1 R16 - R31
OR	Rd, Rr	Logical OR	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR	$Rd \leftarrow Rd \vee K$	Z,N,V	1 R16 - R31
EOR	Rd, Rr	Exclusive OR	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Compl.	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1 инверсия
NEG	Rd	Two's Compl.	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd, K	Set Bit(s)	$Rd \leftarrow Rd \vee K$	Z,N,V	1 R16 - R31
CBR	Rd, K	Clear Bit(s)	$Rd \leftarrow Rd \cdot (0xFF - K)$	Z,N,V	1 R16 - R31
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus		Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow 0x00$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	none	1 R16 - R31
MUL	Rd, Rr	Mul Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2 Atmega
MULS	Rd, Rr	Mul Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2 Atmega
MULSU	Rd, Rr	Mul Sign w Unsign	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2 Atmega
FMUL	Rd, Rr	Fractional Mul	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2 Atmega
FMULS	Rd, Rr	Fractional Mul	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2 Atmega
FMULSU	Rd, Rr	Fractional Mul	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2 Atmega

SREG: Status Register

C: Carry Flag
Z: Zero Flag
N: Negative Flag
V: Two's complement overflow indicator
S: $N \oplus V$, For signed tests
H: Half Carry Flag
T: Transfer bit used by BLD and BST instructions
I: Global Interrupt Enable

Rd: Destination (and source) register in the Register File
Rr: Source register in the Register File
R: Result after instruction is executed
K: Constant data
k: Constant address
b: Bit in the Register File or I/O Register (3-bit)
X, Y, Z: Indirect Address Register
P: I/O location address
q: Displacement for direct addressing (6-bit)

РАБОТА С БИТАМИ					
SBI	P, b	Set Bit in I/O	$I/O(P,b) \leftarrow 1$	none	2 Address range 0x00 - 0x1F
CBI	P, b	Clear Bit in I/O	$I/O(P,b) \leftarrow 0$	none	2 Address range 0x00 - 0x1F
LSL	Rd	Logical Shift Left	$7 \leftarrow 6 \leftarrow 5 \leftarrow 4 \leftarrow 3 \leftarrow 2 \leftarrow 1 \leftarrow 0 \leftarrow (0)$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	$(0) \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left w Carry	$C \leftarrow 7 \leftarrow 6 \leftarrow 5 \leftarrow 4 \leftarrow 3 \leftarrow 2 \leftarrow 1 \leftarrow 0 \leftarrow (C)$	Z,C,N,V	1
ROR	Rd	Rotate Right w Carry	$(C) \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow C$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$(7) \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(7..4) \leftrightarrow Rd(3..0)$	none	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load	$Rd(b) \leftarrow T$	none	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative	$N \leftarrow 1$	N	1
CLN		Clear Negative	$N \leftarrow 0$	N	1
SEZ		Set Zero	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1 Interrupt
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1 Interrupt
SES		Set Signed	$S \leftarrow 1$	S	1
CLS		Clear Signed	$S \leftarrow 0$	S	1
SEV		Set Twos Compl Ovf	$V \leftarrow 1$	V	1
CLV		Clear Twos Comp Ovf	$V \leftarrow 0$	V	1
SET		Set T	$T \leftarrow 1$	T	1
CLT		Clear T	$T \leftarrow 0$	T	1
SEH		Set Half Carry	$H \leftarrow 1$	H	1
CLH		Clear Half Carry	$H \leftarrow 0$	H	1

Операторы в выражениях и директивах

! Логическое отрицание
~ Побитное отрицание
- Минус
***** Умножение
/ Деление
% Остаток от деления
+ Суммирование
- Вычитание
<< Сдвиг влево
>> Сдвиг вправо
< Меньше чем
<= Меньше или равно
> Больше чем
>= Больше или равно
== Равно
!= Не равно
& Побитное OR
^ Побитное XOR
| Побитное OR
&& Логическое AND
|| Логическое OR

1 << n установить n-й бит

СПЕЦИАЛЬНЫЕ ИНСТРУКЦИИ					
NOP		No Operation		none	1
SLEEP		Enter sleep mode		none	1
WDR		Watchdog Reset		none	1
BREAK		Break		none	- On-chip Debug

Ассемблер контроллеров семейства ATtiny, ATmega

Инструкция	Описание	Функция	Флаги	#	Ограничение
ИНСТРУКЦИИ ПЕРЕДАЧИ ДАННЫХ					
MOV	Rd, Rr	Move	$Rd \leftarrow Rr$	none	1
MOVW	Rd, Rr	Move WORD	$Rd+1:Rd \leftarrow Rr+1:Rr$	none	1
LDI	Rd, K	Load immediate	$Rd \leftarrow K$	none	1 R16-R31
LD	Rd, X Y Z	Load Indirect	$Rd \leftarrow (X Y Z)$	none	2 X, Y, Z
LD	Rd, X Y Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (X Y Z), X Y Z \leftarrow X Y Z+1$	none	2 X, Y, Z
LD	Rd, -X Y Z	Load Indirect and Pre-Dec.	$X Y Z \leftarrow X Y Z-1, Rd \leftarrow (X Y Z)$	none	2 X, Y, Z
LDD	Rd, Y Z+q	Load Indirect with offset	$Rd \leftarrow (Y Z+q)$	none	2 Y, Z
LDS	Rd, k	Load Direct SRAM	$Rd \leftarrow (k)$	none	2
ST	X Y Z, Rr	Store Indirect	$(X Y Z) \leftarrow Rr$	none	2 X, Y, Z
ST	X Y Z+, Rr	Store Indirect and Post-Inc.	$(X Y Z) \leftarrow Rr, X Y Z \leftarrow X Y Z+1$	none	2 X, Y, Z
ST	-X Y Z, Rr	Store Indirect and Pre-Dec.	$X Y Z \leftarrow X Y Z-1, Rd \leftarrow (X Y Z)$	none	2 X, Y, Z
STD	Y Z+q, Rr	Store Indirect with offset	$Rd \leftarrow (Y Z+q)$	none	2 Y, Z
STS	K, Rr	Store Direct SRAM	$Rd \leftarrow (k)$	none	2
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	none	3 R0 если не указан
LPM	Rd, Z+	Load Program Memory	$Rd \leftarrow (Z), Z \leftarrow Z+1$	none	3
SPM		Store Program Memory	$(Z) \leftarrow R1:R0$	none	- Отсутствует в TINY
IN	Rd, P	In Port	$Rd \leftarrow P$	none	1 Addr range 0x00-0x3F
OUT	P, Rr	Out Port	$P \leftarrow Rr$	none	1 Addr range 0x00-0x3F
PUSH	Rr	Push to Stack	$STACK \leftarrow Rr$	none	2
POP	Rd	Pop from Stack	$Rd \leftarrow STACK$	none	2

ИНСТРУКЦИИ ВЕТВЛЕНИЯ					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	none	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	none	2
JMP	k	Direct Jump	$PC \leftarrow k$	none	3 ATmega168
RCALL	k	Relative Call	$PC \leftarrow PC + k + 1$	none	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	none	3
CALL	k	Direct Call	$PC \leftarrow k$	none	4 ATmega168
RET		Subroutine Return	$PC \leftarrow STACK$	none	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4 Interrupt
CPSE	Rd, Rr	Compare, Skip if Equal		none	1/2/3
CP	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd, k	Compare	$Rd - K$	Z, N, V, C, H	1 R16-R31
SBRC	Rr, b	Skip if Bit = 0		none	1/2/3
SBRS	Rr, b	Skip if Bit = 1		none	1/2/3
SBIC	P, b	Skip if Bit I/O = 0		none	1/2/3 Addr range 0x00-0x1F
SBIS	P, b	Skip if Bit I/O = 1		none	1/2/3 Addr range 0x00-0x1F

Условные переходы					
BREQ		Z = 1	=	BRHS	H=1 --
BRNE		Z = 0	<>	BRHC	H=0 --
BRCS	BRLO	C = 1	<	BRTS	T=1 --
BRCC	BRSH	C = 0	>=	BRTC	T=0 --
BRMI		N = 1	<0	BRVS	V=1 ovr
BRPL		N = 0	>=0	BRVC	V=0 ovr
BRGE		$N \oplus V = 0$	>=	BRIE	Int ENA
BRLT		$N \oplus V = 1$	<0	BRID	Int DIS

C: Carry Flag
Z: Zero Flag
N: Negative Flag
V: Two's complement overflow indicator
S: $N \oplus V$, For signed tests
H: Half Carry Flag
T: Transfer bit
I: Global Interrupt Enable

CP Rd, Rr
BRxx <метка>

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic
$Rd > Rr$	$Z \cdot (N \oplus V) = 0$	BRLT ⁽¹⁾	$Rd \leq Rr$	$Z + (N \oplus V) = 1$	BRGE *
$Rd \geq Rr$	$(N \oplus V) = 0$	BRGE	$Rd < Rr$	$(N \oplus V) = 1$	BRLT *
$Rd \leq Rr$	$Z + (N \oplus V) = 1$	BRGE ⁽¹⁾	$Rd > Rr$	$Z \cdot (N \oplus V) = 0$	BRLT *
$Rd < Rr$	$(N \oplus V) = 1$	BRLT	$Rd \geq Rr$	$(N \oplus V) = 0$	BRGE
$Rd > Rr$	$C + Z = 0$	BRLO ⁽¹⁾	$Rd \leq Rr$	$C + Z = 1$	BRSH *
$Rd \geq Rr$	$C = 0$	BRSH / BRCC	$Rd < Rr$	$C = 1$	BRLO / BRCS
$Rd \leq Rr$	$C + Z = 1$	BRSH ⁽¹⁾	$Rd > Rr$	$C + Z = 0$	BRLO *
$Rd < Rr$	$C = 1$	BRLO / BRCS	$Rd \geq Rr$	$C = 0$	BRSH / BRCC

1. Поменять местами Rd и Rr в операции перед проверкой, например: CP Rd,Rr → CP Rr,Rd

Директивы

- .INCLUDE** Вложить другой файл
- .EXIT** Выйти из файла
- .ORG** Установить положение в сегменте
- .CSEG** Программный сегмент
- .DSEG** Сегмент данных
- .ESEG** Сегмент EEPROM
- .DB <0xFF>** - байты в FLASH/EEPROM
- .DW <0xFFFF>** - слова
- .BYTE <N>** Резервировать N байт в RAM/FLASH/EEPROM (Директива может предвшаться меткой)
- .DEF** Назначить регистру символическое имя
- .SET** Установить переменный символический эквивалент выражения
- .EQU** Установить постоянное выражение
- .DEVICE** Определить устройство для которого компилируется программа
- .MACRO** Начало макроса
- .ENDMACRO** Конец макроса
- @0...@9** - параметры макроса
- ##** - конкатенация строк в макросах
- .LIST** Включить генерацию листинга
- .NOLIST** Выключить генерацию листинга
- .LISTMAC** Включить разворачивание макросов в листинге
- Условная компиляция**
- .IFDEF <ид>** если <ид> объявлен
- .IFNDEF <ид>** не объявлен
- .IF <лог.выраж>** условная компиляция
- .ELSE**
- .ELSEIF <лог.выраж>**
- .ENDIF** конец блока
- .ERROR** "ошибка компиляции"
- .WARNING** "предупреждение"
- .MESSAGE** "простое сообщение"
- Препроцессор**
- #define <ид> <значение>** подстановка
- #undef <ид>** отмена подстановки
- ** - перенос на другую строку длинного выражения.

Индексные регистры

high : low
X = R27:R26
Y = R29:R28
Z = R31:R30