

Команда ADC - Сложить с переносом

Описание:

Сложение двух регистров и содержимого флага переноса (C), размещение результата в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd + Rr + C$

Синтаксис            Операнды:            Счетчик программ:  
(i)        ADC Rd,Rr         $0 < d < 31, 0 < r < 31$          $PC < PC + 1$

16-разрядный код операции:  
0001 11rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H:  $Rd3 * Rr3 + Rr3 + R3 + R3 * Rd3$

Устанавливается если есть перенос из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * Rr7 * R7 + Rd7 * Rr7 * R7$

Устанавливается если в результате операции образуется переполнение до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $Rd7 * Rr7 * R7 * R7 * R7 * Rd7$

Устанавливается если результат \$00, в ином случае очищается

C:  $Rd7 * Rr7 + Rr7 + R7 + R7 * Rd7$

Устанавливается если есть перенос из MSB результата, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

                  ; Сложить R1 : R0 с R3 : R2  
add r2, r0 ; Сложить младший байт  
adc r3, r1 ; Сложить старший байт с переносом

Слов: 1 (2 байта)

Циклов: 1

.....

Команда ADD - Add without Carry - Сложить без переноса

Описание:

Сложение двух регистров без добавления содержимого флага переноса (C), размещение результата в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd + Rr$

Синтаксис            Операнды:            Счетчик программ:  
(i)        ADD Rd,Rr         $0 < d < 31, 0 < r < 31$          $PC < PC + 1$

16-разрядный код операции:  
0000 11rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H:  $Rd3 * Rr3 + Rr3 + R3 + R3 * Rd3$

Устанавливается если есть перенос из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * Rr7 * R7 + Rd7 * Rr7 * R7$

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в ином случае очищается

C:  $Rd7 * Rr7 + Rr7 * R7 + R7 * Rd7$

Устанавливается если есть перенос из MSB результата, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

add r1,r2 ; Сложить r2 с r1 ( $r1=r1+r2$ )

adc r28,r28 ; Сложить r28 с самим собой  
( $r28=r28+r28$ )

Слов: 1 (2 байта)

Циклов: 1

Команда ADIW - Add Immediate to Word- Сложить непосредственное значение со словом

Описание:

Сложение непосредственного значения (0-63) с парой регистров и размещение результата в паре регистров. Команда работает с четырьмя верхними парами регистров, удобна для работы с регистрами указателями.

Операция:(i) Rdh:Rdl <- Rdh:Rdl + K

Синтаксис Операнды: Счетчик программ:

(i) ADIW Rdl,K dl O {24,26,28,30}, 0 < K < 63 PC < PC + 1

16-разрядный код операции:

1001 0110 KKdd KKKK

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Ы Ы Ы Ы Ы

S: NEV, Для проверок со знаком

V: Rdh7 R15

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R15

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R15 * R14 * R13 * R12 * R11 * R10 * R9 * R8 * R7 * R6 * R5 * R4 * R3 * R2$

Устанавливается если результат \$0000, в ином случае очищается

C:  $R15 * Rdh7$

Устанавливается если есть перенос из MSB результата, в ином случае очищается

R: (Результат) соответствует Rdh:Rdl после выполнения команды ( $Rdh7-Rdh0 = R15-R8, Rdl7-Rdl0 = R7-R0$ )

Пример:

adiw r24, 1 ; Сложить 1 с r25:r24

adiw r30, 63 ; Сложить 63 с Z указателем (r31 : r30)

Слов: 1 (2 байта)

Циклов: 2

Команда AND - Выполнить логическое AND

Описание:

Выполнение логического AND между содержимым регистров Rd и Rr и помещение результата в регистр назначения Rd.

Операция: (i)  $Rd \leftarrow Rd * Rr$

	Синтаксис	Операнды:	Счетчик программ:
(i)	AND Rd,Rr	$0 < d < 31, 0 < r < 31$	PC $\leftarrow$ PC + 1

16-разрядный код операции  
:0010 00rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	0	Ы	Ы	-

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

and r2, r3 ; Поразрядное and r2 и r3, результат поместить в r2

ldi r16, 1 ; Установить маску 0000 0001 в r16

and r2, r16 ; Выделить бит 0 в r2

Слов: 1 (2 байта)

Циклов: 1

.....

Команда ANDI - Выполнить логическое AND с непосредственным значением

Описание:

Выполнение логического AND между содержимым регистра Rd и константой и помещение результата в регистр назначения Rd.

Операция: (i)  $Rd \leftarrow Rd * K$

	Синтаксис	Операнды:	Счетчик программ:
(i)	ANDI Rd,K	$16 < d < 31, 0 < K < 255$	PC $\leftarrow$ PC + 1

16-разрядный код операции:  
0111 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	0	Ы	Ы	-

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

andi r17, \$0F ; Очистить старший ниббл r17

andi r18, \$10 ; Выделить бит 4 в r18

andi r19, \$AA ; Очистить нечетные биты r19

Слов: 1 (2 байта)

Циклов: 1

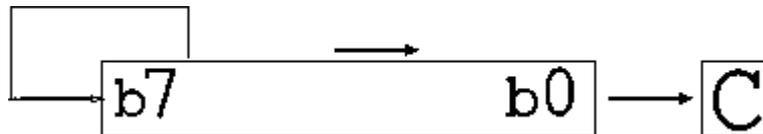
Команда ASR - Арифметически сдвинуть вправо

Описание:

Выполнение сдвига всех битов Rd на одно место вправо. Состояние бита 7 не изменяется. Бит 0 загружается во флаг переноса (C) регистра состояния (SREG). Эта команда эффективно делит значение дополнения до двух на два, без изменения знака. Флаг переноса может быть использован для округления результата.

Операция:

:



Синтаксис      Операнды:      Счетчик программ  
(i)      ASR Rd 0 < d < 31      PC <- PC + 1

16-разрядный код операции:

1001   010d   dddd   0101

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      -      Ы      Ы      Ы      Ы      Ы

S: NEV, Для проверок со знаком

V: NEC (Для N и C после сдвига)

Устанавливается если (N устанавливается и C очищается) или (N очищается а C устанавливается). В ином случае очищается (при наличии значений N и C после сдвига)

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

C: Rd0

Устанавливается если перед сдвигом были установлены LSB или Rd

R: (Результат) соответствует Rd после выполнения команды

Пример:

ldi r16, \$10 ; Загрузить десятичное значение 16 в r16

asr r16 ; r16=r16 / 2

ldi r17, \$FC ; Загрузить -4 в r17

asr r17 ; r17=r17 / 2

Слов: 1 (2 байта)

Циклов: 1

.....  
Команда BCLR - Очистить бит в регистре статуса (SREG)

Описание:

Очистка одного флага в регистре статуса

Операция:(i) SREG(s) <- 0

Синтаксис      Операнды:      Счетчик программ:  
(i)      BCLR s 0 < S < 7      PC <- PC + 1

16-разрядный код операции:  
1001   0100   1sss   1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
Ы	Ы	Ы	Ы	Ы	Ы	Ы	Ы

I:      0 если s = 7: в ином случае не изменяется  
T:      0 если s = 6: в ином случае не изменяется  
H:      0 если s = 5: в ином случае не изменяется  
S:      0 если s = 4: в ином случае не изменяется  
V:      0 если s = 3: в ином случае не изменяется  
N:      0 если s = 2: в ином случае не изменяется  
Z:      0 если s = 1: в ином случае не изменяется  
C:      0 если s = 0: в ином случае не изменяется

Пример:

bclr 0 ; Очистить флаг переноса  
bclr 7 ; Запретить прерывания

Слов: 1 (2 байта)

Циклов: 1

.....  
Команда BLD - Загрузить содержимое T флага регистра статуса (SREG) в бит регистра

Описание:

Копирование содержимого T флага регистра статуса в бит b регистра Rd

Операция:(i) Rd(b) <- T

Синтаксис      Операнды:      Счетчик программ:  
(i)      BLD Rd,b      0 < d < 31, 0 < b < 7      PC <- PC + 1

16-разрядный код операции:  
1111   100d   dddd   0bbb

Булевы выражения регистра статуса (SREG)

I	T	N	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

      ; Скопировать бит  
 bst r1, 0 ; Сохранить бит 2 регистра r1 во флаге T  
 bld r0, 4 ; Загрузить T в бит 4 регистра r0

Слов: 1 (2 байта)

Циклов: 1



Команда BRBC - Перейти если бит в регистре статуса очищен

Описание:

Условный относительный переход. Тестируется один из битов регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух.

Операция:(i) If SREG(s) = 0 then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис	Операнды:	Счетчик программ:
(i) BRBC s, k	0 < s < 7, -64 < k < +63	PC <- PC + k + 1

PC <- PC + 1,  
 если условия  
 не соблюдены

16-разрядный код операции:

1111 01kk kkkk ksss

Булевы выражения регистра статуса (SREG)

I	T	N	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

      ; Сравнить r20 со значением 5  
 brbc 1,noteq ;Перейти если флаг нуля очищен  
 .....  
 noteq: пор ..... ;Перейти по назначению(пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий



Команда BRBS - Перейти если бит в регистре статуса установлен

Описание:

Условный относительный переход. Тестируется один из битов регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно счетчика программ и представлен в форме дополнения до двух.

Операция:(i) If SREG(s) = 1 then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис	Операнды:	Счетчик программ:
(i) BRBS s, k	0 < s < 7, -64 < k < +63	PC <- PC + k + 1

PC <- PC + 1,

если условия  
не соблюдены

16-разрядный код операции:  
1111 kkkk ksss  
1111 00kk kkkk ksss

Булевы выражения регистра статуса (SREG)  
I T H S V N Z C  
- - - - - - - -

Пример:  
bst r0, 3 ;Загрузить Т битом 3 регистра r0  
brbs 6,bitset ;Перейти если бит Т установлен  
bitset: nop ;Перейти по назначению(пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий



Команда BRCC - Перейти если флаг переноса очищен

Описание:

Условный относительный переход. Тестируется бит флага переноса (C) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 0,k).

Операция:(i) If C= 0 then PC <- PC + k +1, else PC <- PC +1  
Синтаксис Операнды: Счетчик программ:  
(i) BRCC k -64 < k <+63 PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции:  
1111 01kk kkkk k000

Булевы выражения регистра статуса (SREG)  
I T H S V N Z C  
- - - - - - - -

Пример:  
add r22, r23 ; Сложить r23 с r22  
brcc posarrg ; Перейти если перенос очищен  
posarrg: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий



Команда BRCS - Перейти если флаг переноса установлен

Описание:

Условный относительный переход. Тестируется бит флага переноса (C) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 <

назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 0,k).

Операция:(i) If C= 1 then PC <- PC + k +1, else PC <- PC +1  
Синтаксис Операнды: Счетчик программ:  
(i) BRCS k -64 < k <+63 PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции:  
1111 00kk kkkk k000

Булевы выражения регистра статуса (SREG)  
I T H S V N Z C  
- - - - - - - -

Пример:  
срi r26, \$56 ; Сравнить r26 с \$56  
brcs carry ; Перейти если перенос установлен  
.....  
carry: пор ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BREQ - Перейти если равно

Описание:

Условный относительный переход. Тестируется бит флага нулевого значения (Z) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число, со знаком или без знака, представленное в Rd, эквивалентно двоичному числу, со знаком или без знака, представленному в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 1,k).

Операция:(i) If Rd = Rr (Z = 1) then PC <- PC + k +1, else PC <- PC +1  
Синтаксис Операнды: Счетчик программ:  
(i) BREQ k -64 < k <+63 PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции:  
1111 00kk kkkk k001

Булевы выражения регистра статуса (SREG)  
I T H S V N Z C  
- - - - - - - -

Пример:  
ср r1, r0 ; Сравнить регистры r1 и r0  
breq equal ; Перейти если содержимое регистров совпадает  
.....  
equal: пор ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRGE - Перейти если больше или равно (с учетом знака)

Описание:

Условный относительный переход. Тестируется бит флага знака (S) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число, со знаком представленное в Rd, больше или эквивалентно двоичному числу со знаком, представленному в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC-64 < \text{назначение} < PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 4,k).

Операция:(i) If Rd > Rr (NEV = 0) then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис      Операнды:      Счетчик программ:

(i)      BRGE k -64 < k < +63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk      kkkk      k100

Булевы выражения регистра статуса (SREG)

I	T	N	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

sr r11, r12 ; Сравнить регистры r11 и r12  
brge greateq ; Перейти если r11 >= r12 (со знаком)

greateq: nop      ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRHC - Перейти если флаг полупереноса очищен

Описание:

Условный относительный переход. Тестируется бит флага полупереноса (H) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC-64 < \text{назначение} < PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 5,k).

Операция:(i) If H = 0 then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис      Операнды:      Счетчик программ:

(i)      BRHC k -64 < k < +63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk      kkkk      k101

Булевы выражения регистра статуса (SREG)

I	T	N	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

brhc hclear ; Перейти если флаг полупереноса очищен

hclear: nop      ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRHS - Перейти если флаг полупереноса установлен

Описание:

Условный относительный переход. Тестируется бит флага полупереноса (H) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 5,k).

Операция:(i) If H = 1 then PC <- PC + k +1, else PC <- PC +1

Синтаксис            Операнды:            Счетчик программ:  
(i)      BRHS k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk    kkkk    k101

Булевы выражения регистра статуса (SREG)

I            T            H            S            V            N            Z            C

-            -            -            -            -            -            -            -  
Пример:  
brhs hset ; Перейти если флаг полупереноса установлен

hset:    nop            ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRID - Перейти если глобальное прерывание запрещено

Описание:

Условный относительный переход. Тестируется бит флага глобального прерывания (I) регистра статуса и, если бит сброшен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 << назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 7,k).

Операция:(i) If I = 0 then PC <- PC + k +1, else PC <- PC +1

Синтаксис            Операнды:            Счетчик программ:  
(i)      BRID k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk    kkkk    k111

Булевы выражения регистра статуса (SREG)

I            T            H            S            V            N            Z            C

-            -            -            -            -            -            -            -

Пример:

brid intdis ; Перейти если глобальное прерывание запрещено

intdis: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRIE - Перейти если глобальное прерывание разрешено

Описание:

Условный относительный переход. Тестируется бит флага глобального прерывания (I) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC+64 <$  назначение  $< PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 7,k).

Операция: (i) If I = 1 then PC  $\leftarrow$  PC + k + 1, else PC  $\leftarrow$  PC + 1

Синтаксис      Операнды:      Счетчик программ:  
(i) BRIE k -64 < k < +63      PC  $\leftarrow$  PC + k + 1  
PC  $\leftarrow$  PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk      kkkk      k111

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

Пример:

brie inten ; Перейти если глобальное прерывание разрешено

inten: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRLO - Перейти если меньше (без знака)

Описание:

Условный относительный переход. Тестируется бит флага переноса (C) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число без знака, представленное в Rd, меньше двоичного числа без знака, представленного в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC-64 <$  назначение  $< PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 0,k).

Операция: (i) If Rd < Rr (C = 1) then PC  $\leftarrow$  PC + k + 1, else PC  $\leftarrow$  PC + 1

Синтаксис      Операнды:      Счетчик программ:  
(i) BRLO k -64 < k < +63      PC  $\leftarrow$  PC + k + 1  
PC  $\leftarrow$  PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk      kkkk      k000

Булевы выражения регистра статуса (SREG)								
I	T	H	S	V	N	Z	C	
-	-	-	-	-	-	-	-	-

Пример:

```

eog r19, r19 ; Очистить r19
loop: inc r19 ; Увеличить на 1 r19
      .....
      cpi r19, $10 ; Сравнить r19 с $10
      brlo loop ; Перейти если r19 < $10 (без знака)
      pop ; Выйти из петли (пустая операция)

```

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRLT - Перейти если меньше чем (со знаком)

Описание:

Условный относительный переход. Тестируется бит флага знака (S) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число со знаком, представленное в Rd, меньше двоичного числа со знаком, представленного в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS 4,k).

Операция: (i) If Rd < Rr (NEV = 1) then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис      Операнды:      Счетчик программ:  
(i)      BRLT k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk      kkkk      k100

Булевы выражения регистра статуса (SREG)								
I	T	H	S	V	N	Z	C	
-	-	-	-	-	-	-	-	-

Пример:

```

cp r16, r1 ; Сравнить r16 с r1
brlt less ; Перейти если r16 < r1 (со знаком)
      .....
      less pop ; Перейти по назначению (пустая операция)

```

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRMI - Перейти если минус

Описание:

Условный относительный переход. Тестируется бит флага отрицательного значения (N) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBS r, k).

Операция:(i) If N = 1 then then PC <- PC + k +1, else PC <- PC +1

Синтаксис      Операнды:      Счетчик программ:  
(i)      BRMI k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk      kkkk      k010

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      -      -      -      -      -      -

Пример:

subi r18, 4 ; Вычесть 4 из r18

brmi negative ; Перейти если результат отрицательный

negative: nop      ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRNE - Перейти если не равно

Описание:

Условный относительный переход. Тестируется бит флага нулевого значения (Z) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число со знаком или без знака, представленное в Rd, не равно двоичному числу со знаком или без знака, представленному в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 1,k).

Операция:(i) If Rd № Rr (Z = 0) then then PC <- PC + k +1, else PC <- PC +1

Синтаксис      Операнды:      Счетчик программ:  
(i)      BRNE k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk      kkkk      k001

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      -      -      -      -      -      -

Пример:

eor r27, r27 ; Очистить r27

loop: inc r27 ; Увеличить на 1 r27

cpi r27, 5 ; Сравнить r27 с 5

bne loop ; Перейти если r27 ≠ 5

por ; Выйти из петли (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRPL - Перейти если плюс

Описание:

Условный относительный переход. Тестируется бит флага отрицательного значения (N) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC-64 < \text{назначение} < PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 2,k).

Операция:(i) If N = 0 then then PC <- PC + k +1, else PC <- PC +1

Синтаксис            Операнды:            Счетчик программ:  
(i)      BRPL k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk    kkkk    k010

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      -      -      -      -      -      -

Пример:

subi r26, \$50 ; Вычесть \$50 из r26

brpl positive ; Перейти если r26 положителен

positive: pop      ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов:1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRSH - Перейти если равно или больше (без знака)

Описание:

Условный относительный переход. Тестируется бит флага перехода (C) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Если команда выполняется непосредственно после выполнения любой из команд CP, CPI, SUB или SUBI переход произойдет если, и только если, двоичное число без знака, представленное в Rd, больше или равно двоичному числу без знака, представленному в Rr. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ ( $PC-64 < \text{назначение} < PC+63$ ). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 0,k).

Операция:(i) If Rd > Rr (C = 0) then then PC <- PC + k +1, else PC <- PC +1

Синтаксис            Операнды:            Счетчик программ:  
(i)      BRSH k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk    kkkk    k000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

subi r19, 4 ; Вычесть 4 из r19  
brsh highsm ; Перейти если r2 >= 4 (без знака)

highsm: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRTC - Перейти если флаг T очищен

Описание:

Условный относительный переход. Тестируется бит флага пересылки (T) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 6,k).

Операция: (i) If T = 0 then then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис      Операнды:      Счетчик программ:  
(i)      BRTC k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk      kkkk      k110

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

bst r3, 5 ; Сохранить бит 5 регистра r3 во флаге T  
brtc tclear ; Перейти если этот бит очищен

tclear: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRTS - Перейти если флаг T установлен

Описание:

Условный относительный переход. Тестируется бит флага пересылки (T) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 6,k).

Операция: (i) If T = 1 then then PC <- PC + k + 1, else PC <- PC + 1

Синтаксис      Операнды:      Счетчик программ:  
(i)      BRTS k -64 < k <+63      PC <- PC + k + 1  
PC <- PC + 1,  
если условия

не соблюдены

16-разрядный код операции: 1111 00kk kkkk k110

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

bst r3, 5 ; Сохранить бит 5 регистра r3 во флаге T  
brts tset ; Перейти если этот бит установлен

tset: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRVC - Перейти если переполнение очищено

Описание:

Условный относительный переход. Тестируется бит флага переполнения (V) регистра статуса и, если бит очищен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 < назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 3,k).

Операция: (i) If V = 0 then PC <- PC + k + 1, else PC <- PC + 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	BRVC k-64 < k <+63	PC <- PC + k + 1	

PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 01kk kkkk k011

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

add r3, r4 ; Сложить r4 с r3  
brvc noover ; Перейти если нет переполнения

noover: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BRVS - Перейти если переполнение установлено

Описание:

Условный относительный переход. Тестируется бит флага переполнения (V) регистра статуса и, если бит установлен, выполняется переход относительно состояния счетчика программ. Данная команда выполняет переход в любом направлении относительно состояния счетчика программ (PC-64 <

назначение < PC+63). Параметр k является смещением относительно состояния счетчика программ и представлен в форме дополнения до двух. (Команда эквивалентна BRBC 3,k).

Операция:(i) If V = 1 then then PC <- PC + k +1, else PC <- PC +1

Синтаксис            Операнды:            Счетчик программ:  
(i)        BRVS k -64 < k <+63        PC <- PC + k + 1  
PC <- PC + 1,  
если условия  
не соблюдены

16-разрядный код операции: 1111 00kk    kkkk    k011

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

add r3, r4 ; Сложит r4 с r3

brvs overfl ; Перейти если есть переполнение

overfl:    por        ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, 2 при соблюдении правильных условий

Команда BSET - Установить бит в регистре статуса (SREG)

Описание:

Установка одного флага в регистре статуса.

Операция:(i) SREG(s)<-- 1

Синтаксис            Операнды:            Счетчик программ:  
(i)        BSET s 0 < s < 7 PC <- PC + 1

16-разрядный код операции:  
1001    0100    0sss    1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
Ы	Ы	Ы	Ы	Ы	Ы	Ы	Ы

I:        1 если s = 7: в ином случае не изменяется

T:        1 если s = 6: в ином случае не изменяется

H:        1 если s = 5: в ином случае не изменяется

S:        1 если s = 4: в ином случае не изменяется

V:        1 если s = 3: в ином случае не изменяется

N:        1 если s = 2: в ином случае не изменяется

Z:        1 если s = 1: в ином случае не изменяется

C:        1 если s = 0: в ином случае не изменяется

Пример:

bset 6 ; Установить флаг T

bset 7 ; Разрешить прерывание

Слов: 1 (2 байта)

Циклов: 1

Команда BST - Переписать бит из регистра во флаг T регистра статуса (SREG)

Описание:

Перезапись бита b из регистра Rd в флаг T регистра статуса (SREG)

Операция: (i)  $T \leftarrow Rd(b)$

	Синтаксис	Операнды:	Счетчик программ:
(i)	BST Rd,b	$0 < d < 31, 0 < b < 7$	$PC \leftarrow PC + 1$

16-разрядный код операции:

1111 101d dddd Xbbb

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	Ы	-	-	-	-	-	-

T : 0 если бит b в Rd очищен; в ином случае устанавливается 1

Пример:

; Копировать бит  
bst r1, 2 ; Сохранить бит 2 регистра r1 во флаге T  
bld r0, 4 ; Загрузить T в бит 4 регистра r0

Слов: 1 (2 байта)

Циклов: 1

Команда CALL - Выполнить длинный вызов подпрограммы

Описание:

Вызов подпрограммы из памяти программ. Адрес возврата (к команде после CALL) сохраняется в стеке. (См. также RCALL).

Операция: (i)  $PC \leftarrow k$  Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128К.

(i)  $PC \leftarrow k$  Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8М.

	Синтаксис	Операнды:	Счетчик программ:
(i)	CALL k	$0 < k < 64K$	$PC \leftarrow kSTACK \leftarrow PC + 2$
	SP $\leftarrow$ SP-2, (2 байта, 16 битов)		
(i)	CALL k	$0 < k < 4M$	$PC \leftarrow kSTACK \leftarrow PC + 2$
	SP $\leftarrow$ SP-3, (3 байта, 22 бита)		

16-разрядный код операции:

1001 010k kkkk 111k  
kkkk kkkk kkkk kkkk

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

Пример:

```
mov r16, r0 ; Копировать r0 в r16
call check ; Вызвать подпрограмму
nop        ; Продолжать (пустая операция)
...
```

```
check:  cri r16, $42 ; Проверить содержит ли r16 заданное значение
        breq error ; Перейти если содержит
        get        ; Вернуться из подпрограммы
        ...
```

```
error:  rjmp error ; Бесконечная петля
```

Слов: 2 (4 байта)

Циклов: 4

.....

Команда CBI - Очистить бит в регистре I/O

Описание:

Очистка определенного бита в регистре ввода/вывода. Команда работает с младшими 32 регистрами ввода/вывода - адреса с 0 по 31.

Операция:(i) I/O(P,b)<-- 0

	Синтаксис	Операнды:	Счетчик программ:
(i)	CBI P,b 0 < P < 31, 0 < b < 7		PC <-- PC + 1

16-разрядный код операции:

1001 1000 pppp pbbb

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

```
cbi $12, 7 ; Очистить бит 7 в Порте D
```

Слов: 1 (2 байта)

Циклов: 2

.....

Команда CBR - очистить биты в регистре

Описание:

Очистка определенных битов регистра Rd. Выполняется логическое AND между содержимым регистра Rd и комплементом постоянной K

Операция:(i) Rd <-- Rd \* (\$FF - K)

	Синтаксис	Операнды:	Счетчик программ:
(i)	CBR Rd 16 < d < 31, 0 < K < 255		PC <-- PC + 1

16-разрядный код операции:

Смотри команду ANDI с комплементом K

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Ы 0 Ы Ы -

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

cbg r16, \$F0 ; Очистить старший ниббл регистра r16

cbg r18, 1 ; Очистить бит в r18

Слов: 1 (2 байта)

Циклов: 1

.....

Команда CLC - очистить флаг переноса в регистре статуса (SREG)

Описание:

Очистка флага переноса (C) в регистре статуса (SREG)

Операция:(i) C <-- 0

Синтаксис Операнды: Счетчик программ:

(i) CLC None PC <- PC + 1

16-разрядный код операции:

1001 0100 1000 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - 0

C: 0

Флаг переноса очищен

Пример:

add r0, r0 ; Сложить r0 с самим собой

clc ; Очистить флаг переноса

Слов: 1 (2 байта)

Циклов: 1

.....

Команда CLH - очистить флаг полупереноса в регистре статуса (SREG)

Описание:

Очистка флага полупереноса (H) в регистре статуса (SREG).

Операция:(i) H <-- 0

Синтаксис Операнды: Счетчик программ:

(i) CLH None PC <- PC + 1

16-разрядный код операции:

1001 0100 1101 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

-	-	0	-	-	-	-	-
---	---	---	---	---	---	---	---

H: 0

Флаг полупереноса очищен

Пример:

clh ; Очистить флаг полупереноса

Слов: 1 (2 байта)

Циклов: 1

Команда CLI - очистить флаг глобального прерывания в регистре статуса (SREG)

Описание:

Очистка флага глобального прерывания (I) в регистре статуса (SREG).

Операция:(i) I <-- 0

	Синтаксис	Операнды:	Счетчик программ:
(i)	CLI	None	PC <- PC + 1

16-разрядный код операции:

1001 0100 1111 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

0	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

I: 0

Флаг глобального прерывания очищен

Пример:

cli ; Запретить прерывания  
in r11, \$16 ; Читать Порт В  
sei ; Разрешить прерывания

Слов: 1 (2 байта)

Циклов: 1

Команда CLN - очистить флаг отрицательного значения в регистре статуса (SREG)

Описание:

Очистка флага отрицательного значения (N) в регистре статуса (SREG).

Операция:(i) N <-- 0

	Синтаксис	Операнды:	Счетчик программ:
(i)	CLN	None	PC <- PC + 1

16-разрядный код операции:

1001 0100 1010 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	0	-	-

N: 0  
 Флаг отрицательного значения очищен

Пример:  
 add r2, r3 ; Сложить r3 с r2  
 cln ; Очистить флаг отрицательного значения

Слов: 1 (2 байта)

Циклов: 1

Команда CLR - очистить регистр

Описание:

Очистка регистра. Команда выполняет Exclusive OR содержимого регистра с самим собой. Это приводит к очистке всех битов регистра.

Операция: (i) Rd <-- Rd ERd  
 Синтаксис      Операнды:      Счетчик программ:  
 (i) CLR Rd 0 < d < 31      PC <- PC + 1

16-разрядный код операции:  
 0010 01dd dddd dddd

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	0	0	0	1	-

S: 0

Очищен

V: 0

Очищен

N: 0

Очищен

Z: 1

Устанавливается

R: (Результат) соответствует Rd после выполнения команды

Пример:  
 clr r18 ; Очистить r18  
 loop: inc r18 ; Увеличить на 1 r18  
 ...  
 cmp r18, \$50 ; Сравнить r18 с \$50  
 brne loop

Слов: 1 (2 байта)

Циклов: 1

Команда CLS - очистить флаг знака

Описание:

Очистка флага знака (S) в регистре статуса (SREG).

Операция: (i) S <-- 0  
 Синтаксис      Операнды:      Счетчик программ:

(i) CLS None PC <- PC + 1

16-разрядный код операции:  
1001 0100 1100 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - 0 - - - -

S: 0

Очищен

Пример:

add r2, r3 ; Сложить r3 с r2

cls ; Очистить флаг знака

Слов: 1 (2 байта)

Циклов: 1

.....

Команда CLT - очистить T флаг

Описание:

Очистка флага пересылки (T) в регистре статуса (SREG).

Операция:(i) T <-- 0

Синтаксис Операнды: Счетчик программ:

(i) CLT None PC <- PC + 1

16-разрядный код операции:  
1001 0100 1110 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- 0 - - - - - -

T: 0

Очищен

Пример:

clt ; Очистить T флаг

Слов: 1 (2 байта)

Циклов: 1

.....

Команда CLV - очистить флаг переполнения

Описание:

Очистка флага переполнения (V) в регистре статуса (SREG).

Операция:(i) V <-- 0

Синтаксис Операнды: Счетчик программ:

(i) CLV None PC <- PC + 1

16-разрядный код операции:  
1001 0100 1011 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	0	-	-	-

V: 0  
Очищен

Пример:

add r2, r3 ; Сложить r3 с r2  
clv ; Очистить флаг переполнения

Слов: 1 (2 байта)

Циклов: 1

.....  
Команда CLZ - очистить флаг нулевого значения

Описание:

Очистка флага нулевого значения (Z) в регистре статуса (SREG).

Операция:(i) Z <-- 0

	Синтаксис	Операнды:	Счетчик программ:
(i)	CLZ None	PC <- PC + 1	

16-разрядный код операции:  
1001 0100 1001 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	0	-

Z: 0  
Очищен

Пример:

add r2, r3 ; Сложить r3 с r2  
clz ; Очистить флаг нулевого значения

Слов: 1 (2 байта)

Циклов: 1

.....  
Команда COM - Выполнить дополнение до единицы

Описание:

Команда выполняет дополнение до единицы (реализует обратный код) содержимого регистра Rd.

Операция:(i) Rd <-- \$FF \* Rd

	Синтаксис	Операнды:	Счетчик программ:
(i)	COM Rd	0 < d < 31	PC <- PC + 1

16-разрядный код операции:

1001 010d dddd 0000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Ы 0 Ы Ы 1

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в ином случае очищается

C: 1

Установлен

R: (Результат) соответствует Rd после выполнения команды

Пример:

com r4 ; Выполнить дополнение до единицы r4

breq zero ; Перейти если ноль

zero: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда CP - сравнить

Описание:

Команда выполняет сравнение содержимого двух регистров Rd и Rr. Содержимое регистров не изменяется. После этой команды можно выполнять любые условные переходы.

Операция: (i)  $Rd = Rr$

Синтаксис Операнды: Счетчик программ:  
(i) Cp Rd, Rr  $0 < d < 31, 0 < r < 31$  PC  $\leftarrow$  PC + 1

16-разрядный код операции:

0001 01rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H:  $Rd3 * Rr3 + Rr3 * R3 + R3 * Rd3$

Устанавливается если есть заем из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * Rd7 * R7 + Rd7 * Rr7 * R7$

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $Rd7 * Rr7 + Rr7 * R7 + R7 * Rd7$

Устанавливается если результат \$00, в ином случае очищается

C:  $Rd7 * Rr7 + Rr7 * R7 + R7 * Rd7$

Устанавливается если абсолютное значение Rr больше абсолютного значения Rd, в ином случае очищается

R: (Результат) после выполнения команды

Пример:

sr r4, r19 ; Сравнить r4 с r19  
bne noteq ; Перейти если r4 < r19

noteq: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда CPC - сравнить с учетом переноса

Описание:

Команда выполняет сравнение содержимого двух регистров Rd и Rr и учитывает также предшествовавший перенос. Содержимое регистров не изменяется. После этой команды можно выполнять любые условные переходы.

Операция: (i) Rd = Rr = C

Синтаксис Операнды: Счетчик программ:  
(i) CPC Rd, Rr 0 < d < 31, 0 < r < 31 PC ← PC + 1

16-разрядный код операции:

0000 01rd dddd rrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H: Rd3\*Rr3+Rr3\*R3+R3\*Rd3

Устанавливается если есть заем из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V: Rd7\*Rd7\*R7+Rd7\*Rr7\*R7

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0\*Z

Предшествующее значение остается неизменным если результатом является ноль, в ином случае очищается

C: Rd7\*Rr7+Rr7\*R7+R7\*Rd7

Устанавливается если абсолютное значение Rr плюс предшествовавший перенос больше абсолютного значения Rd, в ином случае очищается

R: (Результат) после выполнения команды

Пример:

; Сравнить r3 : r2 с r1 : r0  
sr r2, r0 ; Сравнить старший байт  
src r3, r1 ; Сравнить младший байт  
bne noteq ; Перейти если не равно

noteq: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда CPI - сравнить с константой

Описание:

Команда выполняет сравнение содержимого регистра Rd с константой. Содержимое регистра не изменяется. После этой команды можно выполнять любые условные переходы.

Операция: (i) Rd = K

Синтаксис Операнды: Счетчик программ:  
(i) CPI Rd, K 16 =< d =< 31, 0 < K <255 PC <- PC + 1

16-разрядный код операции:  
0011 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H: Rd3\*K3+K3\*R3+R3\*Rd3

Устанавливается если есть заем из бита 3, в противном случае очищается

S: NEV, Для проверок со знаком

V: Rd7\*K7\*R7+ Rd7\*K7\*R7

Устанавливается если в результате операции образуется переполнение дополнения до двух, в противном случае очищается

N: R7

Устанавливается если в результате установлен MSB, в противном случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в противном случае очищается

C: Rd7\*K7+K7\*R7+R7\*Rd7

Устанавливается если абсолютное значение K больше абсолютного значения Rd, в противном случае очищается

R: (Результат) после выполнения команды

Пример:

срi r19, 3 ; Сравнить r19 с 3

bnr error ; Перейти если r4 <> 3

error: por ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда CPSE - сравнить и пропустить если равно

Описание:

Команда выполняет сравнение содержимого регистров Rd и Rr и пропускает следующую команду если Rd = Rr.

Операция: (i) If Rd = Rr then PC <-- PC + 2 (or 3), else PC <-- PC + 1

Синтаксис Операнды: Счетчик программ:  
(i) CPSE Rd,Rr 0 < d <31, 0 < r<31 PC<-- PC + 1, если

условия не соблюдены,

то пропуска нет

PC<-- PC + 2, пропуск

одного слова команды

PC<-- PC + 3, пропуск

двух слов команды

16-разрядный код операции:  
0001 00rd dddd rrrr

Булевы выражения регистра статуса (SREG)  
 I T H S V N Z C  
 - - - - - - - -

Пример:  
 inc r4 ; Увеличить на 1 r4  
 cpse r4, r0 ; Сравнить r4 с r0  
 neg r4 ; Выполнить если r4 <> r0  
 nop ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда DEC - декрементировать

Описание:

Вычитание единицы - 1 - из содержимого регистра Rd и размещение результата в регистре назначения Rd. Флаг переноса регистра статуса данной командой не активируется, что позволяет использовать команду DEC использовать при реализации счетчика циклов для вычислений с повышенной точностью. При обработке чисел без знаков за командой могут выполняться переходы BREQ и BRNE. При обработке значений в форме дополнения до двух допустимы все учитывающие знак переходы.

Операция:(i) Rd<-- Rd - 1

Синтаксис Операнды: Счетчик программ:  
 (i) DEC Rd 0 < d < 31 PC <- PC + 1

16-разрядный код операции:  
 1001 010d dddd 1010

Булевы выражения регистра статуса (SREG)  
 I T H S V N Z C  
 - - - Ы Ы Ы Ы -

S: NEV, Для проверок со знаком  
 V: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если в результате получено переполнение дополнения до двух, в ином случае очищается. Переполнение дополнения до двух будет если и только если перед операцией содержимое Rd было \$80.

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:  
 ldi r17, \$10 ; Загрузить константу в r17  
 loop: add r1, r2 ; Сложить r2 с r1  
 dec r17 ; Уменьшить на 1 r17  
 brne loop ; Перейти если r17 <> 0  
 nop ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда EOR - выполнить исключающее OR

Описание:

Выполнение логического исключающего OR между содержимым регистра Rd и регистром Rr и помещение результата в регистр назначения Rd.

Операция:(i) Rd<-- RdERr

Синтаксис Операнды: Счетчик программ:  
 (i) EOR Rd,Rr 0 < d < 31, 0 < r < 31 PC <- PC + 1

16-разрядный код операции:  
0010 01rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Ы 0 Ы Ы -

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в другом случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в другом случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

eor r4, r4 ; Очистить r4

eor r0, r22 ; Поразрядно выполнить исключающее ор между r0 и r22

Слов: 1 (2 байта)

Циклов: 1



Команда ICALL - вызвать подпрограмму косвенно

Описание:

Косвенный вызов подпрограммы указанной регистром-указателем Z (16 разрядов) в регистровом файле. Регистр-указатель Z (16-разрядного формата) позволяет вызвать подпрограмму из текущей секции пространства памяти программ объемом 64К слов (128 Кбайт).

Операция:(i) PC(15-0)<-- Z(15-0) Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128К.

(i) PC(15-0)<-- Z(15-0) Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8М. PC(21-16) не изменяются

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	ICALL None	См. Операция	STACK<-- PC + 1	
	SP<-- SP-2, (2 байта, 16 битов)			
(i)	ICALL None	См. Операция	STACK<-- PC + 1	
	SP<-- SP-3, (3 байта, 22 бита)			

16-разрядный код операции:  
1001 0101 XXXX 1001

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

mov r30, r0 ; Установить смещение в таблицу вызовов

icall ; Вызвать подпрограмму указанную r31 : r30

Слов: 1 (2 байта)

Циклов: 3



Команда IJMP - перейти косвенно

Описание:

Выполняется косвенный переход по адресу указанному регистром-указателем Z (16 разрядов) в регистровом файле. Регистр-указатель Z (16-разрядного формата) позволяет вызвать подпрограмму из текущей секции пространства памяти программ объемом 64К слов (128 Кбайт).

Операция:(i) PC<-- Z(15-0) Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128К.

(ii) PC(15-0)<-- ZZ(15-0) Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8М. PC(21-16) не изменяются

	Синтаксис	Операнды:	Счетчик программ:	Стек
(ii)	IJMP None	См. Операция	Не задействуется	
(iii)	ICALL None	См. Операция	Не задействуется	

16-разрядный код операции:  
1001 0100 XXXX 1001

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

mov r30, r0 ; Установить смещение в таблицу переходов  
ijmp ; Перейти к подпрограмме указанной r31 : r30

Слов: 1 (2 байта)

Циклов: 2

.....

Команда IN - загрузить данные из порта I/O в регистр

Описание:

Команда загружает данные из пространства входа/выхода (порты, таймеры, регистры конфигурации и т.п.) в регистр Rd регистрового файла.

Операция:(i) Pd<-- P

	Синтаксис	Операнды:	Счетчик программ:
(i)	IN Rd,P	0 < d < 31, 0 < P < 63	PC<-- PC + 1

16-разрядный код операции:  
1011 0PPd dddd PPPP

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

in r25, \$16 ; Считать Порт В  
cpi r25, r4 ; Сравнить считанное значение с константой  
breq exit ; Перейти если r25=4

exit: por ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

.....

Команда INC - инкрементировать

Описание:

Добавление единицы - 1 - к содержимому регистра Rd и размещение результата в регистре назначения Rd. Флаг переноса регистра статуса данной командой не активируется, что позволяет использовать команду DEC использовать при реализации счетчика циклов для вычислений с повышенной точностью. При обработке чисел без знаков за командой могут выполняться переходы BREQ и BRNE. При обработке значений в форме дополнения до двух допустимы все учитывающие знак переходы.

Операция:(i) Rd<-- Rd + 1

(i) Синтаксис: INC Rd 0 < d < 31  
Операнды: PC <- PC + 1  
Счетчик программ:

16-разрядный код операции:  
1001 010d dddd 0011

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	Ы	Ы	Ы	-

S: NEV, Для проверок со знаком

V: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если в результате получено переполнение дополнения до двух, в ином случае очищается. Переполнение дополнения до двух будет если и только если перед операцией содержимое Rd было \$7F.

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

```

clr r22 ; Очистить r22
loop: inc r22 ; Увеличить на 1 r22
      ...
      cpi r22, $4F ; Сравнить r22 с $4F
      bne loop ; Перейти если не равно
      nop ; Продолжать (пустая операция)

```

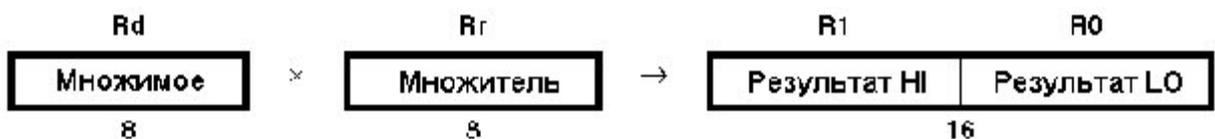
Слов: 1 (2 байта)

Циклов: 1

Команда FMUL - Дробное незнаковое умножение

Описание:

Эта инструкция выполняет 8-бит x 8-бит --> 16-бит знаковое умножение и сдвиг результата на 1 бит влево.



Пусть (N.Q) обозначают дробное число с N двоичных цифр слева от точки, и Q двоичных цифр справа от точки. Умножение между двумя числами в форматах (N1.Q1) и (N2.Q2) даст результат в формате ((N1 + N2). (Q1 + Q2)).

Формат (1.7) широко используется как входные числа для сигнальной обработки, а получившийся в результате формат (2.14) - как результат. Сдвиг влево необходим для того, что бы старший байт результата был в таком же формате как входные числа. Инструкция FMULS, включая операцию сдвига, выполняется за такое же число циклов как инструкция MUL.

Формат (1,7) наиболее широко используется со знаковыми числами, а инструкция FMUL выполняет незнаковое умножение. Эта инструкция наиболее полезна при вычислении части результата при выполнении умножения со знаком 16-битных чисел в формате (1.15), дающие результат в формате (1,31).

Примечание: результат операции FMUL может подвергнуться двойному переполнению, если интерпретировать число в формате (1.15). MSB умножения, до сдвига, должно быть учтено, и находится в бите переноса. См. следующий пример.

Rd множимое и множитель Rr - два регистры, содержащие знаковые дробные числа, у которых мнимая точка расположена между битом 6 и битом 7.

16-битный знаковый дробный результат с мнимой точкой между битом 14 и битом 15 помещается в R1 (старший байт) и R0 (младший байт).

Эта инструкция не доступна во всех устройствах. Обратитесь к сводной инструкции конкретного устройства.

Операция:(i) R1:R0 <- Rd x Rr (знаковое (1.15) <- знаковое (1.7) x знаковое (1.7))

	Синтаксис	Операнды:	Счетчик программ:
(i)	FMUL Rd,Rr	16 d 23, 16 r 23	PC <- PC + 1

16-разрядный код операции: 0000 0011 0ddd 1rrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	+	+

C: R16

Установлен, если бит 15 результата до сдвига в лево установлен, очищен в ином случае.

Z: R15 · R14 · R13 · R12 · R11 · R10 · R9 · R8 · R7 · R6 · R5 · R4 · R3 · R2 · R1 · R0

Установлен, если результат равен \$0000, очищен в ином случае.

Пример:

```

;*****
;* ОПИСАНИЕ
;* Знаковое дробное умножение двух 16 битных чисел с 32 битным результатом.
;* Используются:
;* r19:r18:r17:r16 = ( r23:r22 * r21:r20 ) << 1
;*****
fmuls16x16_32:
clr2
fmulsr23, r21;((знаковое)ah * (знаковое)bh) << 1
movw r19:r18, r1:r0
fmulr22, r20;(al * bl) << 1
adcr18, r2
movw r17:r16, r1:r0
fmulsur23, r20;(( знаковое)ah * bl) << 1
sbc r19, r2
addr17, r0
adcr18, r1
adcr19, r2
fmulsur21, r22;(( знаковое)bh * al) << 1
sbc r19, r2
addr17, r0
adcr18, r1
adcr19, r2

```

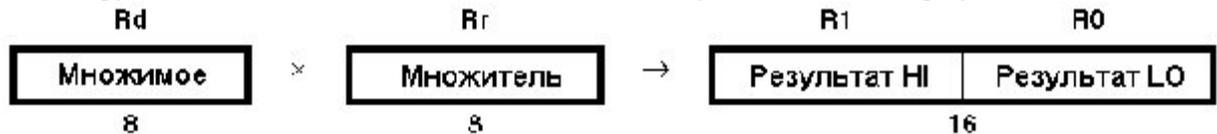
Слов: 1 (2 байта)

Циклов:2

Команда FMULS - Дробное умножение со знаком

Описание:

Эта инструкция выполняет 8-бит x 8-бит -> 16-бит знаковое умножение и сдвиг результата на 1 бит влево.



Пусть (N.Q) обозначают дробное число с N двоичных цифр слева от точки, и Q двоичных цифр справа от точки. Умножение между двумя числами в форматах (N1.Q1) и (N2.Q2) даст результат в формате ((N1 + N2). (Q1 + Q2)). Формат (1.7) широко используется как формат входных чисел для сигнальной обработки, а получившийся в результате формат (2.14) - как результат. Сдвиг влево необходим для того, чтобы старший байт результата был в таком же формате как входные числа. Инструкция FMULS, включая операцию сдвига, выполняется за такое же число циклов как инструкция MULS.

Множимое Rd и множитель Rr - два регистра, содержащих знаковые дробные числа, у которых мнимая точка расположена между битом 6 и битом 7. 16-битный знаковый дробный результат с мнимой точкой между битом 14 и битом 15 помещается в R1 (старший байт) и R0 (младший байт).

Заметим, что, когда умножаются 0x80 (-1) с 0x80 (-1), результатом операции сдвига будет 0x8000 (-1). Операции сдвига, таким образом, даёт два дополнительных переполнения. Это должны быть проверено с помощью программного обеспечения.

Эта инструкция не доступна во всех устройствах. Обратитесь к сводной инструкции конкретного устройства.

Операция:(i) R1:R0 <-- Rd x Rr (знаковое (1.15) <-- знаковое (1.7) x знаковое (1.7))

	Синтаксис	Операнды:	Счетчик программ:
(i)	FMULS Rd,Rr	16 d 23, 16 r 23	PC <- PC + 1

16-разрядный код операции:0000 0011 1ddd 0rrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	+	+

C: R16

Установлен, если бит 15 результата до сдвига в лево установлен, очищен в ином случае.

Z: R15 · R14 · R13 · R12 · R11 · R10 · R9 · R8 · R7 · R6 · R5 · R4 · R3 · R2 · R1 · R0

Установлен, если результат равен \$0000, очищен в ином случае.

R: (результат) соответствует R1,R0 после выполнения команды.

Пример:

fmuls r23,r22 ; Умножить знаковые r23 и r22 в формате (1.7), результат в формате (1.15)  
movw r23:r22,r1:r0 ; Копируем результат обратно в r23:r22

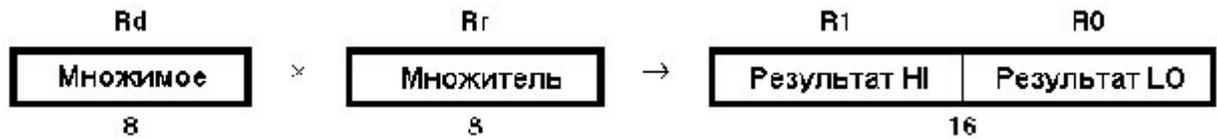
Слов: 1 (2 байта)

Циклов:2

Команда FMULSU - Дробное умножение знакового с незнаковым

Описание:

Эта инструкция выполняет 8-бит x 8-бит --> 16-бит знаковое умножение и сдвиг результата на 1 бит влево.



П

усть (N.Q) обозначают дробное число с N двоичных цифр слева от точки, и Q двоичных цифр справа от точки. Умножение между двумя числами в форматах (N1.Q1) и (N2.Q2) даст результат в формате ((N1 + N2). (Q1 + Q2)). Формат (1.7) широко используется как формат входных чисел для сигнальной обработки, а получившийся в результате формат (2.14) - как результат. Сдвиг влево необходим для того, чтобы старший байт результата был в таком же формате как входные числа. Инструкция FMULSU, включая операцию сдвига, выполняется за такое же число циклов как инструкция MULSU.

Формат (1,7) наиболее широко используется со знаковыми числами, в то время как FMULSU выполняет умножение одного незнакового и одного знакового числа. Поэтому эта инструкция наиболее полезна для расчета двух частичных результатов при выполнении знакового умножения с 16-битными данными в формате (1.15), дающие результат в формате (1.31).

Примечание: результат операции FMULSU может подвергнуться двойному переполнению, если интерпретировать число в формате (1.15). MSB умножения, до сдвига, должно быть учтено, и находится в бите переноса. См. следующий пример.

Множимое Rd и множитель Rr - два регистра, содержащих знаковые дробные числа, у которых мнимая точка расположена между битом 6 и битом 7. Множимое Rd - знаковое дробное число, и множитель Rr - незнаковое дробное число. 16-битный знаковый дробный результат с мнимой точкой между битом 14 и битом 15 помещается в R1 (старший байт) и R0 (младший байт).

Эта инструкция не доступна во всех устройствах. Обратитесь к сводной инструкции конкретного устройства.

Операция: R1:R0 <-- Rd x Rr (знаковое (1.15) <-- знаковое (1.7) x незнаковое (1.7))

Синтаксис      Операнды:      Счетчик программ:  
FMULSU Rd,Rr 16 d 23, 16 r 23      PC <- PC + 1

16-разрядный код операции: 0000 0011    1ddd    1rrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	+	+

C: R16

Установлен, если бит 15 результата до сдвига в лево установлен, очищен в ином случае.

Z: R15 · R14 · R13 · R12 · R11 · R10 · R9 · R8 · R7 · R6 · R5 · R4 · R3 · R2 · R1 · R0

Установлен, если результат равен \$0000, очищен в ином случае.

R: (результат)соответствует R1,R0 после выполнения команды.

Пример:

```

;*****
;* ОПИСАНИЕ
;* Знаковое дробное умножение двух 16-битных чисел и 32-битным результатом.
;* Используются:
;* r19:r18:r17:r16 = ( r23:r22 * r21:r20 ) << 1
;*****
fmuls16x16_32:

```

```

clrr2
fmulsr23, r21;((знаковый)ah * (знаковый)bh) << 1
movwr19:r18, r1:r0
fmulr22, r20;(al * bl) << 1
adcr18, r2
movwr17:r16, r1:r0
fmulsur23, r20;((знаковый)ah * bl) << 1
sbcr19, r2
addr17, r0
adcr18, r1
adcr19, r2
fmulsur21, r22;((знаковый)bh * al) << 1
sbcr19, r2
addr17, r0
adcr18, r1
adcr19, r2

```

Слов: 1 (2 байта)

Циклов:2

Команда JMP - перейти

Описание:

Выполняется переход по адресу внутри всего объема (4М слов) памяти программ. См. также команду RJMP.

Операция:(i) Pd<-- k

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	JMP k	0 < k < 4М	PC<-- k	Не изменяется

32-разрядный код операции:

```

1001 010k kkkk 110k
kkkk kkkk kkkk kkkk

```

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```

mov r1, r0 ; Копировать r0 в r1
jmp farplc ; Безусловный переход

```

farplc: nop ; Перейти по назначению (пустая операция)

Слов: 2 (4 байта)

Циклов: 4

Команда LD - загрузить косвенно из СОЗУ в регистр с использованием индекса X

Описание:

Загружает косвенно один байт из СОЗУ в регистр. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем X в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPX в I/O области. Регистр-указатель X может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Использование регистра-указателя X обеспечивает удобную возможность обращения к матрицам, таблицам, указателю стека.

	Использование X-указателя:	Операция:	Комментарий:
(i)	Rd <-- (X)	X	Неизменен
(ii)	Rd <-- (X)	X <-- X + 1	X: Инкрементирован впоследствии

(iii)	$X \leftarrow X - 1$	$Rd \leftarrow (X)$	X: Предварительно декрементирован
	Синтаксис	Операнды:	Счетчик программ:
(i)	LD Rd,X	$0 < d < 31$	PC $\leftarrow$ + 1
(ii)	LD Rd,X+	$0 < d < 31$	PC $\leftarrow$ + 1
(iii)	LDD Rd,-X	$0 < d < 31$	PC $\leftarrow$ + 1

16-разрядный код операции:

(i)	1001	000d	dddd	1100
(ii)	1001	000d	dddd	1101
(iii)	1001	000d	dddd	1110

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```

clr r27 ;Очистить старший байт X
ldi r26, $20 ;Установить $20 в младший байт X
ld r0, X+ ;Загрузить в r0 содержимое SRAM по адресу $20 (X постинкрементируется)
ld r1, X ;Загрузить в r1 содержимое SRAM по адресу $21
ldi r26, $23 ;Установить $23 в младший байт X
ld r2, X ;Загрузить в r2 содержимое SRAM по адресу $23
ld r3, -X ;Загрузить в r3 содержимое SRAM по адресу $22 (X преддекрементируется)

```

Слов: 1 (2 байта)

Циклов: 2

Команда LD (LDD) - загрузить косвенно из СОЗУ в регистр с использованием индекса Y

Описание:

Загружает косвенно, со смещением или без смещения, один байт из СОЗУ в регистр. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем Y в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPY в I/O области. Регистр-указатель Y может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Использование регистра-указателя Y обеспечивает удобную возможность обращения к матрицам, таблицам, указателю стека.

	Использование Y-указателя:	Операция:	Комментарий:
(i)	$Rd \leftarrow (Y)$	$Y$	Y: Неизменен
(ii)	$Rd \leftarrow (Y)$	$Y \leftarrow Y + 1$	Y: Инкрементирован впоследствии
(iii)	$Y \leftarrow Y + 1$	$Rd \leftarrow (Y)$	Y: Предварительно декрементирован
(iv)	$Rd \leftarrow (Y + q)$	$Y$	Y: Неизменен, q: смещение

	Синтаксис	Операнды:	Счетчик программ:
(i)	LD Rd,Y	$0 < d < 31$	PC $\leftarrow$ + 1
(ii)	LD Rd,Y+	$0 < d < 31$	PC $\leftarrow$ + 1
(iii)	LD Rd,-Y	$0 < d < 31$	PC $\leftarrow$ + 1
(iv)	LDD Rd, Y + q	$0 < d < 31$	
	$0 < q < 63$	PC $\leftarrow$ + 1	

16-разрядный код операции:

(i)	1000	000d	dddd	1000
(ii)	1001	000d	dddd	1001
(iii)	1001	000d	dddd	1010
(iv)	10q0	qq0d	dddd	1qqq

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

Пример:

```
clr r29 ;Очистить старший байт Y
ldi r28, $20 ;Установить $20 в младший байт Y
ld r0, Y+ ;Загрузить в r0 содерж. SRAM по адресу $20 (Y постинкрементируется)
ld r1, Y ;Загрузить в r1 содержимое SRAM по адресу $21
ldi r28, $23 ;Установить $23 в младший байт Y
ld r2, Y ;Загрузить в r2 содержимое SRAM по адресу $23
ld r3, -Y ;Загрузить в r3 содерж. SRAM по адресу $22 (Y преддекрементируется)
ldd r4, Y+2 ;Загрузить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта)

Циклов: 2

Команда LD (LDD) - загрузить косвенно из СОЗУ в регистр с использованием индекса Z

Описание:

Загружает косвенно, со смещением или без смещения, один байт из СОЗУ в регистр. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем Z в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPZ в I/O области. Регистр-указатель Z может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Эта особенность очень удобна при использовании регистра-указателя Z в качестве указателя стека, однако, поскольку регистр-указатель Z может быть использован для косвенного вызова подпрограмм, косвенных переходов и табличных преобразований, более удобно использовать в качестве указателя стека регистры-указатели X и Y. Об использовании указателя Z для просмотра таблиц в памяти программ см. команду LPM.

Использование Z-указателя:	Операция:	Комментарий:
(i) Rd <-- (Y)	Y: Неизменен	
(ii) Rd <-- (Y) Y <-- Y + 1	Y: Инкрементирован впоследствии	
(iii) Y <-- Y + 1 Rd <-- (Y)	Y: Предварительно декрементирован	
(iv) Rd <-- (Y + q)	Y: Неизменен, q: смещение	

	Синтаксис	Операнды:	Счетчик программ:
(i)	LD Rd, Y	0 < d < 31	PC<-- + 1
(ii)	LD Rd, Y+	0 < d < 31	PC<-- + 1
(iii)	LD Rd, -Y	0 < d < 31	PC<-- + 1
(iv)	LDD Rd, Y + q	0 < d < 31	

0 < q < 63 PC<-- + 1

16-разрядный код операции:

(i)	1000	000d	dddd	1000
(ii)	1001	000d	dddd	1001
(iii)	1001	000d	dddd	1010
(iv)	10q0	qq0d	dddd	1qqq

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

Пример:

```
clr r29 ;Очистить старший байт Y
ldi r28, $20 ;Установить $20 в младший байт Y
ld r0, Y+ ;Загрузить в r0 содерж. SRAM по адресу $20 (Y постинкрементируется)
ld r1, Y ;Загрузить в r1 содержимое SRAM по адресу $21
ldi r28, $23 ;Установить $23 в младший байт Y
ld r2, Y ;Загрузить в r2 содержимое SRAM по адресу $23
ld r3, -Y ;Загрузить в r3 содерж. SRAM по адресу $22 (Y преддекрементируется)
ldd r4, Y+2 ;Загрузить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта)

Циклов: 2

Команда LDI - загрузить непосредственное значение

Описание:

Загружается 8-разрядная константа в регистр от 16 по 31

Операция: (i) Rd <-- K  
Синтаксис      Операнды:      Счетчик программ:  
(i)      LDI Rd, K      16 < d < 31, 0 < K < 255      PC<-- + 1

16-разрядный код операции:  
1110    KKKK    dddd    KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

clr r31      ; Очистить старший байт Z  
ldi r30, \$F0 ; Установить \$F0 в младший байт Z  
lpm          ; Загрузить константу из программы  
             ; Память отмечена в Z

Слов: 1 (2 байта)

Циклов: 1

Команда LDS - загрузить непосредственно из СОЗУ

Описание:

Выполняется загрузка одного байта из СОЗУ в регистр. Можно использовать 16-разрядный адрес. Обращение к памяти ограничено текущей страницей СОЗУ объемом 64 Кбайта. Команда LDS использует для обращения к памяти выше 64 Кбайт регистр RAMPZ.

Операция: (i) Rd <-- (k)  
Синтаксис      Операнды:      Счетчик программ:  
(i)      LDS Rd,k      0 < d < 31, 0 < k < 65535      PC<-- + 2

32-разрядный код операции:  
1001    000d    dddd    0000  
kkkk    kkkk    kkkk    kkkk

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

lds r2, \$FF00 ; Загрузить r2 содержимым SRAM по адресу \$FF00  
add r2, r1    ; Сложить r1 с r2  
sts \$FF00, r2 ; Записать обратно

Слов: 2 (4 байта)

Циклов: 3

Команда LPM - загрузить байт памяти программ

Описание:

Загружает один байт, адресованный регистром Z, в регистр 0 (R0). Команда обеспечивает эффективную загрузку констант или выборку постоянных данных. Память программ организована из 16-разрядных слов и младший значащий разряд (LSB) 16-разрядного указателя Z выбирает или младший (0) или старший (1) байт. Команда может адресовать первые 64 Кбайта (32 Кслов) памяти программ.

Операция: Комментарий:  
(i) R0<-- (Z) Z указывает на память программ

Синтаксис Операнды: Счетчик программ:  
(i) LPM None PC<-- + 1

16-разрядный код операции:  
1001 0101 110X 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

```
clr r31 ; Очистить старший байт Z
ldi r30, $F0 ; Установить младший байт Z
lpm ; Загрузить константу из памяти программ
отмеченную Z (r31 : r30)
```

Слов: 1 (2 байта)

Циклов: 3

Команда LSL - логически сдвинуть влево

Описание:

Выполнение сдвига всех битов Rd на одно место влево. Бит 0 стирается. Бит 7 загружается во флаг переноса (C) регистра состояния (SREG). Эта команда эффективно умножает на два значение величины без знака.



Операция:

16-разрядный код операции:  
0000 11dd dddd dddd

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H: Rd3

S: NEV, Для проверок со знаком

V: NEC (Для N и C после сдвига)

Устанавливается если (N устанавливается и C очищается) или (N очищается а C устанавливается). В ином случае очищается (при наличии значений N и C после сдвига)

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

C: Rd7

Устанавливается если перед сдвигом был установлен MSB регистра Rd в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

add r0, r4 ; Сложить r4 с r0  
lsl r0 ; Умножить r0 на 2

Слов: 1 (2 байта)

Циклов: 1

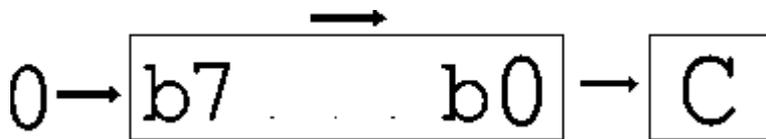
.....

Команда LSR - логически сдвинуть вправо

Описание:

Сдвиг всех битов Rd на одно место вправо. Бит 7 очищается. Бит 0 загружается во флаг переноса (C) регистра состояния (SREG). Эта команда эффективно делит на два величину без знака на два. Флаг переноса может быть использован для округления результата.

Операция:



16-разрядный код операции:  
1001 010d dddd 0110

Булевы выражения регистра статуса (SREG)

I	T	N	S	V	N	Z	C
---	---	---	---	---	---	---	---

-	-	-	Ы	Ы	0	Ы	Ы
---	---	---	---	---	---	---	---

S: NEV, Для проверок со знаком

V: NEC (Для N и C после сдвига)

Устанавливается если (N устанавливается и C очищается) или (N очищается а C устанавливается). В ином случае очищается (при наличии значений N и C после сдвига)

N: 0

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в ином случае очищается

C: Rd0

Устанавливается если перед сдвигом был установлен LSB регистра Rd, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

add r0, r4 ; Сложить r4 с r0  
lsr r0 ; Разделить r0 на 2

Слов: 1 (2 байта)

Циклов: 1

.....

Команда MOV - копировать регистр

Описание:

Команда создает копию одного регистра в другом регистре. Исходный регистр Rr остается неизменным, в регистр назначения Rd загружается копия содержимого регистра Rr.(i) Rd <-- Rr

Синтаксис      Операнды:      Счетчик программ:  
(i)      MOV Rd,Rr      0 < d < 31, 0 < r < 31      PC<-- + 1

16-разрядный код операции:

0010 11rd dddd rrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

```
mov r16, r0 ; Копировать r0 в r16
call check ; Вызвать подпрограмму
...
check cpi r16, $11 ; Сравнить r16 с $11
...
ret ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 1

.....

Команда MUL - перемножить

Описание:

Команда перемножает две 8-разрядные величины без знаков с получением 16-разрядного результата без знака. Множимое и множитель - два регистра - Rr и Rd, соответственно. 16-разрядное произведение размещается в регистрах R1 (старший байт) и R0 (младший байт). Отметим, что если в качестве множимого и множителя выбрать R0 или R1, то результат заместит прежние значения сразу после выполнения операции.

Операция:(i) R1,R0 <-- Rr x Rd

Синтаксис Операнды: Счетчик программ:  
(i) MUL Rd,Rr 0 < d < 31, 0 < r < 31 PC <- PC + 1

16-разрядный код операции:

1001 11rd dddd rrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - Ы

C: R15

Устанавливается если установлен бит 15 результата, в ином случае очищается

R: (Результат)соответствует R1,R0 после выполнения команды

Пример:

```
mul r6, r5 ; Перемножить r6 и r5
mov r6, r1 ; Вернуть результат обратно в r6:r5
mov r5, r1 ; Вернуть результат обратно в r6:r5
```

Слов: 1 (2 байта)

Циклов: 2

В системе команд базовых микроконтроллеров семейства команда отсутствует.

.....

Команда NEG - выполнить дополнение до двух

Описание:

Заменяет содержимое регистра Rd его дополнением до двух. Значение \$80 остается неизменным.

Операция:(i) Rd <-- \$00 - Rd

Синтаксис      Операнды:      Счетчик программ:  
(i)      NEG Rd0 < d < 31      PC <- PC + 1

16-разрядный код операции:  
1001    010d    dddd    0001

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      Ы      Ы      Ы      Ы      Ы      Ы

H:      R3\*Rd3

Устанавливается если есть заем из бита 3, в ином случае очищается

S:      NEV, Для проверок со знаком

V:      R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается при переполнении дополнения до двух от подразумеваемого вычитания из нуля, в ином случае очищается. Переполнение дополнения до двух произойдет если и только если содержимое регистра после операции (результат) будет \$80.

N:      R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:      Rd7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

C:      R7+R6+R5+R4+R3+R2+R1+R0

Устанавливается если есть заем в подразумеваемом вычитании из нуля, в ином случае очищается. Флаг C будет устанавливаться во всех случаях, за исключением случая, когда содержимое регистра после выполнения операции будет \$80.

R:      (Результат)соответствует Rd после выполнения команды

Пример:

sub r11, r0 ; Вычесть r0 из r11

brpl positive ; Перейти если результат положительный

neg r11 ; Выполнить дополнение до двух r11

positive: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

.....

Команда NOP - выполнить холостую команду

Описание:

Команда выполняется за один цикл без выполнения операции.(i) No

Синтаксис      Операнды:      Счетчик программ:

(i)      NOP    None    PC<-- + 1

16-разрядный код операции:  
0000    0000    0000    0000

Булевы выражения регистра статуса (SREG)

I      T      H      S      V      N      Z      C

-      -      -      -      -      -      -      -

Пример:

clr r16 ; Очистить r16

ser r17 ; Установить r17

out \$18, r16 ; Записать ноль в Порт В

por ; Ожидать (пустая операция)

out \$18, r17 ; Записать 1 в Порт В

Слов: 1 (2 байта)

Циклов: 1

Команда OR - выполнить логическое OR

Описание:

Команда выполняет логическое OR содержимого регистров Rd и Rr и размещает результат в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd \vee Rr$

	Синтаксис	Операнды:	Счетчик программ:
(i)	OR Rd,Rr	$0 < d < 31, 0 < r < 31$	PC $\leftarrow$ PC + 1

16-разрядный код операции:

0010 10rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Ы 0 Ы Ы -

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в противном случае очищается

Z:  $Rd7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в противном случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

or r15, r16 ; Выполнить поразрядное or между регистрами

bst r15, 6 ; Сохранить бит 6 регистра 15 во флаге T

brst ok ; Перейти если флаг T установлен

...

ok: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда ORI –

Описание:

Команда выполняет логическое OR между содержимым регистра Rd и константой и размещает результат в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd \vee K$

	Синтаксис	Операнды:	Счетчик программ:
(i)	ORI Rd,K	$16 < d < 31, 0 < K < 255$	PC $\leftarrow$ PC + 1

16-разрядный код операции:

0110 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - Б 0 Б Б -

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: Rd7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

ori r16, \$F0 ; Установить старший ниббл r16

ori r17, 1 ; Установить бит 0 регистра r17

Слов: 1 (2 байта)

Циклов: 1

Команда OUT - записать данные из регистра в порт I/O

Описание:

Команда сохраняет данные регистра Rr в регистровом файле пространства I/O (порты, таймеры, регистры конфигурации и т.п.).

Операция(i) P <-- Rr

Синтаксис Операнды: Счетчик программ:  
(i) OUT P,Rr 0 < r <31, 0 < P <63 PC<-- + 1

16-разрядный код операции:

1011 1PPr rrrr PPPP

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

cli r16 ; Очистить r16

sei r17 ; Установить r17

out \$18, r16 ; Записать нули в Порт В

por ; Ожидать (пустая операция)

out \$18, r17 ; Записать единицы в Порт В

Слов: 1 (2 байта)

Циклов: 1

Команда POP - записать регистр из стека

Описание:

Команда загружает регистр Rd байтом содержимого стека.

Операция(i) Rd <-- STACK

Синтаксис Операнды: Счетчик программ:  
(i) POP Rd 0 < d <31 PC<-- + 1

SP<-- SP + 1

16-разрядный код операции:

1001 000d dddd 1111

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

```

- - - - -
Пример:
    call routine ; Вызвать подпрограмму
    ...
routine: push r14 ; Сохранить r14 в стеке
        push r13 ; Сохранить r13 в стеке
        ...
        pop r13 ; Восстановить r13
        pop r14 ; Восстановить r14
        get     ; Вернуться из подпрограммы

```

Слов: 1 (2 байта)

Циклов: 2

.....

Команда PUSH - поместить регистр в стек

Описание:

Команда помещает содержимое регистра Rd в стек.

Операция(i)	STACK <-- Rr		
	Синтаксис	Операнды:	Счетчик программ:
(i)	PUSH Rr	0 < d < 31	PC<-- + 1
	SP<-- SP - 1		

16-разрядный код операции:  
1001 001d dddd 1111

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

```

- - - - -
Пример:
    call routine ; Вызвать подпрограмму
    ...
routine: push r14 ; Сохранить r14 в стеке
        push r13 ; Сохранить r13 в стеке
        ...
        pop r13 ; Восстановить r13
        pop r14 ; Восстановить r14
        get     ; Вернуться из подпрограммы

```

Слов: 1 (2 байта)

Циклов: 2

.....

Команда RCALL - вызвать подпрограмму относительно

Описание:

Команда вызывает подпрограмму в пределах +2 Кслов (4 Кбайт). Адрес возврата (после выполнения команды RCALL) сохраняется в стеке (См. также команду CALL).

Операция(i)	PC <-- PC + k + 1	Приборы с 16-разрядным счетчиком команд, максимум 128 Кбайт памяти программ
(ii)	PC <-- PC + k + 1	Приборы с 22-разрядным счетчиком команд, максимум 8 Мбайт памяти программ

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	RCALL k	-2K < k < 2K	PC <-- PC + k + 1	STACK <-- PC + 1
			SP <-- SP-2 (2 байта, 16 бит)	
(ii)	RCALL k	-2K < k < 2K	PC <-- PC + k + 1	STACK <-- PC + 1
			SP <-- SP-3 (3 байта, 22 бита)	

16-разрядный код операции:  
1101 kkkk kkkk kkkk

Булевы выражения регистра статуса (SREG)  
 I T H S V N Z C  
 - - - - - - - -

Пример:  
 rcall routine ; Вызвать подпрограмму  
 ...  
 routine: push r14 ; Сохранить r14 в стеке  
 ...  
 pop r14 ; Восстановить r14  
 get ; Вернуться из подпрограммы

Слов: 1 (2 байта)

Циклов: 3

Команда RET - вернуться из подпрограммы

Описание:

Команда возвращает из подпрограммы. Адрес возврата загружается из стека.

Операция(i) PC(15-0) <-- STACK Приборы с 16-разрядным счетчиком команд, максимум 128 Кбайт памяти программ  
 (ii) PC(21-0) <-- STACK Приборы с 22-разрядным счетчиком команд, максимум 8 Мбайт памяти программ

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	RET None	См. операцию	SP <-- SP+2 (2 байта, 16 бит)	
(ii)	RET None	См. операцию	SP <-- SP+3 (3 байта, 22 бита)	

16-разрядный код операции:  
 1001 0101 0XX0 1000

Булевы выражения регистра статуса (SREG)  
 I T H S V N Z C  
 - - - - - - - -

Пример:  
 call routine ; Вызвать подпрограмму  
 ...  
 routine: push r14 ; Сохранить r14 в стеке  
 ...  
 pop r14 ; Восстановить r14  
 get ; Вернуться из подпрограммы

Слов: 1 (2 байта)

Циклов: 4

Команда RETI - вернуться из прерывания

Описание:

Команда возвращает из прерывания. Адрес возврата выгружается из стека и устанавливается флаг глобального прерывания.

Операция(i) PC(15-0) <-- STACK Приборы с 16-разрядным счетчиком команд, максимум 128 Кбайт памяти программ  
 (ii) PC(21-0) <-- STACK Приборы с 22-разрядным счетчиком команд, максимум 8 Мбайт памяти программ

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	RETI None	См. операцию	SP <-- SP+2 (2 байта, 16 бит)	
(ii)	RETI None	См. операцию	SP <-- SP+3 (3 байта, 22 бита)	

16-разрядный код операции:  
1001 0101 0XX1 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
1	-	-	-	-	-	-	-

I: 1  
Флаг установлен

Пример:

```
...  
extint: push r0 ; Сохранить r0 в стеке  
...  
pop r0 ; Восстановить r0  
reti ; Вернуться и разрешить прерывания
```

Слов: 1 (2 байта)

Циклов: 4

Команда RJMP - перейти относительно

Описание:

Команда выполняет относительный переход по адресу в пределах +2 Кслов (4 Кбайт) текущего состояния счетчика команд. В ассемблере вместо относительных операндов используются метки. Для AVR микроконтроллеров с памятью программ не превышающей 4 Кслов (8 Кбайт) данная команда может адресовать всю память программ.

Операция(i) PC  $\leftarrow$  PC + k + 1

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	RJMP k -2K < k < 2K	PC $\leftarrow$ PC + k + 1	Стек не меняется	

16-разрядный код операции:  
1100 kkkk kkkk kkkk

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
1	-	-	-	-	-	-	-

I: 1  
Флаг установлен

Пример:

```
cp r16, $42 ; Сравнить r16 с $42  
bne error ; Перейти если r16 <> $42  
rjmp ok ; Безусловный переход  
error: add r16, r17 ; Сложить r17 с r16  
inc r16 ; Увеличить на 1 r16  
ok: nop ; Назначение для rjmp (пустая операция)
```

Слов: 1 (2 байта)

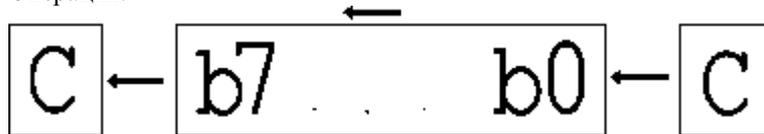
Циклов: 2

Команда ROL - сдвинуть влево через перенос

Описание:

Сдвиг всех битов Rd на одно место влево. Флаг переноса (C) регистра состояния (SREG) смещается на место бита 0 регистра Rd. Бит 7 смещается во флаг переноса (C).

Операция:



16-разрядный код операции:  
0001 11dd dddd dddd

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	Ы	Ы	Ы	Ы	Ы	Ы

H: Rd3

S: NEV, Для проверок со знаком

V: NEC (Для N и C после сдвига) Устанавливается если (N устанавливается и C очищается) или (N очищается а C устанавливается). В ином случае очищается (при наличии значений N и C после сдвига)

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: Rd7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

C: Rd7

Устанавливается если перед сдвигом был установлен MSB регистра Rd, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

rol r15 ; Сдвигать влево

brcs oneenc ; Перейти если установлен перенос

...

oneenc: nop ; Перейти по назначению (пустая операция)

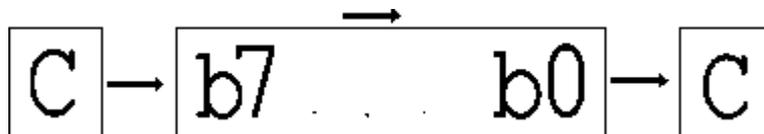
Слов: 1 (2 байта)

Циклов: 1

Команда ROR - сдвинуть вправо через перенос

Описание:

Сдвиг всех битов Rd на одно место вправо. Флаг переноса (C) регистра состояния (SREG) смещается на место бита 7 регистра Rd. Бит 0 смещается во флаг переноса (C).



Операция:

16-разрядный код операции:  
1001 010d dddd 0111

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	Ы	Ы	Ы	Ы

S: NEV, Для проверок со знаком

V: NEC (Для N и C после сдвига)

Устанавливается если (N устанавливается и C очищается) или (N очищается а C устанавливается). В ином случае очищается (при наличии значений N и C после сдвига)

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: Rd7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

C: Rd0

Устанавливается если перед сдвигом был установлен LSB регистра Rd, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

ror r15 ; Сдвигать вправо

brcc zeroenc ; Перейти если перенос очищен

....

zeroenc: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SBC - вычесть с переносом

Описание:

Вычитание содержимого регистра-источника и содержимого флага переноса (C) из регистра Rd, размещение результата в регистре назначения Rd.

Операция:(i) Rd <-- Rd - Rr - C

Синтаксис Операнды: Счетчик программ:  
(i) SBC Rd,Rr 0 < d < 31, 0 < r < 31 PC <- PC + 1

16-разрядный код операции:

0000 10rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H: Rd3\*Rr3+Rr3\*R3+R3\*Rd3

Устанавливается если есть заем из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V: Rd7\*Rr7\*R7+Rd7\*Rr7\*R7

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0\*Z

Предшествовавшее значение остается неизменным если результат равен нулю, в ином случае очищается

C: Rd7\*Rr7+Rr7\*R7+ R7\*Rd7

Устанавливается если абсолютное значение содержимого Rr плюс предшествовавший перенос больше, чем абсолютное значение Rd, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

; Вычесть r1 : r0 из r3 : r2

sub r2, r0 ; Вычесть младший байт

sbc r3, r1 ; Вычесть старший байт с переносом

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SBCI - вычесть непосредственное значение с переносом

Описание:

Вычитание константы и содержимого флага переноса (C) из содержимого регистра, размещение результата в регистре назначения Rd.

Операция:(i) Rd <-- Rd - K - C

Синтаксис Операнды: Счетчик программ:  
(i) SBCI Rd,K 0 < d < 31, 0 < K < 255 PC <- PC + 1

16-разрядный код операции:  
0100 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - Ы Ы Ы Ы Ы Ы

H:  $Rd3 * K3 + K3 * R3 + R3 * Rd3$

Устанавливается если есть заем из бита 3, в ином случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * K7 * R7 + Rd7 * K7 * R7$

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0 * Z$

Предшествовавшее значение остается неизменным если результат равен нулю, в ином случае очищается

C:  $Rd7 * K7 + K7 * R7 + R7 * Rd7$

Устанавливается если абсолютное значение константы плюс предшествовавший перенос больше, чем абсолютное значение Rd, в ином случае очищается

R: (Результат)соответствует Rd после выполнения команды

Пример:

; Вычесть \$4F23 из r17 : r16  
subi r16, r23 ; Вычесть младший байт  
sbci r17, \$4F ; Вычесть старший байт с переносом

Слов: 1 (2 байта)

Циклов: 1

Команда SBI - установить бит в регистр I/O

Описание:

Команда устанавливает заданный бит в регистр I/O. Команда работает с младшими 32 регистрами I/O (адреса с 0 по 31)

Операция(i) I/O(P,b) <-- 1

(i)	Синтаксис	Операнды:	Счетчик программ:
(i)	SBI P,b	$0 < P < 31, 0 < b < 7$	$PC <-- PC + k + 1$

16-разрядный код операции:

1001 1010 pppp pbbb

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

out \$1E, r0 ; Записать адрес EEPROM  
sbi \$1C, 0 ; Установить бит чтения в EECR  
in r1, \$1D ; Читать данные EEPROM

Слов: 1 (2 байта)

Циклов: 2

Команда SBIC - пропустить если бит в регистре I/O очищен

Описание:

Команда проверяет состояние бита в регистре I/O и, если этот бит очищен, пропускает следующую команду. Данная команда работает с младшими 32 регистрами I/O (адреса с 0 по 31).

Операция(i) If I/O(P,b) = 0 then PC <-- PC + 2 (or 3) else PC <-- PC + 1

Синтаксис            Операнды:            Счетчик программ:  
(i)            SBIS P,b            0 < P <31, 0 < b < 7            PC <-- PC + 1, если условия не соблюдены, нет пропуска  
PC <-- PC + 2, если следующая команда длиной в 1 слово  
PC <-- PC + 3, если следующие команды JMP или CALL

16-разрядный код операции:  
1001    1001    pppp    pbbb

Булевы выражения регистра статуса (SREG)  
I            T            H            S            V            N            Z            C  
-            -            -            -            -            -            -            -

Пример:  
e2wait: sbic \$1C, 1 ; Пропустить следующую команду если EEWЕ очищен  
          rjmp e2wait ; Запись EEPROM не завершена  
          nop            ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, нет пропуска  
2 если условия соблюдены, выполняется пропуск

.....  
Команда SBIS - пропустить если бит в регистре I/O установлен

Описание:

Команда проверяет состояние бита в регистре I/O и, если этот бит установлен, пропускает следующую команду. Данная команда работает с младшими 32 регистрами I/O (адреса с 0 по 31).

Операция(i) If I/O(P,b) = 1 then PC <-- PC + 2 (or 3) else PC <-- PC + 1

Синтаксис            Операнды:            Счетчик программ:  
(i)            SBIS P,b            0 < P <31, 0 < b < 7            PC <-- PC + 1, если условия не соблюдены, нет пропуска  
PC <-- PC + 2, если следующая команда длиной в 1 слово  
PC <-- PC + 3, пропускает команды JMP или CALL

16-разрядный код операции:  
1001    1011    pppp    pbbb

Булевы выражения регистра статуса (SREG)  
I            T            H            S            V            N            Z            C  
-            -            -            -            -            -            -            -

Пример:  
waitset: sbis\$10, 0 ; Пропустить следующую команду если установлен бит 0 в Порте D  
          rjmp waitset ; Бит не установлен  
          nop            ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, нет пропуска  
2 если условия соблюдены, выполняется пропуск

.....  
Команда SBIW - вычесть непосредственное значение из слова

Описание:

Вычитание непосредственного значения (0-63) из пары регистров и размещение результата в паре регистров. Команда работает с четырьмя верхними парами регистров, удобна для работы с регистрами указателями.

Операция:(i) Rdh:Rdl <-- Rdh:Rdl - K

(i) Синтаксис Операнды: Счетчик программ:  
SBIW Rdl,K dIO{24,26,28,30}, 0 < K < 63 PC <- PC + 1

16-разрядный код операции:  
1001 0111 KKdd KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	Ы	Ы	Ы	Ы

S: NEV, Для проверок со знаком

V: Rdh7\*R15

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R15

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R15\*R14\*R13\*R12\*R11\*R10\*R9\*R8\*R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$0000, в ином случае очищается

C: R15\*Rdh7

Устанавливается если абсолютное значение константы K больше абсолютного значения содержимого регистра Rd, в ином случае очищается

R: (Результат) соответствует Rdh:Rdl после выполнения команды (Rdh7-Rdh0 = R15-R8, Rdl7-Rdl0 = R7-R0)

Пример:

sbiw r24, 1 ; Вычесть 1 из r25:r24

sbiw r28, 63 ; Вычесть 63 из Y указателя (r29 : r28)

Слов: 1 (2 байта)

Циклов: 2

Команда SBR - установить биты в регистре

Описание:

Команда выполняет установку определенных битов в регистре Rd. Команда выполняет логическое ORI между содержимым регистра Rd и маской-константой K и размещает результат в регистре назначения Rd.

Операция:(i) Rd <-- Rd v K

(i) Синтаксис Операнды: Счетчик программ:  
SBR Rd,K 16 < d < 31, 0 < K < 255 PC <- PC + 1

16-разрядный код операции:  
0110 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	0	Ы	Ы	-

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

sbr r16, 3F0 ; Установить биты 0 и 1 в r16

sbr r17, \$F0 ; Установить старшие 4 бита в r17

Слов: 1 (2 байта)

Циклов: 1

Команда SBRC - пропустить если бит в регистре очищен

Описание:

Команда проверяет состояние бита в регистре и, если этот бит очищен, пропускает следующую команду.

Операция(i) If Rr (b) = 0 then PC <-- PC + 2 (or 3) else PC <-- PC + 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SBRC Rr,b	$0 < r < 31, 0 < b < 7$	PC <-- PC + 1, если условия не соблюдены, нет пропуска PC <-- PC + 2, если следующая команда длиной в 1 слово PC <-- PC + 3, если следующие команды JMP или CALL

16-разрядный код операции:

1111 110r rrr Xbbb

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

sub r0, r1 ; Вычесть r1 из r0  
sbrс r0, 7 ; Пропустить если бит 7 в r0 очищен  
sub r0, r1 ; Выполняется только если бит 7 в r0 не очищен  
por ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, нет пропуска

2 если условия соблюдены, выполняется пропуск

Команда SBRS - пропустить если бит в регистре установлен

Описание:

Команда проверяет состояние бита в регистре и, если этот бит установлен, пропускает следующую команду.

Операция(i) If Rr(b) = 1 then PC <-- PC + 2 (or 3) else PC <-- PC + 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SBRS Rr,b	$0 < r < 31, 0 < b < 7$	PC <-- PC + 1, если условия не соблюдены, нет пропуска PC <-- PC + 2, если следующая команда длиной в 1 слово PC <-- PC + 3, пропускает команды JMP или CALL

16-разрядный код операции:

1111 111r rrr Xbbb

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

sub r0, r1 ; Вычесть r1 из r0  
sbrs r0, 7 ; Пропустить если бит 7 в r0 установлен  
neg r0 ; Выполняется только если бит 7 в r0 не установлен  
por ; Продолжать (пустая операция)

Слов: 1 (2 байта)

Циклов: 1 если условия не соблюдены, нет пропуска  
2 если условия соблюдены, выполняется пропуск

Команда SEC - установить флаг переноса

Описание:

Команда устанавливает флаг переноса (C) в регистре статуса (SREG)

Операция(i) C <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SEC None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0000 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	1

C: 1

Флаг переноса установлен

Пример:

sec ; Установить флаг переноса  
adc r0, r1 ; r0 = r0 + r1 + 1

Слов: 1 (2 байта)

Циклов: 1

Команда SEH - установить флаг полупереноса

Описание:

Команда устанавливает флаг полупереноса (H) в регистре статуса (SREG).

Операция(i) H <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SEH None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0101 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	1	-	-	-	-	-

H: 1

Флаг полупереноса установлен

Пример:

seh ; Установить флаг полупереноса

Слов: 1 (2 байта)

Циклов: 1

Команда SEI - установить флаг глобального прерывания

Описание:

Команда устанавливает флаг глобального прерывания (I) в регистре статуса (SREG).

Операция(i) I <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SEI None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0111 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

1	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

I: 1

Флаг глобального прерывания установлен

Пример:

cli ; Запретить прерывания  
in r13, \$16 ; Читать Порт В  
sei ; Разрешить прерывания

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SEN - установить флаг отрицательного значения

Описание:

Команда устанавливает флаг отрицательного значения (N) в регистре статуса (SREG).

Операция(i) N <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SEN None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0010 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

-	-	-	-	-	1	-	-
---	---	---	---	---	---	---	---

N: 1

Флаг переноса установлен

Пример:

add r2, r19 ; Сложить r19 с r2  
sen ; Установить флаг отрицательного значения

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SER - установить все биты регистра

Описание:

Значение \$FF заносится непосредственно в регистр назначения Rd.

Операция(i) Rd <-- \$FF

	Синтаксис	Операнды:	Счетчик программ:
(i)	SER Rd 16 < d < 31	PC <-- PC + 1	

16-разрядный код операции:

1110 1111 dddd 1111

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - - - - - -

Пример:

clr r16 ; Очистить r16  
set r17 ; Установить r17  
out #18, r16 ; Записать нули в Порт В  
nop ; Задержка (пустая операция)  
out #18, r17 ; Записать единицы в Порт В

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SES - установить флаг знака

Описание:

Команда устанавливает флаг учета знака (S) в регистре статуса (SREG).

Операция(i) S <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SES None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0100 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- - - 1 - - - -

S: 1

Флаг учета знака установлен

Пример:

add r2, r19 ; Сложить r19 с r2  
ses ; Установить флаг отрицательного значения

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SET - установить флаг T

Описание:

Команда устанавливает флаг пересылки (T) в регистре статуса (SREG).

Операция(i) T <-- 1

	Синтаксис	Операнды:	Счетчик программ:
(i)	SET None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0110 1000

Булевы выражения регистра статуса (SREG)

I T H S V N Z C

- 1 - - - - - -

T: 1  
Флаг пересылки установлен

Пример:  
set ; Установить T флаг

Слов: 1 (2 байта)

Циклов: 1

Команда SEV - установить флаг переполнения

Описание:

Команда устанавливает флаг переполнения (V) в регистре статуса (SREG).

Операция(i) V <-- 1

(i)	Синтаксис	Операнды:	Счетчик программ:
(i)	SEV None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0011 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	1	-	-	-

V: 1  
Флаг переполнения установлен

Пример:  
add r2, r19 ; Сложить r19 с r2  
sev ; Установить флаг переполнения

Слов: 1 (2 байта)

Циклов: 1

Команда SEZ - установить флаг нулевого значения

Описание:

Команда устанавливает флаг нулевого значения (Z) в регистре статуса (SREG).

Операция(i) Z <-- 1

(i)	Синтаксис	Операнды:	Счетчик программ:
(i)	SEZ None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0100 0001 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	1	-

Z: 1  
Флаг нулевого значения установлен

Пример:  
add r2, r19 ; Сложить r19 с r2  
sez ; Установить флаг нулевого значения

Слов: 1 (2 байта)

Циклов: 1

Команда SLEEP - установить режим SLEEP

Описание:

Команда устанавливает схему в SLEEP режим, определяемый регистром управления ЦПУ. Когда прерывание выводит ЦПУ из SLEEP режима команда, следующая за командой SLEEP, будет выполнена прежде, чем обработает обработчик прерывания.

Операция

	Синтаксис	Операнды:	Счетчик программ:
(i)	SLEEP None	PC <-- PC + 1	

16-разрядный код операции:  
1001 0101 100X 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```
mov r0, r11 ; Копировать r11 в r0
sleep      ; Перевести MCU в режим sleep
```

Слов: 1 (2 байта)

Циклов: 1

Команда ST - записать косвенно из регистра в СОЗУ с использованием индекса X

Описание:

Записывается косвенно один байт из регистра в СОЗУ. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем X в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPX в I/O области. Регистр-указатель X может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Эта особенность очень удобна при использовании регистра-указателя X в качестве указателя стека.

Использование X-указателя:	Операция:	Комментарий:
(i) (X) <-- Rr	X: Неизменен	
(ii) (X) <-- Rr	X <-- X + 1	X: Инкрементирован впоследствии
(iii) X <-- X - 1	( X) <-- Rr	X: Предварительно декрементирован

	Синтаксис	Операнды:	Счетчик программ:
(i)	ST X,Rr 0 < d < 31	PC <-- PC + 1	
(ii)	ST X+,Rr	0 < d < 31	PC <-- PC + 1
(iii)	ST -X,Rr	0 < d < 31	PC <-- PC + 1

16-разрядный код операции:  
(i) 1001 001r rrr 1100  
(ii) 1001 001r rrr 1101  
(iii) 1001 001r rrr 1110

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```
clr r27 ; Очистить старший байт X
ldi r26, $20 ; Установить $20 в младший байт X
st X+,r0 ; Сохранить в r0 содержимое SRAM по адресу $20 (X постинкрементируется)
st X, r1 ; Сохранить в r1 содержимое SRAM по адресу $21
ldi r26, $23 ; Установить $23 в младший байт X
st r2, X ; Сохранить в r2 содержимое SRAM по адресу $23
```

st r3, -X ; Сохранить в r3 содержимое SRAM по адресу \$22 (X преддекрементируется)

Слов: 1 (2 байта)

Циклов: 2

оманда ST (STD) - записать косвенно из регистра в СОЗУ с использованием индекса Y

Описание:

Записывается косвенно, со смещением или без смещения, один байт из регистра в СОЗУ. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем Y в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPY в I/O области. Регистр-указатель Y может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Эта особенность очень удобна при использовании регистра-указателя Y в качестве указателя стека.

Использование Y-указателя:	Операция:	Комментарий:
(i) (Y) <-- Rr	Y: Неизменен	
(ii) (Y) <-- Rr	Y <-- Y + 1	Y: Инкрементирован впоследствии
(iii) Y <-- Y - 1	(Y) <-- Rr	Y: Предварительно декрементирован
(iv) (Y + q) <-- Rr	Y: Неизменен, q: смещение	

	Синтаксис	Операнды:	Счетчик программ:
(i)	ST Y,Rr 0 < d < 31	PC <-- PC + 1	
(ii)	ST Y+,Rr	0 < d < 31	PC <-- PC + 1
(iii)	ST -Y,Rr	0 < d < 31	PC <-- PC + 1
(iv)	STD Y+q,Rr 0 < q < 63	0 < d < 31, PC <-- PC + 1	

16-разрядный код операции:

(i)	1000	001r	rrr	1000
(ii)	1001	001r	rrr	1001
(iii)	1001	001r	rrr	1010
(iiii)	10q0	qq1r	rrr	1qqq

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```
clr r29 ;Очистить старший байт Y
ldi r28, $20 ;Установить $20 в младший байт Y
st Y+, r0 ;Сохранить в r0 содерж. SRAM по адресу $20 (Y постинкрементируется)
st Y, r1 ;Сохранить в r1 содержимое SRAM по адресу $21
ldi r28, $23 ;Установить $23 в младший байт Y
st Y, r2 ;Сохранить в r2 содержимое SRAM по адресу $23
st -Y, r3 ;Сохранить в r3 содерж. SRAM по адресу $22 (Y преддекрементируется)
std Y+2, r4 ;Сохранить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта)

Циклов: 2

оманда ST (STD) - записать косвенно из регистра в СОЗУ с использованием индекса Y

Описание:

Записывается косвенно, со смещением или без смещения, один байт из регистра в СОЗУ. Положение байта в СОЗУ указывается 16-разрядным регистром-указателем Y в регистровом файле. Обращение к памяти ограничено текущей страницей объемом 64 Кбайта. Для обращения к другой странице СОЗУ необходимо изменить регистр RAMPY в I/O области. Регистр-указатель Y может остаться неизменным после выполнения команды, но может быть инкрементирован или декрементирован. Эта особенность очень удобна при использовании регистра-указателя Y в качестве указателя стека.

Использование Y-указателя:	Операция:	Комментарий:
----------------------------	-----------	--------------

- (i) (Y) <-- Rr Y: Неизменен
- (ii) (Y) <-- Rr Y <-- Y + 1 Y: Инкрементирован впоследствии
- (iii) Y <-- Y - 1 (Y) <-- Rr Y: Предварительно декрементирован
- (iv) (Y + q) <-- Rr Y: Неизменен, q: смещение

	Синтаксис	Операнды:	Счетчик программ:
(i)	ST Y,Rr 0 < d < 31	PC <-- PC + 1	
(ii)	ST Y+,Rr	0 < d < 31	PC <-- PC + 1
(iii)	ST -Y,Rr	0 < d < 31	PC <-- PC + 1
(iv)	STD Y+q,Rr	0 < d < 31, 0 < q < 63	PC <-- PC + 1

16-разрядный код операции:

- (i) 1000 001r rrrr 1000
- (ii) 1001 001r rrrr 1001
- (iii) 1001 001r rrrr 1010
- (iiii) 10q0 qq1r rrrr 1qqq

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```

clr r29 ;Очистить старший байт Y
ldi r28, $20 ;Установить $20 в младший байт Y
st Y+, r0 ;Сохранить в r0 содерж. SRAM по адресу $20 (Y постинкрементируется)
st Y, r1 ;Сохранить в r1 содержимое SRAM по адресу $21
ldi r28, $23 ;Установить $23 в младший байт Y
st Y, r2 ;Сохранить в r2 содержимое SRAM по адресу $23
st -Y, r3 ;Сохранить в r3 содерж. SRAM по адресу $22 (Y преддекрементируется)
std Y+2, r4 ;Сохранить в r4 содержимое SRAM по адресу $24

```

Слов: 1 (2 байта)

Циклов: 2



Команда STS - загрузить непосредственно в СОЗУ

Описание:

Выполняется запись одного байта из регистра в СОЗУ. Можно использовать 16- разрядный адрес. Обращение к памяти ограничено текущей страницей СОЗУ объемом 64 Кбайта. Команда STS использует для обращения к памяти выше 64 Кбайт регистр RAMPZ.

Операция:(i) (k) <-- Rr

	Синтаксис	Операнды:	Счетчик программ:
(i)	STS k,Rr	0 < r < 31, 0 < k < 65535	PC <-- PC + 2

32-разрядный код операции:

- 1001 001d dddd 0000
- kkkk kkkk kkkk kkkk

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```

lds r2, $FF00 ; Загрузить в r2 содержимое SRAM по адресу $FF00
add r2, r1 ; Сложить r1 с r2
sts $FF00, r2 ; Записать обратно

```

Слов: 2 (4 байта)

Циклов: 3

Команда SUB - вычесть без переноса

Описание:

Вычитание содержимого регистра-источника Rr из содержимого регистра Rd, размещение результата в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd - Rr$

	Синтаксис	Операнды:	Счетчик программ:
(i)	SUB Rd,Rr	$16 < d < 31, 0 < r < 31$	PC $\leftarrow$ PC + 1

16-разрядный код операции:  
0001 10rd dddd rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

-	-	Ы	Ы	Ы	Ы	Ы	Ы
---	---	---	---	---	---	---	---

H:  $Rd3 * Rr3 + Rr3 * R3 + R3 * Rd3$

Устанавливается если есть заем из бита 3, в противном случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * Rr7 * R7 + Rd7 * Rr7 * R7$

Устанавливается если в результате операции образуется переполнение дополнения до двух, в противном случае очищается

N: R7

Устанавливается если в результате установлен MSB, в противном случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в противном случае очищается

C:  $Rd7 * Rr7 + Rr7 * R7 + * R7 * Rd7$

Устанавливается если абсолютное значение содержимого Rr больше, чем абсолютное значение Rd, в противном случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

sub r13, r12 ; Вычесть r12 из r13

bne noteq ; Перейти если  $r12 \neq r13$

noteq: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

Команда SUBI - вычесть непосредственное значение

Описание:

Вычитание константы из содержимого регистра, размещение результата в регистре назначения Rd.

Операция:(i)  $Rd \leftarrow Rd - K$

	Синтаксис	Операнды:	Счетчик программ:
(i)	SUBI Rd,K	$16 < d < 31, 0 < K < 255$	PC $\leftarrow$ PC + 1

16-разрядный код операции:  
0101 KKKK dddd KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
---	---	---	---	---	---	---	---

-	-	Ы	Ы	Ы	Ы	Ы	Ы
---	---	---	---	---	---	---	---

H:  $Rd3 * K3 + K3 * R3 + R3 * Rd3$

Устанавливается если есть заем из бита 3, в противном случае очищается

S: NEV, Для проверок со знаком

V:  $Rd7 * K7 * R7 + Rd7 * K7 * R7$

Устанавливается если в результате операции образуется переполнение дополнения до двух, в ином случае очищается

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z:  $R7 * R6 * R5 * R4 * R3 * R2 * R1 * R0$

Устанавливается если результат \$00, в ином случае очищается

C:  $Rd7 * K7 + K7 * R7 + R7 * Rd7$

Устанавливается если абсолютное значение константы больше, чем абсолютное значение Rd, в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

subi r22, \$11 ; Вычесть \$11 из r22

brne noteq ; Перейти если r22 <> \$11

...

noteq: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

.....

Команда SWAP - поменять нибблы местами

Описание:

Команда меняет местами старший и младший нибблы (полубайты) регистра.

Операция:(i)  $R(7-4) \leftarrow R(3-0), R(3-0) \leftarrow R(7-4)$

	Синтаксис	Операнды:	Счетчик программ:
(i)	SWAP Rd	$0 < d < 31$	$PC \leftarrow PC + k + 1$

16-разрядный код операции:

1001 010d dddd 0010

Булевы выражения регистра статуса (SREG)

I T N S V N Z C

- - - - - - - - -

R: (Результат) соответствует Rd после выполнения команды

Пример:

inc r1 ; Увеличить на 1 r1

swap r1 ; Поменять местами нибблы r1

inc r1 ; Увеличить на 1 старший ниббл r1

swap r1 ; Снова поменять местами нибблы r1

Слов: 1 (2 байта)

Циклов: 1

.....

Команда TST - проверить на ноль или минус

Описание:

Регистр проверяется на нулевое или отрицательное состояние. Выполняется логическое AND содержимого регистра с самим собой. Содержимое регистра остается неизменным.

Операция:(i)  $Rd \leftarrow Rd * Rd$

	Синтаксис	Операнды:	Счетчик программ:
(i)	TST Rd	$0 < d < 31$	$PC \leftarrow PC + 1$

16-разрядный код операции:

0010 00dd dddd dddd

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	Ы	0	Ы	Ы	-

S: NEV, Для проверок со знаком

V: 0

Очищен

N: R7

Устанавливается если в результате установлен MSB, в ином случае очищается

Z: R7\*R6\*R5\*R4\*R3\*R2\*R1\*R0

Устанавливается если результат \$00, в ином случае очищается

R: (Результат) соответствует Rd

Пример:

tst r0 ; Проверить r0

breq zero ; Перейти если r0 = 0

...

zero: nop ; Перейти по назначению (пустая операция)

Слов: 1 (2 байта)

Циклов: 1

.....

Команда WDR - сбросить сторожевой таймер

Описание:

Команда сбрасывает сторожевой таймер (Watchdog Timer). Команда может быть выполнена внутри заданного прескалером сторожевого таймера промежутка времени (см. аппаратные характеристики сторожевого таймера).

Операция:(i) Перезапускается WD (сторожевой таймер)

(i)	Синтаксис	Операнды:	Счетчик программ:
(i)	WDR None	PC <-- PC + 1	

16-разрядный код операции:

1001 0101 101X 1000

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

wdr ; Сбросить сторожевой таймер

Слов: 1 (2 байта)

Циклов: 1

