Наши координаты:
Москва, 1-й Щемиловский пер, 16, 2 подъезд, 2 этаж
Тел.: (095) 972 3416, 973 1855, 973 1923, факс 973 1864
E-mail: cec-mc@mega.ru
Наш сервер http://www.cec-mc.ru

ПОЛНЫЙ КОМПЛЕКС РАЗРАБОТКИ ЭЛЕКТРОННЫХ ИЗДЕЛИЙ ЛЮБОЙ СЛОЖНОСТИ, ВКЛЮЧАЮЩИЙ В СЕБЯ ИЗГОТОВЛЕНИЕ ОПЫТНОГО ОБРАЗЦА

ПРОИЗВОДСТВО ИЗДЕЛИЙ ПО ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ ЗАКАЗЧИКА

ПРОЕКТИРОВАНИЕ ПЕЧАТНЫХ ПЛАТ ЛЮБОЙ СЛОЖНОСТИ КАК С ПРИНЦИПИАЛЬНОЙ СХЕМЫ, ТАК И ИСПОЛНЕНИЕ ПРОЕКТА С ТЕХНИЧЕСКОГО ЗАДАНИЯ

ИЗГОТОВЛЕНИЕ ПЕЧАТНЫХ ПЛАТ ДО 5 КЛАССА, ДО 24 СЛОЕВ, С ПОЛНЫМ ЭЛЕКТРОКОНТРОЛЕМ

МОНТАЖ ПЕЧАТНЫХ ПЛАТ ЛЮБОЙ СЛОЖНОСТИ ВКЛЮЧАЯ ПОВЕРХНОСТНЫЙ, В ТОМ ЧИСЛЕ КОРПУСОВ BGA

ИЗГОТОВЛЕНИЕ ШТАМПОВАННЫХ И ФРЕЗЕРОВАННЫХ КОРПУСОВ ДЛЯ ПРОМЫШЛЕННОЙ АППАРАТУРЫ

КОМПЛЕКСНЫЕ ПОСТАВКИ ЭЛЕКТРОННЫХ КОМПОНЕНТОВ ДЛЯ ВЫПУСКАЕМЫХ ИЗДЕЛИЙ

СИСТЕМА КОМАНД

8-РАЗРЯДНЫХ RISC МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА AVR

Система команд 8-разрядных RISC микроконтроллеров семейства AVR.

Принятые обозначения

Регистр статуса (SREG):
SREG: Регистр статуса
C: Флаг переноса
Z: Флаг нулевого значения
N: Флаг отрицательного значения
V: Флаг-указатель переполнения дополнения до двух
S: N ⊕ V, Для проверок со знаком
H: Флаг полупереноса
T: Флаг пересылки, используемый командами BLD и BST
I: Флаг разрешения/запрещения глобального прерывания

Регистры и операнды:
Rd: Регистр назначения (и источник) в регистровом файле
Rr: Регистр источник в регистровом файле
R: Результат выполнения команды
K: Литерал или байт данных (8 бит)
k: Данные адреса константы для счетчика программ
b: Бит в регистровом файле или I/O регистр (3 бита)
s: Бит в регистре статуса (3 бита)

X, Y, Z: Регистр косвенной адресации (X=R27:R26, Y=R29:R28, Z=R31:R30)
P: Адрес I/O порта
q: Смещение при прямой адресации (6 бит)

I/O регистры
RAMPX, RAMPY, RAMPZ: Регистры связанные с X, Y и Z регистрами, обеспечивающие косвенную адресацию всей области СОЗУ микроконтроллера с объемом СОЗУ более 64 Кбайт.

Стек:
STACK: Стек для адреса возврата и опущенных в стек регистров
SP: Указатель стека

Флаги:
<=> Флаг, на который воздействует команда
0: Очищенный командой Флаг
1: Установленный командой флаг
-: Флаг, на который не воздействует команда

Сводная таблица условных переходов

Тестирование условия	Булево выражение	Мнемоника	Комплементарное условие	Булево выражение	Мнемоника	Комментарий
Rd > Rr	Z • (N ⊕ V)=0	BRLT*	Rd ≤ Rr	Z+(N ⊕ V)=1	BRGE*	Со знаком
Rd ≥ Rr	(N ⊕ V)=0	BRGE	Rd < Rr	(N ⊕ V)=1	BRLT	Со знаком
Rd = Rr	Z=1	BREQ	Rd ≠ Rr	Z=0	BRNE	Со знаком
Rd ≤ Rr	Z+(N ⊕ V)=1	BRGE*	Rd > Rr	Z • N ⊕ V)=0	BRLT*	Со знаком
Rd < Rr	(N ⊕ V)=1	BRLT	Rd ≥ Rr	(N ⊕ V)=0	BRGE	Со знаком
Rd ≥ Rr	C+Z=0	BRLO*	Rd ≤ Rr	C+Z=1	BRSH*	Без знака
Rd > Rr	C=0	BRSH/BRCC	Rd < Rr	C=1	BRLO/BRCS	Без знака
Rd = Rr	Z=1	BREQ	Rd ≠ Rr	Z=0	BRNE	Без знака
Rd ≤ Rr	C+Z=1	BRSH*	Rd > Rr	C+Z=0	BRLO*	Без знака
Rd < Rr	C=1	BRLO/BRCS	Rd ≥ Rr	C=0	BRSH/BRCC	Без знака
Перенос	C=1	BRCS	Нет переноса	C=0	BRCC	Простой
Отрицательное значение	N=1	BRMI	Положительное значение	N=0	BRPL	Простой
Переполнение	V=1	BRVS	Нет переполнения	V=0	BRVC	Простой
Нулевое значение	Z=1	BREQ	Не нулевое значение	Z=0	BRNE	Простой

* Смена Rd и Rr при операции перед тестированием, т.е. CP Rd,Rr → CP Rr,Rd

Арифметические и логические команды

Мнем-оника	Опер-анды	Описание	Операция	Флаги	К-во цик-лов
ADD	Rd,Rr	Add without Carry - Сложить без переноса	Rd←Rd + Rr	Z,C,N,V,H	1
ADC	Rd,Rr	Add with Carry - Сложить с переносом	Rd←Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rd,K	Add Immediate to Word- Сложить непосредственное значение со словом	Rdh:Rdl←Rdh:Rdl + K	Z,C,N,V	2
SUB	Rd,Rr	Subtract without Carry - Вычесть без переноса	Rd←Rd - Rr	Z,C,N,V,H	1
SUBI	Rd,K	Subtract Immediate - Вычесть непосредственное значение	Rd←Rd - K	Z,C,N,V,H	1
SBC	Rd,Rr	Subtract with Carry - Вычесть с переносом	Rd←Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd,K	Subtract Immediate with Carry - Вычесть непосредственное значение с переносом	Rd←Rd - K - C	Z,C,N,V,H	1
SBIW	Rd,K	Subtract Immediate from Word- Вычесть непосредственное значение из слова	Rdh:Rdl← Rdh:Rdl - K	Z,C,N,V	2
AND	Rd,Rr	Logical AND- Выполнить логическое AND	Rd← Rd • Rr	Z,N,V	1
ANDI	Rd,K	Logical AND with Immediate- Выполнить логическое AND с непосредственным значением	Rd← Rd • K	Z,N,V	1
OR	Rd,Rr	Logical OR - Выполнить логическое OR	Rd←Rd v Rr	Z,N,V	1
ORI	Rd,K	Logical OR with Immediate - Выполнить логическое OR с непосредственным значением	Rd←Rd v K	Z,N,V	1
EOR	Rd,Rr	Exclusive OR - Выполнить исключающее OR	Rd← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement - Выполнить дополнение до единицы	Rd←$FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement - Выполнить дополнение до двух	Rd←$00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bits in Register- Установить биты в регистре	Rd←Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bits in Register - Очистить биты в регистре	Rd←Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment - Инкрементировать	Rd← Rd + 1	Z,N,V	1
DEC	Rd	Decrement - Декрементировать	Rd← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus - Проверить на ноль или минус	Rd← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register- Очистить регистр	Rd← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set all bits in Register- Установить все биты регистра	Rd← $FF	Нет	1
CP	Rd,Rr	Compare- Сравнить	Rd - Rr	Z,C,N,V,H	1
CPC	Rd,Rr	Compare with Carry - Сравнить с учетом переноса	Rd - Rr - C	Z,C,N,V,H	1
CPI	Rd,K	Compare with Immediate- Сравнить с константой	Rd - K	Z,C,N,V,H	1

Команды переходов

Мнемоника	Операнды	Описание	Операция	Флаги	К-во циклов
RJMP	k	Relative Jump - Перейти относительно	PC←PC + k + 1	Нет	2
IJMP		Indirect Jump- Перейти косвенно	PC←Z	Нет	2
JMP	k	Jump- Перейти	PC←k	Нет	3
RCALL	k	Relative Call to Subroutine - Вызвать подпрограмму относительно	PC←PC + k + 1	Нет	3
ICALL		Indirect Call to Subroutine - Вызвать подпрограмму косвенно	PC←Z	Нет	3
CALL	k	Long Call to a Subroutine - Выполнить длинный вызов подпрограммы	PC←k	Нет	4
RET		Return from Subroutine - Вернуться из подпрограммы	PC← STACK	Нет	4
RETI		Return from Interrupt- Вернуться из прерывания	PC← STACK	I	4
CPSE	Rd,Rr	Compare Skip if Equal- Сравнить и пропустить если равно	If Rd = Rr then PC←PC + 2 (or 3),	Нет	1/2/3
SBRC	Rr,b	Skip if Bit in Register is Cleared- Пропустить если бит в регистре очищен	If Rr (b) = 0 then PC←PC + 2 (or 3)	Нет	1/2/3
SBRS	Rr,b	Skip if Bit in Register is Set- Пропустить если бит в регистре установлен	If Rr(b) = 1 then PC←PC + 2 (or 3)	Нет	1/2/3
SBIC	P,b	Skip if Bit I/O Register is Cleared - Пропустить если бит в регистре I/O очищен	If I/O(P,b) = 0 then PC← PC + 2 (or 3)	Нет	1/2/3
SBIS	P,b	Skip if Bit I/O Register is Set- Пропустить если бит в регистре I/O установлен	If I/O(P,b) = 1 then PC← PC + 2 (or 3)	Нет	1/2/3
BRBS	s,k	Branch if Bit in SREG is Set- Перейти если бит в регистре статуса установлен	If SREG(s) = 1 then PC←PC + k +1	Нет	1/2
BRBC	s,k	Branch if Bit in SREG is Cleared - Перейти если бит в регистре статуса очищен	If SREG(s) = 0 then PC←PC + k +1	Нет	1/2
BREQ	k	Branch if Equal- Перейти если равно	If Rd = Rr (Z = 1) then PC←PC + k +1	Нет	1/2
BRNE	k	Branch if Not Equal- Перейти если не равно	If Rd ≠ Rr (Z = 0) then PC←PC + k +1	Нет	1/2
BRCS	k	Branch if Carry Set- Перейти если флаг переноса установлен	If C= 1 then PC←PC + k +1	Нет	1/2
BRCC	k	Branch if Carry Cleared- Перейти если флаг переноса очищен	If C= 0 then PC←PC + k +1	Нет	1/2
BRSH	k	Branch if Same or Higher (Unsigned) - Перейти если равно или больше (без знака)	If Rd ≥ Rr (C = 0) then PC←PC + k +1	Нет	1/2
BRLO	k	Branch if Lower (Unsigned) - Перейти если меньше (без знака)	If Rd < Rr (C = 1) then PC←PC + k +1	Нет	1/2
BRMI	k	Branch if Minus - Перейти если минус	If N = 1 then PC←PC + k +1	Нет	1/2
BRPL	k	Branch if Plus- Перейти если плюс	If N = 0 then PC←PC + k +1	Нет	1/2
BRGE	k	Branch if Greater or Equal (Signed)- Перейти если больше или равно (с учетом знака)	f Rd ≥ Rr (N⊕V = 0) then PC←PC + k +1	Нет	1/2

BRLT	k	Branch if Less Then (Signed) - Перейти если меньше чем (со знаком)	If Rd < Rr (N ⊕ V= 1) then PC<—PC + k +1	Нет	1/2
BRHS	k	Branch if Half Carry Flag is Set- Перейти если флаг полупереноса установлен	If H = 1 then PC< PC + k +1	Нет	1/2
BRHC	k	Branch if Half Carry Flag is Cleared - Перейти если флаг полупереноса очищен	If H = 0 then PC←PC + k +1	Нет	1/2
BRTS	k	Branch if T Flag is Set- Перейти если флаг T установлен	If T = 1 then PC←PC + k +1	Нет	1/2
BRTC	k	Branch if T Flag is Cleared- Перейти если флаг T очищен	If T = 0 then PC←PC + k +1	Нет	1/2
BRVS	k	Branch if Overflow Set - Перейти если переполнение установлено	If V = 1 then PC←PC + k +1	Нет	1/2
BRVC	k	Branch if Overflow Cleared- Перейти если переполнение очищено	If V = 0 then PC←PC + k +1	Нет	1/2
BRIE	k	Branch if Global Interrupt is Enabled - Перейти если глобальное прерывание разрешено	If I = 1 then PC←PC + k +1	Нет	1/2
BRID	k	Branch if Global Interrupt is Disabled - Перейти если глобальное прерывание запрещено	If I = 0 then PC< PC + k +1	Нет	1/2

Команды пересылки данных

Мнемоника	**Операнды**	**Описание**	**Операция**	**Флаги**	**К-во циклов**
ELPM		Extended Load Program Memory - Расширенная загрузка памяти программ	R_0←(Z+RAMPZ)	Нет	3
MOV	Rd,Rr	Copy Register- Копировать регистр	Rd← Rr	Нет	1
LDI	Rd,K	Load Immediate - Загрузить непосредственное значение	Rd←K	Нет	1
LDS	Rd,k	Load Direct from RAM - Загрузить непосредственно из СОЗУ	Rd←(k)	Нет	3
LD	Rd,X	LD - Load Indirect - Загрузить косвенно	Rd←(X)	Нет	2
LD	Rd,X+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	Rd←(X), X←X + 1	Нет	2
LD	Rd,-X	Load Indirect and Pre-Decrement - Загрузить косвенно декрементировав предварительно	X←X - 1, Rd←(X)	Нет	2
LD	Rd,Y	Load Indirect - Загрузить косвенно	Rd←(Y)	Нет	2
LD	Rd,Y+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	Rd←(Y), Y←Y + 1	Нет	2
LD	Rd,-Y	Load Indirect and Pre-Decrement -Загрузить косвенно декрементировав предварительно	Y←Y - 1, Rd←(Y)	Нет	2
LDD	Rd,Y+q	Load Indirect with Displacement - Загрузить косвенно со смещением	Rd←(Y + q)	Нет	2
LD	Rd,Z	Load Indirect - Загрузить косвенно	Rd←(Z)	Нет	2
LD	Rd,Z+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	Rd←(Z), Z←Z + 1	Нет	2
LD	Rd,-Z	Load Indirect and Pre-Decrement - Загрузить косвенно декрементировав предварительно	Z←Z - 1, Rd←(Z)	Нет	2
LDD	Rd,Z+q	Load Indirect with Displacement - Загрузить косвенно со смещением	Rd←(Z + q)	Нет	2

STS	k,Rr	Store Direct to RAM - Загрузить непосредственно в СОЗУ	(k)←Rr	Нет	3
ST	X,Rr	Store Indirect - Записать косвенно	(X)←Rr	Нет	2
ST	X+,Rr	Store Indirect and Post-Increment - Записать косвенно инкрементировав впоследствии	(X)←Rr, X←X + 1	Нет	2
ST	-X,Rr	Store Indirect and Pre-Decrement - Записать косвенно декрементировав предварительно	X←X - 1, (X)←Rr	Нет	2
ST	Y,Rr	Store Indirect - Записать косвенно	(Y)←Rr	Нет	2
ST	Y+,Rr	Store Indirect and Post-Increment - Записать косвенно инкрементировав впоследствии	(Y) ←Rr, Y←Y + 1	Нет	2
ST	-Y,Rr	Store Indirect and Pre-Decrement - Записать косвенно декрементировав предварительно	Y←Y - 1, (Y)←Rr	Нет	2
STD	Y+q,Rr	Store Indirect with Displacement - Записать косвенно со смещением	(Y + q)← Rr	Нет	2
ST	Z,Rr	Store Indirect - Записать косвенно	(Z)←Rr	Нет	2
ST	Z+,Rr	Store Indirect and Post-Increment - Записать косвенно инкрементировав впоследствии	(Z)←Rr, Z←Z + 1	Нет	2
ST	-Z,Rr	Store Indirect and Pre-Decrement - Записать косвенно декрементировав предварительно	Z←Z - 1, (Z)← Rr	Нет	2
STD	Z+q, Rr	Store Indirect with Displacement - Записать косвенно со смещением	(Z + q)← Rr	Нет	2
LPM		Load Program Memory - Загрузить байт памяти программ	R0← (Z)	Нет	3
IN	Rd,P	Load an I/O Port to Register - Загрузить данные из порта I/O в регистр	Rd← P	Нет	1
OUT	P,Rr	Store Register to I/O port - Записать данные из регистра в порт I/O	P←Rr	Нет	1
PUSH	Rr	Push Register on Stack - Поместить регистр в стек	STACK←Rr	Нет	2
POP	Rd	Pop Register from Stack - Загрузить регистр из стека	Rd←STACK	Нет	2

По-битовые команды и команды тестирования битов

Мне-мо-ника	Опера нды	Описание	Операция	Флаги	К-во цик-лов
LSL	Rd	Logical Shift Left- Логически сдвинуть влево	Rd(n+1)←Rd(n), Rd(0)← 0, C←Rd(7)	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right- Логически сдвинуть вправо	Rd(n)← Rd(n+1), Rd(7)←0, C←Rd(0)	Z,C,N,V	1
ROL		Rotate Left trough Carry- Сдвинуть влево через перенос	Rd(0)←C, Rd(n+1)←Rd(n), C← Rd(7)	Z,C,N,V,H	1
ROR	Rd	Rotate Right trough Carry- Сдвинуть вправо через перенос	Rd(7)←C, Rd(n)←Rd(n+1), C← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right- Арифметически сдвинуть вправо	Rd(n)← Rd(n+1), n=0...6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles - Поменять нибблы местами	Rd(3...0)←→Rd(7...4)	Нет	1
BSET	s	Flag Set -Установить флаг	SREG(s)←1	SREG(s)	1
BCLR	s	Flag Clear -Очистить флаг	SREG(s)←0	SREG(s)	1
SBI	P,b	Set bit to I/O Register - Установить бит в регистр I/O	I/O (P,b)←1	Нет	2

CBI	P,b	Clear Bit in I/O Register - Очистить бит в регистре I/O	I/O(P,b)←0	Нет	2
BST	Rd,b	Bit Store from Register to T - Переписать бит из регистра во флаг T	T← Rd(b)	T	1
BLD	Rd,b	Bit Load from T to Register -Загрузить T флаг в бит регистра	Rd(b)←T	Нет	1
SEC		Set Carry Flag- Установить флаг переноса	C←1	C	1
CLC		Clear Carry Flag -Очистить флаг переноса	C←0	C	1
SEN		Set Negative Flag- Установить флаг отрицательного значения	N←1	N	1
CLN		Clear Negative Flag -Очистить флаг отрицательного значения	N←0	N	1
SEZ		Set Zero Flag- Установить флаг нулевого значения	Z←1	Z	1
CLZ		Clear Zero Flag - Очистить флаг нулевого значения	Z←0	Z	1
SEI		Set Global Interrupt Flag- Установить флаг глобального прерывания	I←1	I	1
CLI		Clear Global Interrupt Flag -Очистить флаг глобального прерывания	I←0	I	1
SES		Set Signed Flag- Установить флаг знака	S←1	S	1
CLS		Clear Signed Flag - Очистить флаг знака	S←0	S	1
SEV		Set Overflow Flag- Установить флаг переполнения	V←1	V	1
CLV		Clear Overflow Flag - Очистить флаг переполнения	V←0	V	1
SET		Set T Flag- Установить флаг T	T←1	T	1
CLT		Clear T Flag - Очистить флаг T	T←0	T	1
SEH		Set Half Carry Flag- Установить флаг полу переноса	H←1	H	1
CLH		Clear Half Carry Flag -Очистить флаг полу переноса	H←0	H	1
NOP		No Operation - Выполнить холостую команду		Нет	1
SLEEP		Sleep - Установить режим SLEEP	См. описание команды	Нет	1
WDR		Watchdog Reset- Сбросить сторожевой таймер	См. описание команды	Нет	1

ADC - Add with Carry - Сложить с переносом

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

Операция:
(i) Rd ←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: $Rd3 \bullet Rr3 + Rr3 + \overline{R3} + \overline{R3} \bullet Rd3$
Устанавливается если есть перенос из бита 3, в ином случае очищается
S: $N \oplus V$, Для проверок со знаком
V: $Rd7 \bullet Rr7 \bullet \overline{R7} + \overline{Rd7} \bullet \overline{Rr7} \bullet R7$
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: $\overline{Rd7} \bullet \overline{Rr7} \bullet \overline{R7} \bullet \overline{R7} \bullet \overline{R7} \bullet \overline{Rd7}$
Устанавливается если результат $00, в ином случае очищается
C: $Rd7 \bullet Rr7 + Rr7 + \overline{R7} + \overline{R7} \bullet Rd7$
Устанавливается если есть перенос из MSB результата,
в ином случае очищается

R: (Результат) соответствует Rd после выполнения команды

Пример:

```
                        ; СложитьR1 : R0 с R3 : R2
        add r2, r0      ; Сложить младший байт
        adc r3, r1      ; Сложить старший байт с переносом
```

Слов: 1 (2 байта)

Циклов: 1

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←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: $Rd3 \bullet Rr3 + Rr3 + \overline{R3} + \overline{R3} \bullet Rd3$
Устанавливается если есть перенос из бита 3, в ином случае очищается
S: $N \oplus V$, Для проверок со знаком
V: $Rd7 \bullet Rr7 \bullet \overline{R7} + \overline{Rd7} \bullet \overline{Rr7} \bullet R7$
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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $00, в ином случае очищается
C: $Rd7 \bullet Rr7 + Rr7 \bullet \overline{R7} + \overline{R7} \bullet Rd7$
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

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∈ {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
-	-	-	<=>	<=>	<=>	<=>	<=>

S: N ⊕ V, 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: $\overline{R15} \bullet \overline{R14} \bullet \overline{R13} \bullet \overline{R12} \bullet \overline{R11} \bullet \overline{R10} \bullet \overline{R9} \bullet \overline{R8} \bullet \overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $0000, в ином случае очищается
C: $\overline{R15} \bullet$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

AND - Logical AND- Выполнить логическое AND

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

Операция:
(i) Rd←Rd • Rr

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

16-разрядный код операции:

0010	00rd	dddd	rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

ANDI - Logical AND with Immediate -
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←Rd • K

	Синтаксис	Операнды:	Счетчик программ:
(i)	ANDI Rd,K	16 ≤ d ≤31, 0 ≤ K ≤255	PC←PC + 1

16-разрядный код операции:

0111	KKKK	dddd	KKKK

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB,
в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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
```

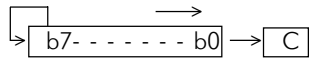
Слов: 1 (2 байта)

Циклов: 1

ASR - Arithmetic Shift Right- 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:
(i)



	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
-	-	-	<=>	<=>	<=>	<=>	<=>

S: N ⊕ V, Для проверок со знаком
V: N⊕C (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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

BCLR - Bit Clear in SREG -
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
<=>	<=>	<=>	<=>	<=>	<=>	<=>	<=>

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

BLD - Bit Load from the T Flag in SREG to a bit in Register - 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	H	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

BRBC - Branch if Bit in SREG is Cleared -
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	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

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

Слов: 1 (2 байта)

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

BRBS - Branch if Bit in SREG is Set-
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 состояния 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	00kk	kkkk	ksss

Булевы выражения регистра статуса (SREG)

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

Пример:

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

Слов: 1 (2 байта)

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

BRCC - Branch if Carry Cleared-
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    nocarry         ; Перейти если перенос очищен
                .....
nocarry:        nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BRCS - Branch if Carry Set-
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:
```
        cpi     r26, $56        ; Сравнить r26 с $56
        brcs    carry           ; Перейти если перенос установлен
        .....
carry:  nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BREQ - Branch if Equal- 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:
```
        cp      r1, r0          ; Сравнить регистры r1 и r0
        breq    equal           ; Перейти если содержимое регистров совпадает
        .....
equal:  nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BRGE - Branch if Greater or Equal (Signed)-
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 ≤ 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 (N⊕ V = 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	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Пример:

```
                cp      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

BRHC - Branch if Half Carry Flag is Cleared -
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 ≤ 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	H	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

BRHS - Branch if Half Carry Flag is Set-
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

BRID - Branch if Global Interrupt is Disabled -
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

BRIE - Branch if Global Interrupt is Enabled - 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 (PC64≤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←PC + k +1, else PC←PC +1

	Синтаксис	Операнды:	Счетчик программ:
(i)	BRIE k	-64 ≤ k ≤+63	PC←PC + k + 1 PC←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

BRLO - Branch if Lower (Unsigned) -
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←PC + k +1, else PC←PC +1

	Синтаксис	Операнды:	Счетчик программ:
(i)	BRLO 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:
```
        eor     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)
        nop                         ; Выйти из петли (пустая операция)
```

Слов: 1 (2 байта)

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

BRLT - Branch if Less Then (Signed) - 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 (N⊕V = 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        nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BRMI - Branch if Minus - 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 (PC64≤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 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

BRNE - Branch if Not Equal- 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 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
            brne    loop            ; Перейти если r27 <> 5
            nop                     ; Выйти из петли (пустая операция)
```

Слов: 1 (2 байта)

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

BRPL - Branch if Plus- 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 ≤ 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 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:       nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BRSH - Branch if Same or Higher (Unsigned) - 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≤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 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

BRTC - Branch if T Flag is Cleared -
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 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

BRTS - Branch if T Flag is Set - 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 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

BRVC - Branch if Overflow Cleared-
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

BRVS - Branch if Overflow Set -
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 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:         nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

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

BSET - Bit Set in SREG -
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
<=>	<=>	<=>	<=>	<=>	<=>	<=>	<=>

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

BST - Bit Store from Bit in Register to T flag in SREG - 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←Rd(b)

	Синтаксис	Операнды:	Счетчик программ:
(i)	BST Rd,b	0 ≤ d ≤31, 0 ≤ b ≤7	PC←PC + 1

16-разрядный код операции:

1111	101d	dddd	Xbbb

Булевы выражения регистра статуса (SREG)

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

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 4регистра r0
```

Слов: 1 (2 байта)

Циклов: 1

CALL - Long Call to a Subroutine -
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 ← k Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128K.
(ii) PC ← k Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8M.

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

32-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:          cpi     r16, $42        ; Проверить содержит ли r16 заданное значение
                breq    error           ; Перейти если содержит
                ret                     ; Вернуться из подпрограммы
                . . .
error:          rjmp    error           ; Бесконечная петля
```

Слов: 2 (4 байта)

Циклов: 4

CBI - Clear Bit in I/O Register - 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

CBR - Clear Bits in Register -
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,K	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
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $00, в ином случае очищается

R (Результат) соответствует Rd после выполнения команды

Пример:
```
cbr     r16, $F0      ; Очистить старший ниббл регистра r16
cbr     r18, 1        ; Очистить бит в r18
```

Слов: 1 (2 байта)

Циклов: 1

CLC - Clear Carry Flag -
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

CLH - Clear Half Carry Flag -
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

CLI - Clear Global Interrupt Flag - 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    ; Считать Порт B
sei                 ; Разрешить прерывания
```

Слов: 1 (2 байта)

Циклов: 1

CLN - Clear Negative Flag -
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

CLR - Clear Register- 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 ⊕ Rd

	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
       . . .
       cpi     r18, $50        ; Сравнить r18 с $50
       brne    loop
```

Слов: 1 (2 байта)

Циклов: 1

CLS - Clear Signed Flag -
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

CLT - Clear T Flag -
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

CLV - Clear Overflow Flag -
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

CLZ - Clear Zero Flag -
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

COM- One's Complement -
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
-	-	-	<=>	0	<=>	<=>	1

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{Rd7} \bullet \overline{Rd6} \bullet \overline{Rd5} \bullet \overline{Rd4} \bullet \overline{Rd3} \bullet \overline{Rd2} \bullet \overline{Rd1} \bullet \overline{Rd0}$
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

CP - Compare - 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←PC + 1

16-разрядный код операции:

0001	01rd	dddd	rrrr

Булевы выражения регистра статуса (SREG)

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

H: $\overline{Rd3}$•Rr3+Rr3•R3+R3 • $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7 • $\overline{Rd7}$ • $\overline{R7}$+$\overline{Rd7}$•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: $\overline{Rd7}$•Rr7+Rr7•R7+R7 • $\overline{Rd7}$
Устанавливается если абсолютное значение Rr больше абсолютного значения Rd, в ином случае очищается
R (Результат) после выполнения команды

Пример:

```
        cp      r4, r19         ; Сравнить r4 с r19
        brne    noteq           ; Перейти если r4 <> r19
        . . .
noteq:  nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

Циклов: 1

CPC - Compare with Carry - 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	rrrr

Булевы выражения регистра статуса (SREG)

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

H: $\overline{Rd3}$•Rr3+Rr3•R3+R3• $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7 • $\overline{Rr7}$ • $\overline{R7}$+$\overline{Rd7}$•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: $\overline{R7}$ • $\overline{R6}$ • $\overline{R5}$ • $\overline{R4}$ • $\overline{R3}$ • $\overline{R2}$ • $\overline{R1}$ • $\overline{R0}$ • Z
Предшествующее значение остается неизменным если результатом является ноль, в ином случае очищается
C: $\overline{Rd7}$•Rr7+Rr7•R7+R7 • $\overline{Rd7}$
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
            cp      r2, r0      ; Сравнить старший байт
            cpc     r3, r1      ; Сравнить младший байт
            brne    noteq       ; Перейти если не равно
            . . .
noteq:      nop                 ; Перейти по назначению (пустая операция)
```
Слов: 1 (2 байта)
Циклов: 1

CPI - Compare with Immediate- 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
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H: $\overline{Rd3} \bullet K3+K3 \bullet R3+R3 \bullet \overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N⊕ V, Для проверок со знаком
V: $Rd7 \bullet \overline{K7} \bullet \overline{R7} + \overline{Rd7} \bullet K7 \bullet R7$
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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $00, в ином случае очищается
C: $\overline{Rd7} \bullet K7+K7 \bullet R7+R7 \bullet \overline{Rd7}$
Устанавливается если абсолютное значение K больше абсолютного значения Rd, в ином случае очищается
R (Результат) после выполнения команды

Пример:

```
            cpi     r19, 3          ; Сравнить r19 с 3
            brne    error           ; Перейти если r4 <> 3
            . . .
error:      nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

Циклов: 1

CPSE- Compare Skip if Equal-
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

DEC - Decrement - 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
-	-	-	<=>	<=>	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: $\overline{R7} \bullet R6 \bullet R5 \bullet R4 \bullet R3 \bullet R2 \bullet R1 \bullet R0$
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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

EOR- Exclusive OR -
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⟵Rd ⊕ Rr

	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
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $00, в ином случае очищается

R (Результат) соответствует Rd после выполнения команды

Пример:
```
eor     r4, r4          ; Очистить r4
eor     r0, r22         ; Поразрядно выполнить исключающее or между r0 и r22
```

Слов: 1 (2 байта)

Циклов: 1

ICALL - Indirect Call to Subroutine - 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 64K слов (128 Кбайт).

Операция:

(i) PC(15-0)⟵Z(15-0) Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128K.

(ii) PC(15-0)⟵Z(15-0) Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8M.
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)
(ii)	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

IJMP - Indirect Jump -
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 64K слов (128 Кбайт).

Операция:
(i) PC⟵Z(15-0) Приборы с 16-разрядным счетчиком программ, максимальный объем памяти программ 128K.
(ii) PC(15-0)⟵Z(15-0) Приборы с 22-разрядным счетчиком программ, максимальный объем памяти программ 8M.
PC(21-16) не изменяются

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

IN - Load an I/O Port to Register - 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) Rd←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 B
            cpi     r25, r4         ; Сравнить считанное значение с константой
            breq    exit            ; Перейти если r25=4
            . . .
exit:       nop                     ;  Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

Циклов: 1

INC - Increment - 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

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

16-разрядный код операции:

1001	010d	dddd	0011

Булевы выражения регистра статуса (SREG)

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

S: N ⊕ V, Для проверок со знаком
V: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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
            brne    loop            ; Перейти если не равно
            nop                     ; Продолжать (пустая операция)
```

Слов: 1 (2 байта)
Циклов: 1

JMP - Jump - Перейти

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

Операция:
(i) PC←k

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	JMP k	0 ≤ k ≤ 4M	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: 3

LD - Load Indirect from SRAM to Register using Index X -
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←X - 1	Rd←(X)	X: Предварительно декрементирован

	Синтаксис	Операнды:	Счетчик программ:
(i)	LD Rd,X	0 ≤ d ≤ 31	PC←PC + 1
(ii)	LD Rd,X+	0 ≤ d ≤ 31	PC←PC + 1
(iii)	LDD Rd,-X	0 ≤ d ≤ 31	PC←PC + 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

LD (LDD) - Load Indirect from SRAM to Register using Index Y - 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←(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←PC + 1
(ii)	LD Rd, Y+	0 ≤ d ≤ 31	PC←PC + 1
(iii)	LD Rd,-Y	0 ≤ d ≤ 31	PC←PC + 1
(iv)	LDD Rd, Y + q	0 ≤ d ≤ 31, 0 ≤ q ≤ 63	PC←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

LD (LDD) - Load Indirect from SRAM to Register using Index Z - 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←(Z)		Z: Неизменен
(ii)	Rd←(Z)	Z←Z + 1	Z: Инкрементирован впоследствии
(iii)	Z←Z - 1	Rd←(Z)	Z: Предварительно декрементирован
(iv)	Rd←(Z + q)		Z: Неизменен, q: смещение
	Синтаксис	Операнды:	Счетчик программ:
(i)	LD Rd, Z	0 ≤ d ≤ 31	PC←PC + 1
(ii)	LD Rd, Z+	0 ≤ d ≤ 31	PC←PC + 1
(iii)	LD Rd,-Z	0 ≤ d ≤ 31	PC←PC + 1
(iv)	LDD Rd, Z + q	0 ≤ d ≤ 31, 0 ≤ q ≤63	PC←PC + 1

16-разрядный код операции:

(i)	1000	000d	dddd	0000
(ii)	1001	000d	dddd	0001
(iii)	1001	000d	dddd	0010
(iv)	10q0	qq0d	dddd	0qqq

Булевы выражения регистра статуса (SREG)

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

Пример:
```
clr   r31     ;Очистить старший байт Z
ldi   r30, $20 ;Установить $20 в младший байт Z
ld    r0, Z+  ;Загрузить в r0 содерж. SRAM по адресу $20 (Z постинкрементируется)
ld    r1, Z   ;Загрузить в r1 содержимое SRAM по адресу $21
ldi   r30, $23 ;Установить $23 в младший байт Z
ld    r2, Z   ;Загрузить в r2 содержимое SRAM по адресу $23
ld    r3, -Z  ;Загрузить в r3 содерж. SRAM по адресу $22 (Z преддекрементируется)
ldd   r4, Z+2 ;Загрузить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта) Циклов: 2

LDI - Load Immediate -
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←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

LDS- Load Direct from SRAM -
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←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

LPM - Load Program Memory - 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←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)
```

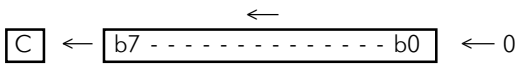
Слов: 1 (2 байта)

Циклов: 3

LSL- Logical Shift Left- 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:
(i)



	Синтаксис	Операнды:	Счетчик программ:
(i)	LSL Rd	0 ≤ d ≤31	PC←PC + 1

16-разрядный код операции:

0000	11dd	dddd	dddd

Булевы выражения регистра статуса (SREG)

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

H: Rd3
S: N ⊕ V, Для проверок со знаком
V: N ⊕ C (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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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
```

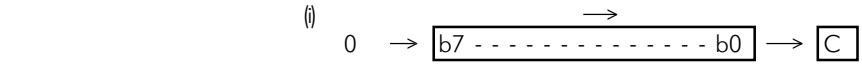
Слов: 1 (2 байта)

Циклов: 1

LSR - Logical Shift Right - Логически сдвинуть вправо

Описание:
Сдвиг всех битов Rd на одно место вправо. Бит 7очищается. 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:
(i)



	Синтаксис	Операнды:	Счетчик программ:
(i)	LSR Rd	0 ≤ d ≤ 31	PC←PC + 1

16-разрядный код операции:

1001	010d	dddd	0110

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	<=>	<=>	0	<=>	<=>

S: N ⊕ V, Для проверок со знаком
V: N ⊕ C (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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

MOV - Copy Register -
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.

Операция:
(i) Rd← Rr

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

16-разрядный код операции:

0010	11rd	dddd	rrrr

Булевы выражения регистра статуса (SREG)

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

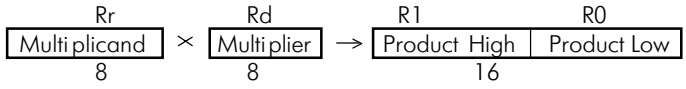
Пример:
```
        mov     r16, r0         ; Копировать r0 в r16
        call    check           ; Вызвать подпрограмму
        . . .
check   cpi     r16, $11        ; Сравнить r16 с $11
        . . .
        ret                     ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 1

MUL - Multiply- 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	rrrr

Булевы выражения регистра статуса (SREG)

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

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.

NEG - Two's Complement - 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 Rd	0 ≤ 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
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H: $R3 \bullet \overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: $N \oplus V$, Для проверок со знаком
V: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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: $\overline{Rd7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

NOP - No Operation -
Выполнить холостую команду

Описание:
Команда выполняется за один цикл без выполнения операции

Операция:
(i) No

	Синтаксис	Операнды:	Счетчик программ:
(i)	NOP	None	PC←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 B
nop                     ; Ожидать (пустая операция)
out     $18, r17        ; Записать 1 в Порт B
```

Слов: 1 (2 байта)

Циклов: 1

OR - Logical OR - Выполнить логическое OR

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

Операция:
(i) Rd←Rd v Rr

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

16-разрядный код операции:

0010	10rd	dddd	rrrr

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, 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:
```
        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

ORI - Logical OR with Immediate - Выполнить логическое OR с непосредственным значением

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

Операция:
(i) Rd←Rd v K

	Синтаксис	Операнды:	Счетчик программ:
(i)	ORI 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
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

OUT - Store Register to I/O port -
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←PC + 1

16-разрядный код операции:

1011	1PPr	rrrr	PPPP

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 нули в Порт B
nop                     ; Ожидать (пустая операция)
out     $18, r17        ; Записать единицы в Порт B
```

Слов: 1 (2 байта)

Циклов: 1

POP - Pop Register from Stack -
Загрузить 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←PC + 1SP←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
                ret                     ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 2

PUSH - Push Register on Stack -
Поместить регистр в стек

Описание:
Команда помещает содержимое регистра Rd в стек.

Операция:
(i) STACK← Rr

	Синтаксис	Операнды:	Счетчик программ:
(i)	PUSH Rr	0 ≤ r ≤ 31	PC←PC + 1SP←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
                ret                     ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 2

RCALL - Relative Call to Subroutine - 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
            ret                     ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 3

RET - Return from Subroutine -
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, 22бита)

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
                ret                     ; Вернуться из подпрограммы
```

Слов: 1 (2 байта)

Циклов: 4

RETI - Return from Interrupt -
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, 22бита)

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

RJMP - Relative Jump -
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←PC + k + 1

	Синтаксис	Операнды:	Счетчик программ:	Стек
(i)	RJMP k	-2K ≤ k ≤ 2K	PC←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
-	-	-	-	-	-	-	-

Пример:

```
            cpi     r16, $42        ; Сравнить r16 с $42
            brne    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)
```

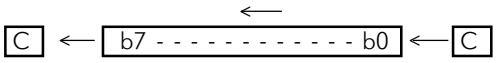
Слов: 1 (2 байта)

Циклов: 2

ROL - Rotate Left trough Carry - 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:



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

16-разрядный код операции:

0001	11dd	dddd	dddd

Булевы выражения регистра статуса (SREG)

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

H: Rd3
S: N ⊕ V, Для проверок со знаком
V: N ⊕ C (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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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)
```

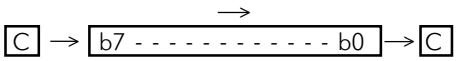
Слов: 1 (2 байта)

Циклов: 1

ROR - Rotate Right trough Carry - 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:



	Синтаксис	Операнды:	Счетчик программ:
(i)	ROR Rd	0 ≤ d ≤ 31	PC←PC + 1

16-разрядный код операции:

1001	010d	dddd	0111

Булевы выражения регистра статуса (SREG)

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

S: N ⊕ V, Для проверок со знаком
V: N ⊕ C (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: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

SBC- Subtract with Carry - 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
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H: $\overline{Rd3}$•Rr3+Rr3•R3+R3 • $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7 • $\overline{Rr7}$ • $\overline{R7}$+$\overline{Rd7}$•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: $\overline{R7}$ • $\overline{R6}$ • $\overline{R5}$ • $\overline{R4}$ • $\overline{R3}$ • $\overline{R2}$ • $\overline{R1}$ • $\overline{R0}$ •Z
Предшествовавшее значение остается неизменным если результат равен нулю, в ином случае очищается
C: $\overline{Rd7}$•Rr7+Rr7 • R7+ R7 • $\overline{Rd7}$
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

SBCI - Subtract Immediate with Carry -
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	16 ≤ 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
-	-	<=>	<=>	<=>	<=>	<=>	<=>

H: $\overline{Rd3}$•K3+K3•R3+R3 • $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7 • $\overline{K7}$ • $\overline{R7}$+$\overline{Rd7}$•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: $\overline{R7}$ • $\overline{R6}$ • $\overline{R5}$ • $\overline{R4}$ • $\overline{R3}$ • $\overline{R2}$ • $\overline{R1}$ • $\overline{R0}$ • Z
Предшествовавшее значение остается неизменным если результат равен нулю, в ином случае очищается
C: $\overline{Rd7}$•K7+K7•R7+ R7 • $\overline{Rd7}$
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

SBI - Set bit to I/O Register - 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

	Синтаксис	Операнды:	Счетчик программ:
(i)	SBI P,b	0 ≤ P ≤31, 0 ≤ b ≤ 7	PC←PC + 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         ; Считать данные EEPROM
```

Слов: 1 (2 байта)

Циклов: 2

SBIC - Skip if Bit I/O Register is Cleared -
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)	SBIC 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 1слово 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 EEWE 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

SBIS - Skip if Bit I/O Register is Set - 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, пропускает команду длиной в одно 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

SBIW - Subtract Immediate from Word -
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

	Синтаксис	Операнды:	Счетчик программ:
(i)	SBIW Rdl,K	dl∈ {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
-	-	-	<=>	<=>	<=>	<=>	<=>

S: N ⊕ V, Для проверок со знаком
V: Rdh7 • $\overline{R15}$
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: $\overline{R15} \bullet \overline{R14} \bullet \overline{R13} \bullet \overline{R12} \bullet \overline{R11} \bullet \overline{R10} \bullet \overline{R9} \bullet \overline{R8} \bullet \overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Устанавливается если результат $0000, в ином случае очищается
C: R15 • $\overline{Rdh7}$
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

SBR- Set Bits in Register - 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

	Синтаксис	Операнды:	Счетчик программ:
(i)	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
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

SBRC - Skip if Bit in Register is Cleared -
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 1слово PC←PC + 3, если следующие команды JMP или CALL

16-разрядный код операции:

1111	110r	rrrr	Xbbb

Булевы выражения регистра статуса (SREG)

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

Пример:

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

Слов: 1 (2 байта)

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

SBRS - Skip if Bit in Register is Set - 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, пропускает команду длиной в одно слово PC←PC + 3, пропускает команды JMP или CALL

16-разрядный код операции:

1111	111r	rrrr	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
nop                     ; Продолжать (пустая операция
```

Слов: 1 (2 байта)

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

SEC - Set Carry Flag -
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

SEH - Set Half Carry Flag -
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

SEI - Set Global Interrupt Flag -
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    ; Считать Порт B
sei                 ; Разрешить прерывания
```

Слов: 1 (2 байта)

Циклов: 1

SEN - Set Negative Flag -
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

SER - Set all bits in Register -
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
ser     r17             ; Установить r17
out     #18, r16        ; Записать нули в Порт B
nop                     ; Задержка (пустая операция)
out     #18, r17        ; Записать единицы в Порт B
```

Слов: 1 (2 байта)

Циклов: 1

SES - Set Signed Flag -
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

SET - Set T Flag -
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

SEV - Set Overflow Flag -
Установить флаг переполнения

Описание:
Команда устанавливает флаг переполнения (V) в регистре статуса (SREG)

Операция:
(i) V←1

	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

SEZ - Set Zero Flag -
Установить флаг нулевого значения

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

Операция:
(i) Z←1

	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

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 отработает 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

ST- Store Indirect from Register to SRAM using Index X - 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	rrrr	1100
(ii)	1001	001r	rrrr	1101
(iii)	1001	001r	rrrr	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

ST (STD)- Store Indirect from Register to SRAM using Index Y - 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 (Yпостинкрементируется)
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 (Yпреддекрементируется)
std   Y+2, r4   ;Сохранить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта)
Циклов: 2

ST (STD) - Store Indirect from Register to SRAM using Index Z - Записать косвенно из регистра в СОЗУ с использованием индекса Z

Описание:

Записывается косвенно, со смещением или без смещения, один байт из регистра в СОЗУ. 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.

Использование Z-указателя:

	Операция:		Комментарий:
(i)	(Z)⟵Rr		Z: Неизменен
(ii)	(Z)⟵Rr	Z⟵Z + 1	Z: Инкрементирован впоследствии
(iii)	Z⟵Z - 1	(Z)⟵Rr	Z: Предварительно декрементирован
(iv)	(Z + q)⟵Rr		Z: Неизменен, q: смещение
	Синтаксис	Операнды:	Счетчик программ:
(i)	ST Z,Rr	0 ≤ r ≤ 31	PC⟵PC + 1
(ii)	ST Z+,Rr	0 ≤ r ≤ 31	PC⟵PC + 1
(iii)	ST -Z,Rr	0 ≤ r ≤ 31	PC⟵PC + 1
(iv)	STd Z + q,Rr	0 ≤ r ≤31, 0 ≤ q ≤63	PC⟵PC + 1

16-разрядный код операции:

(i)	1000	001r	rrrr	0000
(ii)	1001	001r	rrrr	0001
(iii)	1001	001r	rrrr	0010
(iiii)	10q0	qq1r	rrrr	0qqq

Булевы выражения регистра статуса (SREG)

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

Пример:
```
clr r31        ; Очистить старший байт Z
ldi r30, $20   ; Установить $20 в младший байт Z
st Z+,r0       ;Сохранить в r0 содерж. SRAM по адр. $20(Zпостинкрементируется)
st Z, r1       ; Сохранить в r1 содержимое SRAM по адресу $21
ldi r30, $23   ; Установить $23 в младший байт Z
st Z, r2       ; Сохранить в r2 содержимое SRAM по адресу $23
st -Z, r3      ; Сохранить в r3 содерж.SRAM по адр. $22(Zпреддекрементируется)
std Z+2, r4    ; Сохранить в r4 содержимое SRAM по адресу $24
```

Слов: 1 (2 байта)
Циклов: 2

STS- Store Direct to RAM - 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

SUB- Subtract without Carry -
Вычесть без переноса

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

Операция:
(i) Rd⟵Rd - Rr

	Синтаксис	Операнды:	Счетчик программ:
(i)	SUB Rd,Rr	0 ≤ d ≤31, 0 ≤ r ≤31	PC⟵PC + 1

16-разрядный код операции:

0001	10rd	dddd	rrrr

Булевы выражения регистра статуса (SREG)

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

H: $\overline{Rd3}$•Rr3+Rr3•R3+R3 • $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7• $\overline{Rr7}$ •$\overline{R7}$+$\overline{Rd7}$ 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: $\overline{R7}$ • $\overline{R6}$ • $\overline{R5}$ • $\overline{R4}$ • $\overline{R3}$ • $\overline{R2}$ • $\overline{R1}$ • $\overline{R0}$
Устанавливается если результат равен $00, в ином случае очищается
C: $\overline{Rd7}$•$\overline{Rr7}$+$\overline{Rr7}$•$\overline{R7}$+ $\overline{R7}$• $\overline{Rd7}$
Устанавливается если абсолютное значение содержимого Rr больше, чем абсолютное значение Rd, в ином случае очищается

R (Результат) соответствует Rd после выполнения команды

Пример:
```
        sub     r13, r12        ; Вычесть r12 из r13
        brne    noteq           ; Перейти если r12 <> r13
noteq:  nop                     ; Перейти по назначению (пустая операция)
```

Слов: 1 (2 байта)

Циклов: 1

SUBI - Subtract Immediate - Вычесть непосредственное значение

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

Операция:
(i) Rd←Rd - K

	Синтаксис	Операнды:	Счетчик программ:
(i)	SUBI Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC←PC + 1

16-разрядный код операции:

0101	KKKK	dddd	KKKK

Булевы выражения регистра статуса (SREG)

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

H: $\overline{Rd3}$•K3+K3•R3+R3 • $\overline{Rd3}$
Устанавливается если есть заем из бита 3, в ином случае очищается
S: N ⊕ V, Для проверок со знаком
V: Rd7 • $\overline{K7}$ • $\overline{R7}$+$\overline{Rd7}$•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: $\overline{R7}$ • $\overline{R6}$ • $\overline{R5}$ • $\overline{R4}$ • $\overline{R3}$ • $\overline{R2}$ • $\overline{R1}$ • $\overline{R0}$
Устанавливается если результат равен $00, в ином случае очищается
C: $\overline{Rd7}$•K7+K7•R7+ R7 • $\overline{Rd7}$
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

SWAP - Swap Nibbles -
Поменять нибблы местами

Описание:
Команда меняет местами старший и младший нибблы (полубайты) регистра

Операция:
(i) R(7-4)⟵Rd(3-0), R(3-0)⟵Rd(7-4)

	Синтаксис	Операнды:	Счетчик программ:
(i)	SWAP Rd	0 ≤ d ≤ 31	PC⟵PC + k + 1

16-разрядный код операции:

1001	010d	dddd	0010

Булевы выражения регистра статуса (SREG)

I	T	H	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

TST - Test for Zero or Minus - 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←Rd • Rd

	Синтаксис	Операнды:	Счетчик программ:
(i)	TST Rd	0 ≤ d ≤ 31	PC←PC + 1

16-разрядный код операции:

0010	00dd	dddd	dddd

Булевы выражения регистра статуса (SREG)

I	T	H	S	V	N	Z	C
-	-	-	<=>	0	<=>	<=>	-

S: N ⊕ V, Для проверок со знаком
V: 0
Очищен
N: R7
Устанавливается если в результате установлен MSB, в ином случае очищается
Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
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

WDR- Watchdog Reset -
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)

	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