Регистр статуса - SREG - размещен в пространстве I/O по адресу \$3F (\$5F) и его биты определяются как:

Биты	7	6	5	4	3	2	1	0	
\$3F (\$5F)	I	Т	Н	S	V	N	Z	C	REG
Чтение/Запись	R/W								
Начальное состояние	0	0	0	0	0	0	0	0	

- Bit 7 I: Global Interrupt Enable Разрешение глобального прерывания. Бит разрешения глобального прерывания для разрешения прерывания должен быть установлен в состояние 1. Управление разрешением конкретного прерывания выполняется регистрами маски прерывания GIMSK и TIMSK. Если бит глобального прерывания очищен (в состоянии 0), то ни одно из разрешений конкретных прерываний, установленных в регистрах GIMSK и TIMSK, не действует. Бит I аппаратно очищается после прерывания и устанавливается для последующего разрешения глобального прерывания командой RETI.
- Bit 6 T: Bit Copy Storage Бит сохранения копии. Команды копирования бита BLD (Bit LoaD) и BST (Bit STore) используют бит Т как бит источник и бит назначения при операциях с битами. Командой BST бит регистра регистрового файла копируется в бит Т, командой BLD бит Т копируется в регистр регистрового файла.
- Bit 5 H: Half Carry Flag Флаг полупереноса Флаг полупереноса указывает на полуперенос в ряде арифметических операций.
- Bit 4 S: Sign Bit, S = N eor V Бит знака Бит S всегда находится в состоянии, определяемом логическим исключающим ИЛИ (exclusive OR) между флагом отрицательного значения N и дополнением до двух флага переполнения V.
- Bit 3 V: Two's Complement Overflow Flag Дополнение до двух флага переполнения Дополнение до двух флага переполнения V поддерживает арифметику дополнения до двух.
- Bit 2 N: Negative Flag Флаг отрицательного значения Флаг отрицательного значения N указывает на отрицательный результат ряда арифметических и логических операций.
- Bit 1 Z: Zero Flag Флаг нулевого значения Флаг нулевого значения Z указывает на нулевой результат ряда арифметических и логических операций.
- Bit 0 C: Carry Flag Флаг переноса Флаг переноса С указывает на перенос в арифметических и логических операциях.

Название	Операнды	Описание	Операция	Флаги	Циклы
ADD	Rd, Rr	Сложение без переноса	Rd = Rd + Rr	Z,C,N,V,H,S	1
ADC	Rd, Rr	Сложение с переносом	Rd = Rd + Rr + C	Z,C,N,V,H,S	1
SUB	Rd, Rr	Вычитание без переноса	Rd = Rd - Rr	Z,C,N,V,H,S	1
SUBI*	Rd, K8	Вычитание константы	Rd = Rd - K8	Z,C,N,V,H,S	1
SBC	Rd, Rr	Вычитание с переносом	Rd = Rd - Rr - C	Z,C,N,V,H,S	1

SBCI*	Rd, K8	Вычитание константы с переносом	Rd = Rd - K8 - C	Z,C,N,V,H,S	1
INC	Rd	Инкремент	Rd = Rd + 1	Z,N,V,S	1
DEC	Rd	Декремент	Rd = Rd -1	Z,N,V,S	1
NEG	Rd	Дополнение до двух	Rd=0x00-Rd	Z,C,N,V,H,S	1
MUL	Rd, Rr	Умножение	R1:R0 = Rd * Rr	Z,C	2
LDI*	Rd, K8	Загрузка константы	Rd = K	None	1
MOV	Rd, Rr	Копирование значения регистра	Rd = Rr	None	1

^{*}Операции с константами работают только с регистрами R16-R31

Директивы ассемблера

.DEF	Назначение символьного имени регистру		
.INCLUDE	Включение в код другого файла		
.EXIT	Прекращение компиляции файла		
.LIST	Включение генерации списочного файла		
.NOLIST	Выключение генерации списочного файла		
.CSEG	Начало сегмента кода		
.ORG	Установка позиции для дальнейшей записи кода		
.EQU	Назначение символьного эквивалента выражению		

Таблица 2 – Команды передачи данных

Название	Операнды	Описание	Операция	Флаги	Циклы
LDI*	Rd, K8	Загрузка константы	Rd = K	None	1
MOV	Rd, Rr	Копирование значения регистра	Rd = Rr	None	1
IN	Rd, P	Чтение регистра I/O	Rd = P	None	1
OUT	P, Rr	Запись регистра І/О	P = Rr	None	1

^{*}Операции с константами работают только с регистрами R16-R31

Таблица 3- Логические команды

	, <u>, </u>		1 -	_	
Название	Операнды	Описание	Операция	Флаги	Циклы
AND	Rd, Rr	Логическое И	Rd = Rd и Rr	Z,N,V,S	1
ANDI*	Rd, Rr	Логическое И с константой	Rd = Rd и K8	Z,N,V,S	1
OR	Rd, Rr	Логическое ИЛИ	Rd = Rd или Rr	Z,N,V,S	1
ORI*	Rd, K8	Логическое ИЛИ с константой	Rd = Rd или K8	Z,N,V,S	1
COM	Rd, Rr	Инверсия	Rd =0xFF - Rd	Z,N,V,S,C	1
EOR	Rd, K8	Логическое исключающее ИЛИ	Rd = Rd исключающее или Rr	Z,N,V,S	1
CLR	Rd	Очистка регистра	Rd = 0x00	Z,N,V,S	1
SER	Rd	Установка регистра	Rd = 0xFF		1

^{*}Операции с константами работают только с регистрами R16-R31

Таблица 3 – Битовые команды

Название	Операнды	Описание	Операция	Флаги	Циклы
SBI	P,b	Установка бита в регистре ввода-вывода	I/O(P,b) = 1	Нет	2
CBI	P,b	Сброс бита в регистре ввода-вывода	I/O(P,b) = 0	Нет	2
LSL	Rd	Логический сдвиг влево	Rd(n+1)=Rd(n), Rd(0)=0, C=Rd(7)	Z,C,N,V	1
LSR	Rd	Логический сдвиг вправо	Rd(n)=Rd(n+1), Rd(7)=0, C=Rd(0)	Z,C,N,V	1
ROL	Rd	Вращение влево через перенос	Rd(0)=C, Rd(n+1)=Rd(n), C=Rd(7)	Z,C,N,V	1
ROR	Rd	Вращение вправо через перенос	Rd(7)=C, Rd(n)=Rd(n+1), C=Rd(0)	Z,C,N,V	1
ASR	Rd	Арифметический сдвиг вправо	Rd(n)=Rd(n+1), n=0,,6	Z,C,N,V	1
SWAP	Rd	Обмен тетрадами	Rd(30) = Rd(74), Rd(74) = Rd(30)	Нет	1
BSET	S	Установка флага регистра SREG	SREG(s) = 1	SREG(s)	1
BCLR	S	Сброс флага регистра SREG	SREG(s) = 0	SREG(s)	1
BST	Rr, b	Запись бита регистра в Т	T = Rr(b)	T	1
BLD	Rd, b	Чтение из Т в бит регистра	Rd(b) = T	Нет	1
SEC		Установка переноса	C =1	С	1
CLC		Сброс переноса	C = 0	С	1
SEN		Установка флага N	N = 1	N	1
CLN		Сброс флага N	N = 0	N	1
SEZ		Установка флага нуля Z	Z = 1	Z	1
CLZ		Сброс флага нуля Z	Z = 0	Z	1
SEI		Общее разрешение прерываний	I = 1	I	1
CLI		Общий запрет прерываний	I = 0	I	1
SES		Установка флага S	S = 1	S	1
CLS		Сброс флага S	S = 0	S	1
SEV		Установка флага V в регистре SREG	V = 1	V	1
CLV		Сброс флага V в регистре SREG	V = 0	V	1
SET		Установка флага Т в регистре SREG	T = 1	T	1
CLT		Сброс флага Т в регистре SREG	T = 0	T	1
SEH		Установка флага Н в регистре SREG	H = 1	Н	1
CLH		Сброс флага H в регистре SREG	H = 0	Н	1

Таблица 1 – Команды

TST Rd	Название	лица 1 – ком Операнды	Описание	Операция	Флаги	Циклы
RJMP None Косаенный переход PC = PC + k + 1 None 2 JMP None Косаенный переход PC = Z None 2 JMP K Aбсолютный переход PC = X None 2 JMP K Aбсолютный переход PC = X None 3 RCALL K Относительный вызов STACK = PC+1, PC = PC None None	TST	Rd		Rd = Rd · Rd	Z,C,N,V,S	1
MMP	RJMP	k		PC = PC + k +1	None	2
MP		_	,	PC = Z		
RCALL К Относительный вызов подпрограммы нейт STACK = PC+1, PC = PC None 3/4* ICALL None Косеенный вызов подпрограммы подпрограммы STACK = PC+1, PC = Z None 3/4* CALL К Вызов подпрограммы STACK = PC+2, PC = k None 4/5* RET None Bosspar из подпрограммы PC = STACK None 4/5* RET None Bosspar из подпрограммы PC = STACK None 4/5* RET None Bosspar из подпрограммы PC = STACK None 4/5* CPSE Rd.Rr Cpasнить портустить если If (Rd = Rr) PC = PC 2 or долу None 1/2/3 CP Rd.Rr Сравнить в переносом Rd - Rr Z,C,N,V,H,S 1 CPC Rd.Rr Сравнить в переносом Rd - K Z,C,N,V,H,S 1 SBRC Rr.b Пропустить если бит в регистре общей (R(r(b)=0) PC = PC + 2 ог долу N,H,S 1 None 1/2/3 SBIC P.b Пропустить если бит в регистре общей (R(r(b)=0) PC = PC + 2 долу N,H,S 1 None <			Абсолютный переход	PC = k		
ICALL None подпрограммы STACK = PC+1, PC = k None 3/4"s RET None Boasde подпрограммы PC = STACK None 4/5"s RETI None Boaspar из подпрограммы PC = STACK None 4/5"s RETI None Boaspar из прерывания PC = STACK I 4/5"s CPSE Rd,Rr Сравнить регистите стали (Rd = Rr) PC = PC 2 or 3 and Rd - Rr None 1/2/3 CP Rd,Rr Сравнить с константой Rd - Rr Z,C,N,V,H,S 1 CPL Rd,Rr Сравнить с константой Rd - Rr Z,C,N,V,H,S 1 SBRC RI,D Пропустить если бит в регистре (Portugue) (RR(f(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре (Portugue) (RR(f(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре (Portugue) (RR(f(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре (Portugue) (RR(f(b)=b)=1) PC = PC + None <						
CALL k Вызов подпрограммы STACK = PC+2, PC = k None 4/5* RET None Bosspar из подпрограммы PC = STACK None 4/5* RETI None Bosspar из прерывания PC = STACK I 4/5* CPSE Rd,Rr Сравнить пропустить если If (Rd ==Rf) PC = PC 2 or 3 None 1/2/3 CP Rd,Rr Сравнить пропустить если Rd - Rr - C Z,C,N,V,H,S 1 CPC Rd,Rr Сравнить константой Rd - Rr - C Z,C,N,V,H,S 1 SBRC Rf,B Пропустить если бит в регистре очищен If(Rr(b)=0) PC = PC + 2 or 3 None 1/2/3 SBRS Rf,D Пропустить если бит в регистре ог и (If(O(P,b)=0) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре установлен If(Rr(b)=1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре установлен If(If(O(P,b)=1) PC = PC + k + 1 None 1/2/3 SBIS P,b Преход если флаг очищен If(SREG(s)=0) PC = PC + k + 1 <td><u>ICALL</u></td> <td>None</td> <td></td> <td>STACK = PC+1, PC = Z</td> <td>None</td> <td>3/4*</td>	<u>ICALL</u>	None		STACK = PC+1, PC = Z	None	3/4*
RETI None Возврат из прерывания PC = STACK I 4/5* CPSE Rd,Rr Сравнить, пропустить если раяны If (Rd = Rr) PC = PC 2 or None 1/2/3 CP Rd,Rr Сравнить Rd - Rr - C Z,C,N,V,H,S 1 CPC Rd,Rr Сравнить с константой Rd - Rr - C Z,C,N,V,H,S 1 CPI Rd,K8 Сравнить с константой Rd - K Z,C,N,V,H,S 1 SBRC Rr,b Пропустить если бит в регистре очищен (if(Rr(b)=0) PC = PC + 2 or 3 None 1/2/3 SBRS Rr,b Пропустить если бит в регистре (головия в головия в головия в головия в головия в головия (головия в головия в голови в головия в голови в головия в голови в головия в головия в головия в голо	CALL	<u>k</u>	Вызов подпрограммы	STACK = PC+2, PC = k	None	4/5*
CPSE Rd.Rr Сравнить, пропустить если равны If (Rd ==Rr) PC = PC 2 or равнить None 1/2/3 CP Rd.Rr Сравнить Rd - Rr Z,C,N,V,H,S 1 CPC Rd.Rr Сравнить в переносом Rd - Rr - C Z,C,N,V,H,S 1 CPI Rd.K8 Сравнить с константой Rd - K Z,C,N,V,H,S 1 SBRC Rr.b Пропустить если бит в регистре от дицен If(Rr(b)==0) PC = PC + 2 or 3 None 1/2/3 SBRS Rr.b Пропустить если бит в регистре (II(O)(P,b)==0) PC = PC + 2 or 3 None 1/2/3 SBIS P.b Пропустить если бит в регистре (II(O)(P,b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P.b Пропустить если бит в регистре (II(O)(P,b)==1) PC = PC + 2 or 3 None 1/2/3 BRBC S.k Переход если бит в регистре (II(O)(P,b)==0) PC = PC + 2 or 3 None 1/2/3 BRBS S.k Переход если бит в регистве (II(I)(II(I)(P,b)==0) PC = PC + R + 1 None 1/2/3 BRBS S.k Переход если фага переноса (II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I)(II(I	RET	None	Возврат из подпрограммы	PC = STACK	None	4/5*
СРЕ В КО, КГ равны 3 None 1/2/3 СРС Rd, КГ Сравнить в переносом Rd - Rr - C Z, C, N, V, H, S 1 СРІ Rd, КВ Сравнить с константой Rd - K Z, C, N, V, H, S 1 SBRC Rr, b Пропустить если бит в регистре онищен КВ - K Z, C, N, V, H, S 1 SBRS Rr, b Пропустить если бит в регистре онищен К(Rr(b)==1) PC = PC + 2 ог 3 None 1/2/3 SBIC P.b Пропустить если бит в регистре (VC очищен) К(R(D(P,b)==1)) PC = PC + 2 ог 3 None 1/2/3 SBIS P.b Пропустить если бит в регистре (VC очищен) К(R(REG(S)==0)) PC = PC + 2 ог 3 None 1/2/3 SBIS P.b Пропустить если бит в регистре (VC очищен) К(R(REG(S)==0)) PC = PC + 2 ог 3 None 1/2/3 BRBC S.k Переход если флаг очищен К(R(REG(S)==0)) PC = PC + 2 ог 3 None 1/2/3 BREQ k Переход если флаг очищен К(R(REG(S)==0)) PC = PC + None 1/2 BRNE k Переход если флаг переноса К(R(REG(S)==1)) PC = PC + None 1/2 BRC k Переход если флаг переноса К(С==0) PC = PC + k + 1<	RETI	None	Возврат из прерывания	PC = STACK	I	4/5*
СРС Rd,Rr Сравнить в переносом Rd - Rr - C Z,C,N,V,H,S 1 СРІ Rd,IXB Сравнить с константой Rd - K Z,C,N,V,H,S 1 SBRC Rr,b Пропустить если бит в регистре очищен If(Rr(b)==0) PC = PC + 2 or 3 None 1/2/3 SBRS Rr,b Пропустить если бит в регистре установлен If(Rr(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIC P,b Пропустить если бит в регистре установлен If(Rr(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре установлен If(ICOP(b)==0) PC = PC + 2 or 3 None 1/2/3 BRBC S,k Переход если флаг очищен If(SREG(s)==0) PC = PC + 2 or 3 None 1/2/3 BRBS S,k Переход если флаг очищен If(SREG(s)==0) PC = PC + None 1/2 BRBS S,k Переход если флаг очищен If(SREG(s)==1) PC = PC + None 1/2 BREQ k Переход если флаг переноса If(Z=-1) PC = PC + k + 1 None 1/2 BRCS k Переход если флаг переноса	CPSE	<u>Rd,Rr</u>	1		None	1/2/3
СРС Rd,Rr Сравнить в переносом Rd - Rr - C Z,C,N,V,H,S 1 СРІ Rd,K8 Сравнить с константой Rd - K Z,C,N,V,H,S 1 SBRC Rt,b Пропустить если бит в регистре очищен (п(Rr(b)==0) PC = PC + 2 or 3 None 1/2/3 SBRS Rt,b Пропустить если бит в регистре установлен (п(Rr(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIC P,b Пропустить если бит в регистре установлен (п(Rr(b)==1) PC = PC + 2 or 3 None 1/2/3 SBIS P,b Пропустить если бит в регистре установлен (п(Qr(b)==1) PC = PC + 2 or 3 None 1/2/3 BRBC s,k Переход если флаг очищен (п(SREG(s)==0) PC = PC + 2 or 3 None 1/2/3 BRBS s,k Переход если флаг очищен (п(SREG(s)==0) PC = PC + None 1/2 BRBS s,k Переход если флаг очищен (п(SREG(s)==1) PC = PC + None 1/2 BREQ k Переход если флаг переноса (п(SREG(s)==1) PC = PC + k+1 None 1/2 BRC k Переход если флаг переноса	CP	Rd,Rr	Сравнить	Rd -Rr	Z,C,N,V,H,S	1
CPI Rd,K8 Сравнить с константой Rd - K Z,C,N,V,H,S 1 SBRC Rr.b Пропустить если бит в регистре очищен (if(Rr(b)=0) PC = PC + 2 or 3 or 3 None 1/2/3 SBRS Rr.b Пропустить если бит в регистре установлен (if(Rr(b)=1) PC = PC + 2 or 3 or 3 None 1/2/3 SBIC P.b Пропустить если бит в регистре установлен (if(QP(b)==0) PC = PC + 2 or 3 or 3 None 1/2/3 SBIS P.b Пропустить если бит в регистре установлен (if(QP(b)==0) PC = PC + None 1/2/3 BRBC s.k Переход если флаг очищен (if(SREG(s)==0) PC = PC + None 1/2/3 BRBS s.k Переход если флаг очищен (if(Z==1) PC = PC + k+1 None 1/2 BREQ k Переход если флаг очищен (if(Z==0) PC = PC + k+1 None 1/2 BRCS k Переход если флаг переноса установлен (if(Z==1) PC = PC + k+1 None 1/2 BRCS k Переход если флаг переноса установлен (if(C==0) PC = PC + k+1 None 1/2 BRC k Переход если флаг	CPC		Сравнить в переносом	Rd - Rr - C		1
SBRC Rr.b Пропустить если бит в регистре очищен (f(Rr(b)==0) PC = PC + 2) cr 3 cr			Сравнить с константой	Rd - K		1
SBRS Rr.b Пропустить если бит в регистре установлен установлен установлен установлен (По очищен						1/2/3
SBIS P.b П/О очищен 2 or 3 None 1/2/3 BRBC S.k Переход если флаг очищен if(NP(P,b)==1) PC = PC + 2 or 3 None 1/2/3 BRBS S.k Переход если флаг очищен if(SREG(s)==0) PC = PC + k + 1 None None 1/2 BRBS S.k Переход если флаг очищен if(SREG(s)==0) PC = PC + k + 1 None None 1/2 BREQ k Переход если флаг перенова установлен if(Z==1) PC = PC + k + 1 None 1/2 BRNE k Переход если флаг переноса очищен if(C==0) PC = PC + k + 1 None 1/2 BRCC k Переход если флаг переноса очищен if(C==0) PC = PC + k + 1 None 1/2 BRSH k Переход если флаг переноса очищен if(C==0) PC = PC + k + 1 None 1/2 BRLO k Переход если флаг переноса очи меньше if(C==0) PC = PC + k + 1 None 1/2 BRLD k Переход если больше или равно или переход если больше или равно или переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None None 1/2 BRLT k Переход если флаг полуперноса очищен	<u>SBRS</u>	<u>Rr,b</u>			None	1/2/3
BRBC S.K Переход если флаг очищен 2 or 3 if(SREG(s)==0) PC = PC + k + 1 None 1/2 BRBS S.K Переход если флаг установлен if(SREG(s)==1) PC = PC + k + 1 None 1/2 BREQ К Переход если равны if(SREG(s)==1) PC = PC + k + 1 None 1/2 BRNE К Переход если флаг переноса установлен if(Z==1) PC = PC + k + 1 None 1/2 BRCS К Переход если флаг переноса очищен if(C==1) PC = PC + k + 1 None 1/2 BRCC К Переход если больше или равно if(C==0) PC = PC + k + 1 None 1/2 BRSH К Переход если больше или равно if(C==0) PC = PC + k + 1 None 1/2 BRLO К Переход если больше или равно if(C==1) PC = PC + k + 1 None 1/2 BRPL К Переход если больше или равно знаковое if(N==1) PC = PC + k + 1 None 1/2 BRPL К Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRHS К Переход если флаг полупереноса установлен if(S==1) PC = PC + k + 1 None 1/2 BRHS К Пе	SBIC	<u>P,b</u>			None	1/2/3
BRBS S.K Переход если флаг установлен к+1 K+1 None 1/2 BRBS S.K Переход если флаг установлен к+1 if(SREG(s)==1) PC = PC + k+1 None 1/2 BRVE K Переход если иравны if(Z==1) PC = PC + k+1 None 1/2 BRCS K Переход если флаг переноса установлен if(C==1) PC = PC + k+1 None 1/2 BRCC K Переход если флаг переноса установлен или равно if(C==0) PC = PC + k+1 None 1/2 BRSH K Переход если больше или равно if(C==0) PC = PC + k+1 None 1/2 BRLO K Переход если отрицательное if(N==1) PC = PC + k+1 None 1/2 BRMI K Переход если отрицательное if(N==0) PC = PC + k+1 None 1/2 BRPL K Переход если отрицательное if(N==0) PC = PC + k+1 None 1/2 BRGE K Переход если отрицательное if(N==0) PC = PC + k+1 None 1/2 BRHS K Переход если флаг if(S==0) PC = PC + k+1 None 1/2 <td><u>SBIS</u></td> <td><u>P,b</u></td> <td></td> <td></td> <td>None</td> <td>1/2/3</td>	<u>SBIS</u>	<u>P,b</u>			None	1/2/3
BREQ k Переход если равны if(Z=1) PC = PC + k + 1 None 1/2 BRNE k Переход если не равны if(Z=0) PC = PC + k + 1 None 1/2 BRCS k Переход если флаг переноса установлен if(C=1) PC = PC + k + 1 None 1/2 BRCC k Переход если флаг переноса очищен if(C=0) PC = PC + k + 1 None 1/2 BRSH k Переход если больше или равно if(C=0) PC = PC + k + 1 None 1/2 BRLO k Переход если меньше if(C=1) PC = PC + k + 1 None 1/2 BRMI k Переход если отрицательное if(N=1) PC = PC + k + 1 None 1/2 BRPL k Переход если больше или равно знаковое if(N=1) PC = PC + k + 1 None 1/2 BRGE k Переход если больше или равно знаковое if(S=0) PC = PC + k + 1 None 1/2 BRHS k Переход если флаг полуперноса установлен if(H=1) PC = PC + k + 1 None 1/2 BRHS k Переход если флаг Т установлен if(T=1) PC =	BRBC	<u>s,k</u>	Переход если флаг очищен	k+1	None	1/2
BRNE k Переход если не равны if(Z==0) PC = PC + k + 1 None 1/2 BRCS k Переход если флаг переноса установлен if(C==1) PC = PC + k + 1 None 1/2 BRCC k Переход если флаг переноса очищен if(C==0) PC = PC + k + 1 None 1/2 BRSH k Переход если больше или равно if(C==0) PC = PC + k + 1 None 1/2 BRMI k Переход если меньше if(N==1) PC = PC + k + 1 None 1/2 BRPL k Переход если отомительное if(N==1) PC = PC + k + 1 None 1/2 BRPL k Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT k Переход если флаг полупереноса установлен if(S==1) PC = PC + k + 1 None 1/2 BRHS k Переход если флаг полупереноса учщен if(H==1) PC = PC + k + 1 None 1/2 BRTS k Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRYS k Переход если флаг V учщен <t< td=""><td>BRBS</td><td><u>s,k</u></td><td>Переход если флаг установлен</td><td>` ` ` ,</td><td>None</td><td>1/2</td></t<>	BRBS	<u>s,k</u>	Переход если флаг установлен	` ` ` ,	None	1/2
BRCS к Переход если флаг переноса установлен if(C==1) PC = PC + k + 1 None 1/2 BRCC к Переход если флаг переноса очищен if(C==0) PC = PC + k + 1 None 1/2 BRSH к Переход если больше или равно if(C==0) PC = PC + k + 1 None 1/2 BRLO к Переход если меньше if(C==1) PC = PC + k + 1 None 1/2 BRMI к Переход если отрицательное if(N==1) PC = PC + k + 1 None 1/2 BRPL к Переход если отрицательное if(N==0) PC = PC + k + 1 None 1/2 BRGE к Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT к Переход если флаг полупереноса установлен if(S==1) PC = PC + k + 1 None 1/2 BRHS к Переход если флаг полупереноса очищен if(H==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Т установлен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V очищен	BREQ	<u>k</u>	Переход если равны	if(Z==1) PC = PC + k + 1	None	1/2
BRCC к установлен III(C==1) PC = PC + k+1 None 1/2 BRCC к Переход если флаг переноса очищен if(C==0) PC = PC + k+1 None 1/2 BRSH к Переход если больше или равно if(C==0) PC = PC + k+1 None 1/2 BRLO к Переход если отрицательное if(N==1) PC = PC + k+1 None 1/2 BRMI к Переход если отрицательное if(N==0) PC = PC + k+1 None 1/2 BRPL к Переход если больше или равно знаковое if(S==0) PC = PC + k+1 None 1/2 BRLT к Переход если флаг полупереноса установлен if(H==1) PC = PC + k+1 None 1/2 BRHS к Переход если флаг полупереноса очищен if(H==0) PC = PC + k+1 None 1/2 BRTS к Переход если флаг тустановлен if(T==0) PC = PC + k+1 None 1/2 BRVS к Переход если флаг Точищен if(V==0) PC = PC + k+1 None 1/2 BRVC к Переход если флаг Vочищен if(V==0) PC = PC + k+1	BRNE	<u>k</u>	Переход если не равны	if(Z==0) PC = PC + k + 1	None	1/2
BRSH К Переход если больше или равно if(C==0) PC = PC + k + 1 None 1/2 BRLO К Переход если меньше if(C==1) PC = PC + k + 1 None 1/2 BRMI К Переход если отрицательное if(N==1) PC = PC + k + 1 None 1/2 BRPL К Переход если отрицательное if(N==0) PC = PC + k + 1 None 1/2 BRGE К Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT К Переход если флаг полупереноса установлен if(S==1) PC = PC + k + 1 None 1/2 BRHS К Переход если флаг полупереноса очищен if(H==0) PC = PC + k + 1 None 1/2 BRTS К Переход если флаг тустановлен if(T==0) PC = PC + k + 1 None 1/2 BRVS К Переход если флаг Тустановлен if(V==1) PC = PC + k + 1 None 1/2 BRVC К Переход если флаг V очищен if(V==1) PC = PC + k + 1 None 1/2 BRIE К Переход если флаг V очищен if(I==0	<u>BRCS</u>	<u>k</u>	установлен	if(C==1) PC = PC + k + 1	None	1/2
BRLO к равно If(C==0) PC = PC + k + 1 None 1/2 BRMI к Переход если меньше if(C==1) PC = PC + k + 1 None 1/2 BRMI к Переход если отрицательное if(N==1) PC = PC + k + 1 None 1/2 BRPL к Переход если положительное if(N==0) PC = PC + k + 1 None 1/2 BRGE к Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRHS к Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRHC к Переход если флаг полупереноса очищен if(H==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRTC к Переход если флаг Т очищен if(V==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания if(I==0) PC = PC + k + 1	BRCC	<u>k</u>	очищен	if(C==0) PC = PC + k + 1	None	1/2
BRMI k Переход если отрицательное if(N==1) PC = PC + k + 1 None 1/2 BRPL k Переход если положительное if(N==0) PC = PC + k + 1 None 1/2 BRGE k Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT k Переход если флаг полупереноса установлен if(S==1) PC = PC + k + 1 None 1/2 BRHS k Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRTS k Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRVS k Переход если флаг V установлен if(V==0) PC = PC + k + 1 None 1/2 BRVC k Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRID k Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	<u>BRSH</u>	<u>k</u>		if(C==0) PC = PC + k + 1	None	1/2
BRPL k Переход если положительное if(N==0) PC = PC + k + 1 None 1/2 BRGE k Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT k Переход если меньше знаковое if(S==1) PC = PC + k + 1 None 1/2 BRHS k Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRTS k Переход если флаг т установлен if(T==0) PC = PC + k + 1 None 1/2 BRTC k Переход если флаг T очищен if(T==0) PC = PC + k + 1 None 1/2 BRVS k Переход если флаг V установлен if(V==0) PC = PC + k + 1 None 1/2 BRVC k Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRID k Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	<u>BRLO</u>	<u>k</u>	Переход если меньше	if(C==1) PC = PC + k + 1		
BRGE к Переход если больше или равно знаковое if(S==0) PC = PC + k + 1 None 1/2 BRLT к Переход если меньше знаковое if(S==1) PC = PC + k + 1 None 1/2 BRHS к Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг толупереноса очищен if(T==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг тустановлен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V установлен if(V==1) PC = PC + k + 1 None 1/2 BRVC к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2	BRMI	<u>k</u>	Переход если отрицательное	if(N==1) PC = PC + k + 1	None	
BRLT к равно знаковое II(S==0) PC = PC + k + 1 None 1/2 BRHS к Переход если меньше знаковое if(S==1) PC = PC + k + 1 None 1/2 BRHS к Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRTC к Переход если флаг Т очищен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V установлен if(V==1) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRPL	<u>k</u>		if(N==0) PC = PC + k + 1	None	1/2
BRHS к Переход если флаг полупереноса установлен if(H==1) PC = PC + k + 1 None 1/2 BRHC к Переход если флаг полупереноса очищен if(H==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRTC к Переход если флаг Т очищен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V установлен if(V==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRGE	_	•	if(S==0) PC = PC + k + 1	None	
BRHC к полупереноса установлен II(H==1) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг толупереноса очищен if(H==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Тустановлен if(T==1) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V установлен if(V==0) PC = PC + k + 1 None 1/2 BRVC к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRLT	<u>k</u>		if(S==1) PC = PC + k + 1	None	1/2
BRTS к полупереноса очищен II(H==0) PC = PC + k + 1 None 1/2 BRTS к Переход если флаг Т установлен if(T==1) PC = PC + k + 1 None 1/2 BRTC к Переход если флаг Т очищен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V установлен if(V==1) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	<u>BRHS</u>	<u>k</u>		if(H==1) PC = PC + k + 1	None	1/2
BRTS к установлен II(I==I) PC = PC + k + 1 None I/2 BRTC к Переход если флаг Т очищен if(T==0) PC = PC + k + 1 None 1/2 BRVS к Переход если флаг V очищен if(V==1) PC = PC + k + 1 None 1/2 BRVC к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRHC	<u>k</u>		if(H==0) PC = PC + k + 1	None	1/2
BRVS к Переход если флаг V установлен if(V==1) PC = PC + k + 1 None 1/2 BRVC к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRTS	<u>k</u>		if(T==1) PC = PC + k + 1	None	1/2
BRVC к Переход если флаг V очищен if(V==0) PC = PC + k + 1 None 1/2 BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRTC	<u>k</u>		if(T==0) PC = PC + k + 1	None	1/2
BRIE к Переход если прерывания разрешены if(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	BRVS	<u>k</u>		if(V==1) PC = PC + k + 1	None	1/2
BRID к разрешены II(I==1) PC = PC + k + 1 None 1/2 BRID к Переход если прерывания if(I==0) PC = PC + k + 1 None 1/2	<u>BRVC</u>	<u>k</u>	Переход если флаг V очищен	if(V==0) PC = PC + k + 1	None	1/2
BRID IK I ' ' ' I IIII==0) $PC = PC + K + 1 - I NODE = 11/2$	BRIE	<u>k</u>		if(I==1) PC = PC + k + 1	None	1/2
	BRID	<u>k</u>		if(I==0) PC = PC + k + 1	None	1/2

Таблица 1 – Команды

Название	Операнды	Описание	Операция	Флаги	Циклы
MOV	Rd,Rr	Копировать регистр	Rd = Rr	None	1
MOVW	Rd,Rr	Копировать пару регистров	Rd+1:Rd = Rr+1:Rr, r,d even	None	1

LD F	Rd,k Rd,X Rd,X+ Rd,-X Rd,-X Rd,Y+ Rd,-Y	Загрузить прямо Загрузить косвенно Загрузить косвенно с постинкрементом Загрузить косвенно с преддекрементом Загрузить косвенно Загрузить косвенно с постинкрементом Загрузить косвенно с	Rd = (k) Rd = (X) Rd = (X), $X=X+1$ X=X-1, $Rd = (X)Rd = (Y)Rd = (Y), Y=Y+1$	None None None None None	2* 2* 2* 2*
LD F LD F LD F LD F LD F LD F	Rd,X+ Rd,-X Rd,Y Rd,Y+	Загрузить косвенно с постинкрементом Загрузить косвенно с преддекрементом Загрузить косвенно Загрузить косвенно с постинкрементом	Rd = (X), X=X+1 X=X-1, Rd = (X) Rd = (Y)	None None	2*
LD F LD F LD F LDD F LDD F	Rd,-X Rd,Y Rd,Y+	постинкрементом Загрузить косвенно с преддекрементом Загрузить косвенно Загрузить косвенно с постинкрементом	X=X-1, Rd = (X) Rd = (Y)	None	2*
LD F LD F LDD F LDD F	Rd, Y Rd, Y+	преддекрементом Загрузить косвенно Загрузить косвенно с постинкрементом	Rd = (Y)		
LD F	Rd,Y+	Загрузить косвенно с постинкрементом	` '	None	0*
LD F	-	постинкрементом	Rd = (Y), Y=Y+1		2*
LDD F	Rd,-Y	Загрузить косвенно с	(.,,	None	2*
LD F		преддекрементом	Y=Y-1, Rd = (Y)	None	2*
	Rd,Y+q	Загрузить косвенно со смещением	Rd = (Y+q)	None	2*
	Rd,Z	Загрузить косвенно	Rd = (Z)	None	2*
LD F	Rd,Z+	Загрузить косвенно с постинкрементом	Rd = (Z), Z=Z+1	None	2*
LD F	Rd,-Z	Загрузить косвенно с преддекрементом	Z=Z-1, $Rd = (Z)$	None	2*
LDD F	Rd,Z+q	Загрузить косвенно со смещением	Rd = (Z+q)	None	2*
STS k	k,Rr	Сохранить прямо	(k) = Rr	None	2*
ST X	X,Rr	Сохранить косвенно	(X) = Rr	None	2*
ST X	X+,Rr	Сохранить косвенно с постинкрементом	(X) = Rr, X = X + 1	None	2*
ST -	-X,Rr	Сохранить косвенно с преддекрементом	X=X-1, (X)=Rr	None	2*
ST I	Y,Rr	Сохранить косвенно	(Y) = Rr	None	2*
ST Y	Y+,Rr	Сохранить косвенно с постинкрементом	(Y) = Rr, Y=Y+1	None	2
ST -	·Y,Rr	Сохранить косвенно с преддекрементом	Y=Y-1, (Y) = Rr	None	2
ST Y	Y+q,Rr	Сохранить косвенно со смещением	(Y+q) = Rr	None	2
ST Z	Z,Rr	Сохранить косвенно	(Z) = Rr	None	2
ST Z	Z+,Rr	Сохранить косвенно с постинкрементом	(Z) = Rr, Z=Z+1	None	2
ST -	-Z,Rr	Сохранить косвенно с преддекрементом	Z=Z-1, (Z) = Rr	None	2
ST Z	Z+q,Rr	Сохранить косвенно со смещением	(Z+q) = Rr	None	2
LPM /	Vone	Загрузить из памяти программ	R0 = (Z)	None	3
LPM I	Rd,Z	Загрузить из памяти программ	Rd = (Z)	None	3
LPM F	Rd,Z+	Загрузить из памяти программ с постинрементом	Rd = (Z), Z=Z+1	None	3
	None	Сохранить в память программ	(Z) = R1:R0	None	-
	Rd,P	Чтение регистра	Rd = P	None	1
	P,Rr	Запись регистра	P = Rr	None	1
	₹r ₹d	Поместить в стек Извлечь из стека	STACK = Rr Rd = STACK	None None	2