

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	T	H	S	V	N	Z	C	REG
Чтение/Запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 T как бит источник и бит назначения при операциях с битами. Командой BST бит регистра регистрового файла копируется в бит T, командой BLD бит T 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 C 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 I/O	$P = Rr$	None	1

*Операции с константами работают только с регистрами R16-R31

Таблица 3– Логические команды

Название	Операнды	Описание	Операция	Флаги	Циклы
AND	Rd, Rr	Логическое И	$Rd = Rd \text{ и } Rr$	Z,N,V,S	1
ANDI*	Rd, Rr	Логическое И с константой	$Rd = Rd \text{ и } K8$	Z,N,V,S	1
OR	Rd, Rr	Логическое ИЛИ	$Rd = Rd \text{ или } Rr$	Z,N,V,S	1
ORI*	Rd, K8	Логическое ИЛИ с константой	$Rd = Rd \text{ или } K8$	Z,N,V,S	1
COM	Rd, Rr	Инверсия	$Rd = 0xFF - Rd$	Z,N,V,S,C	1
EOR	Rd, K8	Логическое исключающее ИЛИ	$Rd = Rd \text{ исключающее или } Rr$	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	$Rd(3..0) = Rd(7..4)$, $Rd(7..4) = Rd(3..0)$	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 T	$T = Rr(b)$	T	1
BLD	Rd, b	Чтение из T в бит регистра	$Rd(b) = T$	Нет	1
SEC		Установка переноса	$C = 1$	C	1
CLC		Сброс переноса	$C = 0$	C	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 T в регистре SREG	$T = 1$	T	1
CLT		Сброс флага T в регистре SREG	$T = 0$	T	1
SEN		Установка флага H в регистре SREG	$H = 1$	H	1
CLH		Сброс флага H в регистре SREG	$H = 0$	H	1

Таблица 1 – Команды

Название	Операнды	Описание	Операция	Флаги	Циклы
TST	Rd	Проверка на ноль и отрицательное	$Rd = Rd \cdot Rd$	Z,C,N,V,S	1
RJMP	k	Относительный переход	$PC = PC + k + 1$	None	2
IJMP	None	Косвенный переход	$PC = Z$	None	2
JMP	k	Абсолютный переход	$PC = k$	None	3
RCALL	k	Относительный вызов подпрограммы	$STACK = PC + 1, PC = PC + k + 1$	None	3/4*
ICALL	None	Косвенный вызов подпрограммы	$STACK = PC + 1, PC = Z$	None	3/4*
CALL	k	Вызов подпрограммы	$STACK = PC + 2, PC = k$	None	4/5*
RET	None	Возврат из подпрограммы	$PC = STACK$	None	4/5*
RETI	None	Возврат из прерывания	$PC = STACK$	I	4/5*
CPSE	Rd,Rr	Сравнить, пропустить если равны	if ($Rd == Rr$) $PC = PC + 2$ or 3	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 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 I/O очищен	if($I/O(P,b) == 0$) $PC = PC + 2$ or 3	None	1/2/3
SBIS	P,b	Пропустить если бит в регистре установлен	if($I/O(P,b) == 1$) $PC = PC + 2$ or 3	None	1/2/3
BRBC	s,k	Переход если флаг очищен	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	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 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
BRHC	k	Переход если флаг полупереноса очищен	if($H == 0$) $PC = PC + k + 1$	None	1/2
BRTS	k	Переход если флаг T установлен	if($T == 1$) $PC = PC + k + 1$	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($V == 1$) $PC = PC + k + 1$	None	1/2
BRVC	k	Переход если флаг V очищен	if($V == 0$) $PC = PC + k + 1$	None	1/2
BRIE	k	Переход если прерывания разрешены	if($I == 1$) $PC = PC + k + 1$	None	1/2
BRID	k	Переход если прерывания запрещены	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	$Rd+1:Rd = Rr+1:Rr, r,d \text{ even}$	None	1

LDI	$Rd, K8$	Загрузить константу	$Rd = K$	None	1
LDS	Rd, k	Загрузить прямо	$Rd = (k)$	None	2*
LD	Rd, X	Загрузить косвенно	$Rd = (X)$	None	2*
LD	$Rd, X+$	Загрузить косвенно с постинкрементом	$Rd = (X), X=X+1$	None	2*
LD	$Rd, -X$	Загрузить косвенно с преддекрементом	$X=X-1, Rd = (X)$	None	2*
LD	Rd, Y	Загрузить косвенно	$Rd = (Y)$	None	2*
LD	$Rd, Y+$	Загрузить косвенно с постинкрементом	$Rd = (Y), Y=Y+1$	None	2*
LD	$Rd, -Y$	Загрузить косвенно с преддекрементом	$Y=Y-1, Rd = (Y)$	None	2*
LDD	$Rd, Y+q$	Загрузить косвенно со смещением	$Rd = (Y+q)$	None	2*
LD	Rd, Z	Загрузить косвенно	$Rd = (Z)$	None	2*
LD	$Rd, Z+$	Загрузить косвенно с постинкрементом	$Rd = (Z), Z=Z+1$	None	2*
LD	$Rd, -Z$	Загрузить косвенно с преддекрементом	$Z=Z-1, Rd = (Z)$	None	2*
LDD	$Rd, Z+q$	Загрузить косвенно со смещением	$Rd = (Z+q)$	None	2*
STS	k, Rr	Сохранить прямо	$(k) = Rr$	None	2*
ST	X, Rr	Сохранить косвенно	$(X) = Rr$	None	2*
ST	$X+, Rr$	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	Y, Rr	Сохранить косвенно	$(Y) = Rr$	None	2*
ST	$Y+, Rr$	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	$Y+q, Rr$	Сохранить косвенно со смещением	$(Y+q) = Rr$	None	2
ST	Z, Rr	Сохранить косвенно	$(Z) = Rr$	None	2
ST	$Z+, Rr$	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	$Z+q, Rr$	Сохранить косвенно со смещением	$(Z+q) = Rr$	None	2
LPM	None	Загрузить из памяти программ	$R0 = (Z)$	None	3
LPM	Rd, Z	Загрузить из памяти программ	$Rd = (Z)$	None	3
LPM	$Rd, Z+$	Загрузить из памяти программ с постинкрементом	$Rd = (Z), Z=Z+1$	None	3
SPM	None	Сохранить в память программ	$(Z) = R1:R0$	None	-
IN	Rd, P	Чтение регистра	$Rd = P$	None	1
OUT	P, Rr	Запись регистра	$P = Rr$	None	1
PUSH	Rr	Поместить в стек	$STACK = Rr$	None	2
POP	Rd	Извлечь из стека	$Rd = STACK$	None	2