

Система команд AVR.

Режимы адресации

Подразумеваемый операнд.

В команде не содержится явных указаний об операнде; в этом случае операнд подразумевается и фактически задаётся кодом операции команды.

Укороченная адресация.

В адресном поле командного слова содержатся только младшие разряды адресуемой ячейки.

Регистровая адресация.

Операнды и результат хранятся в рабочих регистрах центрального процессора.

Непосредственная адресация.

Операнд содержится непосредственно в команде. Используется для занесения в регистры констант.

Прямая адресация.

Адрес указывается непосредственно в виде некоторого значения.

Система команд AVR.

Режимы адресации (продолжение)

Косвенная адресация.

В коде команды указан регистр, который содержит адрес операнда.

Автоинкрементная (постинкрементная) адресация.

При автоинкрементной адресации сначала содержимое регистра используется как адрес операнда, а затем получает приращение, равное числу байт в элементе массива.

Автодекрементная (предекрементная) адресация.

Сначала содержимое указанного в команде регистра уменьшается на число байт в элементе массива, а затем используется как адрес операнда.

Индексация.

Адрес операнда вычисляется как сумма содержимого индексного регистра и смещения-константы.

Система команд 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: $N \oplus V$, 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

a: 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

Система команд AVR.

Арифметические команды

Мнем-оника	Операнды	Описание	Операция	Флаги	К-во циклов
ADD	Rd,Rr	Add without Carry - Сложить без переноса	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd,Rr	Add with Carry - Сложить с переносом	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd,K	Add Immediate to Word- Сложить непосредственное значение со словом	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V	2
SUB	Rd,Rr	Subtract without Carry - Вычесть без переноса	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd,K	Subtract Immediate - Вычесть непосредственное значение	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd,Rr	Subtract with Carry - Вычесть с переносом	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd,K	Subtract Immediate with Carry - Вычесть непосредственное значение с переносом	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd,K	Subtract Immediate from Word- Вычесть непосредственное значение из слова	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V	2

Система команд AVR.

Логические команды

Мнем-оника	Операнды	Описание	Операция	Флаги	К-во циклов
AND	Rd,Rr	Logical AND- Выполнить логическое AND	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd,K	Logical AND with Immediate- Выполнить логическое AND с непосредственным значением	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd,Rr	Logical OR - Выполнить логическое OR	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd,K	Logical OR with Immediate - Выполнить логическое OR с непосредственным значением	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd,Rr	Exclusive OR - Выполнить исключающее OR	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement - Выполнить дополнение до единицы	$Rd \leftarrow \sim Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement - Выполнить дополнение до двух	$Rd \leftarrow \sim Rd + 1$	Z,C,N,V,H	1

Система команд AVR.

Команды работы с битами и проверки

Мнем-оника	Операнды	Описание	Операция	Флаги	К-во циклов
SBR	Rd,K	Set Bits in Register- Установить биты в регистре	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bits in Register - Очистить биты в регистре	$Rd \leftarrow Rd \bullet (\text{SFF} - K)$	Z,N,V	1
INC	Rd	Increment - Инкрементировать	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement - Декрементировать	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus - Проверить на ноль или минус	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register- Очистить регистр	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set all bits in Register- Установить все биты регистра	$Rd \leftarrow \text{SFF}$	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

Система команд AVR.

Команды перехода

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
RJP	k	Relative Jump- Перейти относительно	$PC \leftarrow PC + k + 1$	Нет	2
IJP		Indirect Jump- Перейти косвенно	$PC \leftarrow Z$	Нет	2
JMP	k	Jump- Перейти	$PC \leftarrow k$	Нет	3
RCALL	k	Relative Call to Subroutine - Вызвать подпрограмму относительно	$PC \leftarrow PC + k + 1$	Нет	3
ICALL		Indirect Call to Subroutine - Вызвать подпрограмму косвенно	$PC \leftarrow Z$	Нет	3
CALL	k	Long Call to a Subroutine - Выполнить длинный вызов подпрограммы	$PC \leftarrow k$	Нет	4
RET		Return from Subroutine - Вернуться из подпрограммы	$PC \leftarrow STACK$	Нет	4
RETI		Return from Interrupt- Вернуться из прерывания	$PC \leftarrow STACK$	I	4

Система команд AVR.

Команды перехода (продолжение 1)

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
CPSE	Rd,Rr	Compare Skip if Equal- Сравнить и пропустить если равно	If Rd = Rr then $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$,	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 $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$	Нет	1/2/3
SBRSC	Rr,b	Skip if Bit in Register is Set- Пропустить если бит в регистре Установлен	If Rr(b) = 1 then $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$	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 $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$	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 $PC \leftarrow PC + 2 \text{ (or } 3\text{)}$	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 $PC \leftarrow PC + k + 1$	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 $PC \leftarrow PC + k + 1$	Нет	1/2
BREQ	k	Branch if Equal- Перейти если равно	If Rd = Rr (Z = 1) then $PC \leftarrow PC + k + 1$	Нет	1/2
BRNE	k	Branch if Not Equal- Перейти если не равно	If Rd \neq Rr (Z = 0) then $PC \leftarrow PC + k + 1$	Нет	1/2
BRCS	k	Branch if Carry Set- Перейти если флаг переноса Установлен	If C = 1 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRCC	k	Branch if Carry Cleared- Перейти если флаг переноса очищен	If C = 0 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRSH	k	Branch if Same or Higher (Unsigned) - Перейти если равно или больше (без знака)	If Rd \geq Rr (C = 0) then $PC \leftarrow PC + k + 1$	Нет	1/2

Система команд AVR.

Команды перехода (продолжение 2)

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
BRLO	k	Branch if Lower (Unsigned) - Перейти если меньше (без знака)	If $R_d < R_r$ ($C = 1$) then $PC \leftarrow PC + k + 1$	Нет	1/2
BRMI	k	Branch if Minus - Перейти если минус	If $N = 1$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRPL	k	Branch if Plus- Перейти если плюс	If $N = 0$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRGE	k	Branch if Greater or Equal (Signed)- Перейти если больше или равно (с Учетом знака)	If $R_d \geq R_r$ ($N \oplus V = 0$) then $PC \leftarrow PC + k + 1$	Нет	1/2
BRLT	k	Branch if Less Than (Signed) - Перейти если меньше чем (со знаком)	If $R_d < R_r$ ($N \oplus V = 1$) then $PC \leftarrow PC + k + 1$	Нет	1/2
BRHS	k	Branch if Half Carry Flag is Set- Перейти если флаг полупереноса Установлен	If $H = 1$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRHC	k	Branch if Half Carry Flag is Cleared - Перейти если флаг полупереноса очищен	If $H = 0$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRTS	k	Branch if T Flag is Set- Перейти если флаг T Установлен	If $T = 1$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRTC	k	Branch if T Flag is Cleared- Перейти если флаг T очищен	If $T = 0$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRVS	k	Branch if Overflow Set - Перейти если переполнение Установлено	If $V = 1$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRVC	k	Branch if Overflow Cleared- Перейти если переполнение очищено	If $V = 0$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRIE	k	Branch if Global Interrupt is Enabled - Перейти если глобальное прерывание разрешено	If $I = 1$ then $PC \leftarrow PC + k + 1$	Нет	1/2
BRID	k	Branch if Global Interrupt is Disabled - Перейти если глобальное прерывание запрещено	If $I = 0$ then $PC \leftarrow PC + k + 1$	Нет	1/2

Система команд AVR.

Команды пересылки

Мне- мони- ка	Опер- анды	Описание	Операция	Фла- ги	К-во цик- лов
ELPM		Extended Load Program Memory - Расширенная загрузка памяти программ.	$R_0 \leftarrow [Z + RAMPZ]$	Нет	3
MOV	Rd,Rr	Copy Register- Копировать регистр	$Rd \leftarrow Rr$	Нет	1
LDI	Rd,K	Load Immediate - Загрузить непосредственное значение	$Rd \leftarrow K$	Нет	1
LDS	Rd,k	Load Direct from RAM - Загрузить непосредственно из ОЗУ	$Rd \leftarrow [k]$	Нет	3
LD	Rd,X	LD - Load Indirect - Загрузить косвенно	$Rd \leftarrow (X)$	Нет	2
LD	Rd,X+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	$Rd \leftarrow (X),$ $X \leftarrow X + 1$	Нет	2
LD	Rd,X-	Load Indirect and Pre-Decrement - Загрузить косвенно декрементировав предварительно	$X \leftarrow X - 1,$ $Rd \leftarrow (X)$	Нет	2
LD	Rd,Y	Load Indirect - Загрузить косвенно	$Rd \leftarrow (Y)$	Нет	2
LD	Rd,Y+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	$Rd \leftarrow (Y),$ $Y \leftarrow Y + 1$	Нет	2
LD	Rd,Y-	Load Indirect and Pre-Decrement - Загрузить косвенно декрементировав предварительно	$Y \leftarrow Y - 1,$ $Rd \leftarrow (Y)$	Нет	2
LDD	Rd,Y+a	Load Indirect with Displacement - Загрузить косвенно со смещением.	$Rd \leftarrow (Y + a)$	Нет	2
LD	Rd,Z	Load Indirect - Загрузить косвенно	$Rd \leftarrow (Z)$	Нет	2
LD	Rd,Z+	Load Indirect and Post-Increment - Загрузить косвенно инкрементировав впоследствии	$Rd \leftarrow (Z),$ $Z \leftarrow Z + 1$	Нет	2
LD	Rd,Z-	Load Indirect and Pre-Decrement - Загрузить косвенно декрементировав предварительно	$Z \leftarrow Z - 1,$ $Rd \leftarrow (Z)$	Нет	2
LDD	Rd,Z+a	Load Indirect with Displacement - Загрузить косвенно со смещением.	$Rd \leftarrow (Z + a)$	Нет	2

Система команд AVR.

Команды пересылки (продолжение)

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

Система команд AVR.

Команды сдвига и тестирования

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
LSL	Rd	Logical Shift Left- Логически сдвинуть влево	$Rd(n+1) \leftarrow Rd(n),$ $Rd(0) \leftarrow 0, C \leftarrow Rd(7)$	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right- Логически сдвинуть вправо	$Rd(n) \leftarrow Rd(n+1),$ $Rd(7) \leftarrow 0, C \leftarrow Rd(0)$	Z,C,N,V	1
ROL		Rotate Left through Carry- Сдвинуть влево через перенос	$Rd(0) \leftarrow C,$ $Rd(n+1) \leftarrow Rd(n),$ $C \leftarrow Rd(7)$	Z,C,N,V,H	1
ROR	Rd	Rotate Right through Carry- Сдвинуть вправо через перенос	$Rd(7) \leftarrow C,$ $Rd(n) \leftarrow Rd(n+1),$ $C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right- Арифметически сдвинуть вправо	$Rd(n) \leftarrow Rd(n+1),$ $n=0...6$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles - Поменять нибблы местами	$Rd(3...0) \longleftrightarrow Rd(7...4)$	Нет	1
BSET	s	Flag Set -Установить флаг	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear -Очистить флаг	$SREG(s) \leftarrow 0$	SREG(s)	1
SBI	P,b	Set bit to I/O Register - Установить бит в регистр I/O	$I/O(P,b) \leftarrow 1$	Нет	2

Система команд AVR.

Команды сдвига и тестирования (прод. 1)

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
CBI	P, b	Clear Bit in I/O Register - Очистить бит в регистре I/O	$I/O[P, b] \leftarrow 0$	Нет	2
BST	Rd, b	Bit Store from Register to T - Перенести бит из регистра во флаг T	$T \leftarrow Rd[b]$	T	1
BLD	Rd, b	Bit Load from T to Register - Загрузить T флаг в бит регистра	$Rd[b] \leftarrow T$	Нет	1
SEC		Set Carry Flag - Установить флаг переноса	$C \leftarrow 1$	C	1
CLC		Clear Carry Flag - Очистить флаг переноса	$C \leftarrow 0$	C	1
SEN		Set Negative Flag - Установить флаг отрицательного значения	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag - Очистить флаг отрицательного значения	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag - Установить флаг нулевого значения	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag - Очистить флаг нулевого значения	$Z \leftarrow 0$	Z	1
SEI		Set Global Interrupt Flag - Установить флаг глобального прерывания	$I \leftarrow 1$	I	1
CLI		Clear Global Interrupt Flag - Очистить флаг глобального прерывания	$I \leftarrow 0$	I	1

Система команд AVR.

Команды сдвига и тестирования (прод. 2)

Мне- мони- ка	Опе- ран- ды	Описание	Операция	Фла- ги	К-во цик- лов
SES		Set Signed Flag- Установить флаг знака	$S \leftarrow 1$	S	1
CLS		Clear Signed Flag - Очистить флаг знака	$S \leftarrow 0$	S	1
SEV		Set Overflow Flag- Установить флаг переполнения	$V \leftarrow 1$	V	1
CLV		Clear Overflow Flag - Очистить флаг переполнения	$V \leftarrow 0$	V	1
SET		Set T Flag- Установить флаг T	$T \leftarrow 1$	T	1
CLT		Clear T Flag - Очистить флаг T	$T \leftarrow 0$	T	1
SEN		Set Half Carry Flag- Установить флаг полу переноса	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag -Очистить флаг полу переноса	$H \leftarrow 0$	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