

Сводная таблица команд AVR Tiny&Mega микроконтроллеров с аналогиями MCS51

AND Rd, Rr	ANDI Rd, #k	EOR Rd, Rr XOR Rd=Rd⊕Rr	OR Rd, Rr	ORI Rd, #k команды SUBI, SBCI с отрицательным k -1+ -127 равноценны ADI, ADCI с k=1+127		COM Rd CPL	NEG Rd Rd=00-Rd	CLR Rd Rd=Rd⊕Rd	SER Rd Rd=FF	TST Rd Rd•Rd	SWAP Rd (7:4)↔(3:0)				
ADD Rd, Rr	ADC Rd, Rr	ADIW Rd+1:Rd, k d=24,26,28,30 0≤k≤63	SUB Rd, Rr	SUBI Rd, #k	SBC Rd, Rr Rd=Rd-Rr-C	SUBB Rd, #k	SBCI Rd, #k	SBIW Rd+1:Rd, k d=24,26,28,30 0≤k≤63	DEC Rd	INC Rd	ASR Rd Rd←(7:0)→0	LSL Rd Rd←(7:0)←0	LSR Rd Rd←(7:0)→0	ROL Rd Rd←(7:0)←(0)→(7)	ROR Rd Rd←(7:0)→(0)←(7)
MUL Rd, Rr R1:R0=Rd×Rr	MULS Rd, Rr	MULSU Rd, Rr R1:R0=Rd×Rr	FMUL Rd, Rr	FMULS Rd, Rr	FMULSU Rd, Rr R1:R0=(Rd×Rr)≪1										
CBR Rd, #k Rd=Rd•(FF-k)	SBR Rd, #k Rd=Rd+k	CBI #P, b P.b=0	SBI #P, b P.b=1	BCLR s sreg.s=0	BSET s sreg.s=1	BLD Rd, b Rd.b←T	BST Rr, b T←Rr.b		SBRC Rr, b пропуск при Rr.b=0	SBRS Rr, b пропуск при Rr.b=1	SBIC #P, b пропуск при P.b=0	SBIS #P, b пропуск при P.b=1		BRBC s, re переход при sreg.s=0	BRBS s, re переход при sreg.s=1
CLC C=0	SEC C=1	CLN N=0	SEN N=1	CLZ Z=0	SEZ Z=1	CLI I=0	SEI I=1	CLS S=0	SES S=1	CLV V=0	SEV V=1	CLT T=0	SET T=1	CLH H=0	SEH H=1
MOV Rd, Rr Rd←Rr	MOVW Rd+1:Rd, Rr+1:Rr Rd+1:Rd←Rr+1:Rr	LDI Rd, #k Rd←k	LD Rd, X Rd←(X)	LD Rd, X+ Rd←(X) x=x+1	LD Rd, -X Rd←(X) x=x-1	LD Rd, Y Rd←(Y)	LD Rd, Y+ Rd←(Y) y=y+1	LD Rd, -Y Rd←(Y) y=y-1	LDD Rd, Y+rel Rd←(Y+rel)	LD Rd, Z Rd←(Z)	LD Rd, Z+ Rd←(Z) z=z+1	LD Rd, -Z Rd←(Z) z=z-1	LDD Rd, Z+rel Rd←(Z+rel)	LDS Rd, #adr Rd←(adr) 0≤adr≤65535	
			ST X, Rd (X)←Rd	ST X+, Rd (X)←Rd x=x+1	ST -X, Rd (X)←Rd x=x-1	ST Y, Rd (Y)←Rd	ST Y+, Rd (Y)←Rd y=y+1	ST -Y, Rd (Y)←Rd y=y-1	STD Y+rel, Rd (Y+rel)←Rd	ST Z, Rd (Z)←Rd	ST Z+, Rd (Z)←Rd z=z+1	ST -Z, Rd (Z)←Rd z=z-1	STD Z+rel, Rd (Z+rel)←Rd	STS #adr, Rd (adr)←Rd 0≤adr≤65535	
LPM R0+(Z) Rd←(Z)	LPM Rd, Z Rd←(Z)	LPM Rd, Z+ Rd←(Z) z=z+1	ELPM R0+(RAMPZ:Z) Rd←(RAMPZ:Z)	ELPM Rd, Z Rd←(RAMPZ:Z)	ELPM Rd, Z+ Rd←(RAMPZ:Z) z=z+1	SPM см. прогп. FLASH			IN Rd, #Pخ Rd←Pخ	OUT #Pخ, Rr Pخ←Rr		PUSH Rr (stack)←Rr SP=SP-1	POP Rd Rd←(stack) SP=SP+1		
RJMP k -2048≤k≤2047	IJMP PC+Z JMP @PTR	JMP k 0≤k≤4Mb	RCALL k -2048≤k≤2047	ICALL CALL @PTR	CALL k 0≤k≤64Kb	RET	RETI		NOP	WDR	SLEEP см. выбр. реж. sleep		BREAK см. реж. внутр. отладки		
BRID re переход при I=0	BRIE re переход при I=1		CP Rd, Rr Rd-Rr	CPC Rd, Rr Rd-Rr-C	CPI Rd, #k Rd-k	CPSE Rd, Rr пропуск при Rd=Rr	устанавливает флаги согласно результату содержимое регистров Rd и Rr не изменяют				SPM Z+	EIJMP PC(15:0)+Z(15:0) PC(21:16)+EIND	EICALL PC(15:0)+Z(15:0) PC(21:16)+EIND	DES k 0x00≤k≤0x0F	
BRCS re jmp C=1	BRCC re jmp C=0	BREQ re jmp Rd=Rr Z=1	BRNE re jmp Rd≠Rr Z=0	BRSH re jmp Rd>Rr C=0	BRL0 re jmp Rd<Rr C=1	BRMI re jmp N=1	BRPL re jmp N=0	BRGE re jmp Rd>Rr NOV=0	BRLT re jmp Rd<Rr NOV=1	BRHS re jmp H=1	BRHC re jmp H=0	BRTS re jmp T=1	BRTC re jmp T=0	BRVS re jmp V=1	BRVC re jmp V=0

по умолчанию данные в десятичном формате

- AND
- +OR
- ⊕XOR
- NOT
- (CMP, CPL)

0≤Rd/r≤31
для команд с зеленой рамкой
16≤Rd/r≤31

0≤P≤31 (0x1F)
0≤Pخ≤63 (0x3F)
0≤b≤7
0≤s≤7
0≤rel≤63
-64≤re≤63

0≤k≤255
если не указано ниже

стек растёт вниз
CALL
SP 1-байт
h-байт

N-S

#P - знак # ставится, если порт обозначен как код (двоичный/десятичный) если порт обозначается как имя знак # не ставится

в командах, содержащих Rd+1:Rd, k или Rd+1:Rd, Rr+1:Rr допускается сокращенная форма с младшим регистром пары Rd, k или Rd, Rr

CBR в маске бит подлежащий сбросу должен быть равен 1 и наоборот

SBR в маске бит подлежащий установке должен быть равен 1

R<<n сдвинуть n на m раз

функции битового аккумулятора MCS51 (флаг C) выполняет бит T регистра SREG

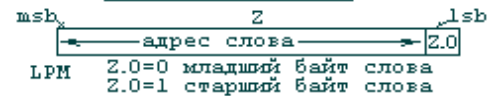
■ - синим вертикально отмечены аналоги из MCS51/INTEL8080/280
rstep-переход на +1команду

табличный декодер (Zh: Z1+rel)×2

строка данных читается побайтово т.е Zh: Z1+1 на каждый следующий байт строки

обмен с EEPROM данных производится с помощью соотв. регистров PC# (регистровый файл)

I	T	H	S	V	N	Z	C	SREG
7							0	аналогии от PSW MCS51
		AC	S	OV		Z	C	аналогии от 280
		H	S	PV	N	Z	C	



■ - применение команд SPM, SLEEP и BREAK требуется согласовывать с конкретным datasheet