

Команды микроконтроллера семейства МК51

Мнемоника	Описание команды	Байты	Циклы
Группа команд пересылки данных			
MOV A,Rn	Пересылка в аккумулятор из РОН (n=0-7)	1	1
MOV A,ad	Пересылка в А прямоадресуемого байта	2	1
MOV A,@Ri	Пересылка в аккумулятор байта РПД (i=1,2)	1	1
MOV A,#d	Загрузка в аккумулятор константы	2	1
MOV Rn,A	Пересылка в регистр из аккумулятора	1	1
MOV Rn,ad	Пересылка в РОН прямоадресуемого байта	2	2
MOV Rn,#d	Пересылка в регистр константы	2	1
MOV ad,A	Пересылка по прямому адресу А	2	1
MOV ad,Rn	Пересылка по прямому адресу регистра	2	2
MOV add,ads	Пересылка прямоадресуемый байт по прямому адресу	3	2
MOV ad,@Ri	Пересылка байта РПД по прямому адресу	2	2
MOV ad,#d	Пересылка константы по прямому адресу	3	2
MOV @Ri,A	Пересылка байта в РПД из А	1	1
MOV @Ri,ad	Пересылка прямоадресуемого байта в РПД	2	2
MOV @Ri,#d	Пересылка константы в РПД	2	1
MOV DPTR,#d16	Загрузка указателя данных	3	2
MOVC A,@A+DPTR	Пересылка в А байта из памяти программ	1	2
MOVC A,@A+PC	Пересылка в А байта из памяти программ	1	2
MOVX A,@Ri	Пересылка в А байта из ВПД	1	2
MOVX A,@DPTR	Пересылка в А байта из расширенной ВПД	1	2
MOVX @Ri,A	Пересылка в ВПД из аккумулятора	1	2
MOVX @DPTR,A	Пересылка в расширенную ВПД из А	1	2
PUSH ad	Загрузка в стек	2	2
POP ad	Извлечение из стека	2	2
XCH A,Rn	Обмен аккумулятора с регистром	1	1
XCH A,ad	Обмен А с прямоадресуемым байтом	2	1
XCH A,@Ri	Обмен А с байтом из РПД	1	1
XCHD A,@Ri	Обмен младшими тетрадами А и байта РПД	1	1
Мнемоника			
Группа команд арифметических операций			
ADD A,Rn	Сложение А с регистром (n=0-7)	1	1
ADD A,ad	Сложение А с прямоадресуемым байтом	2	1
ADD A,@Ri	Сложение А с байтом из РПД (i=0,1)	1	1
ADD A,#d	Сложение А с константой	2	1
ADDC A,Rn	Сложение А с регистром и переносом	1	1
ADDC A,ad	Сложение А с байтом и переносом	2	1
ADDC A,@Ri	Сложение А с байтом РПД и переносом	1	1
ADDC A,#d	Сложение А с константой и переносом	2	1
DA A	Десятичная коррекция аккумулятора	1	1
SUBB A,Rn	Вычитание из А регистра и заёма	1	1
SUBB A,ad	Вычитание из А байта и заёма	2	1
SUBB A,@Ri	Вычитание из А байта РПД и заёма	1	1

SUBB A,#d	Вычитание из А константы и заёма	2	1
INC A	Инкремент аккумулятора	1	1
INC Rn	Инкремент регистра	1	1
INC ad	Инкремент прямоадресуемого байта	2	1
INC @Ri	Инкремент байта РПД	1	1
INC DPTR	Инкремент указателя данных	1	2
DEC A	Декремент аккумулятора	1	1
DEC Rn	Декремент регистра	1	1
DEC ad	Декремент прямоадресуемого байта	2	1
DEC @Ri	Декремент байта РПД	1	1
MUL AB	Умножение аккумулятора на регистр В	1	4
DIV AB	Деление аккумулятора на регистр В	1	4
Мнемоника			
Группа команд логических операций			
ANL A,Rn	Логическое И регистра и А	1	1
ANL A,ad	Логическое И А и прямоадресуемого байта	2	1
ANL A,@Ri	Логическое И байта РПД и А	1	1
ANL A,#d	Логическое И константы и А	2	1
ANL ad,A	Логическое И прямоадресуемого байта и А	2	1
ANL ad,#d	Логическое И байта и константы	3	2
ORL A,Rn	Логическое ИЛИ регистра и А	1	1
ORL A,ad	Логическое ИЛИ А и байта	2	1
ORL A,@Ri	Логическое ИЛИ байта РПД и А	1	1
ORL A,#d	Логическое ИЛИ константы и А	2	1
ORL ad,A	Логическое ИЛИ байта и А	2	1
ORL ad,#d	Логическое ИЛИ байта и константы	3	2
XRL A,Rn	Исключающие ИЛИ регистра и А	1	1
XRL A,ad	Исключающие ИЛИ А и байта	2	1
XRL A,@Ri	Исключающие ИЛИ байта РПД и А	1	1
XRL A,#d	Исключающие ИЛИ константы и А	2	1
XRL ad,A	Исключающие ИЛИ байта и А	2	1
XRL ad,#d	Исключающие ИЛИ байта и константы	3	2
CLR A	Сброс аккумулятора	1	1
CPL A	Инверсия аккумулятора	1	1
RL A	Циклический сдвиг аккумулятора влево	1	1
RLC A	Сдвиг аккумулятора влево через перенос	1	1
RR A	Циклический сдвиг аккумулятора вправо	1	1
RRC A	Сдвиг аккумулятора вправо через перенос	1	1
SWAP A	Обмен местами тетрад в аккумуляторе	1	1
Мнемоника			
Группа команд операций с битами			
CLR C	Сброс переноса	1	1
CLR bit	Сброс бита	2	1
SETB C	Установка переноса	1	1
SETB bit	Установка бита	2	1
CPL C	Инверсия переноса	1	1
CPL bit	Инверсия бита	2	1
ANL C,bit	Логическое И бита и переноса	2	2
ANL C,/bit	Логическое И инверсии бита и переноса	2	2

ORL C,bit	Логическое ИЛИ бита и переноса	2	2
ORL C,/bit	Логическое ИЛИ инверсии бита и переноса	2	2
MOV C,bit	Пересылка бита в перенос	2	1
MOV bit,C	Пересылка переноса в бит	2	2
Мнемоника	Группа команд передачи управления	Байты	Циклы
LJMP ad16	Длинный переход в полном объеме ПП	3	2
AJMP ad11	Абсолютный переход внутри стр. в 2Кбайта	2	2
SJMP \$+2+rel	Короткий относительный переход	2	2
JMP @A+DPTR	Косвенный относительный переход	1	2
JZ \$+2+rel	Переход, если А равен нулю	2	2
JNZ \$+2+rel	Переход, если А не равен нулю	2	2
JC \$+2+rel	Переход, если перенос равен единице	2	2
JNC \$+2+rel	Переход, если перенос равен нулю	2	2
JB \$+3+rel	Переход, если бит равен единице	3	2
JNB \$+3+rel	Переход, если бит равен нулю	3	2
JBC \$+3+rel	Переход, если бит установлен, с последующим сбросом бита	3	2
DJNZ Rn, \$+2+rel	Декремент РОН и переход, если не нуль	2	2
DJNZ ad, \$+3+rel	Декремент прямоадресуемого байта и переход, если не нуль	3	2
CJNE A,ad,\$+3+rel	Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	3	2
CJNE A,#d,\$+3+rel	Сравнение аккумулятора с константой и переход, если не равно	3	2
CJNE Rn,#d,\$+3+rel	Сравнение регистра с константой и переход, если не равно	3	2
CJNE @Ri,#d,\$+3+rel	Сравнение байта в РПД с константой и переход, если не равно	3	2
LCALL ad16	Длинный вызов подпрограммы	3	2
ACALL ad11	Абсолютный вызов подпрограммы в пределах страницы в 2Кбайта	2	2
RET	Возврат из подпрограммы	1	2
RETI	Возврат из подпрограммы обработки прерывания	1	2
NOP	Холостая команда	1	1

Система команд K1816BE51 в алфавитном порядке

Мнемокод	КОП	Ц	Мнемокод	КОП	Ц
ACALL 0xxH	11	2	ANL A,@R1	57	1
ACALL 1xxH	31	2	ANL A,#d	54	1
ACALL 2xxH	51	2	ANL ad,A	52	2
ACALL 3xxH	71	2	ANL ad,#d	53	2
ACALL 4xxH	91	2	ANL C,bit	82	2
ACALL 5xxH	B1	2	ANL C,/bit	B0	2
ACALL 6xxH	D1	2	AJMP 0xxH	01	2
ACALL 7xxH	F1	2	AJMP 1xxH	21	2
ADD A,ad	25	1	AJMP 2xxH	41	2
ADD A,R0	28	1	AJMP 3xxH	61	2
ADD A,R1	29	1	AJMP 4xxH	81	2
ADD A,R2	2A	1	AJMP 5xxH	A1	2
ADD A,R3	2B	1	AJMP 6xxH	C1	2
ADD A,R4	2C	1	AJMP 7xxH	E1	2
ADD A,R5	2D	1	CJNE A,ad,adr	B5	2
ADD A,R6	2E	1	CJNE A,#d,adr	B4	2
ADD A,R7	2F	1	CJNE R0,#d,adr	B8	2
ADD A,@R0	26	1	CJNE R1,#d,adr	B9	2
ADD A,@R1	27	1	CJNE R2,#d,adr	BA	2
ADD A,#d	24	1	CJNE R3,#d,adr	BB	2
ADDC A,ad	35	1	CJNE R4,#d,adr	BC	2
ADDC A,R0	38	1	CJNE R5,#d,adr	BD	2
ADDC A,R1	39	1	CJNE R6,#d,adr	BE	2
ADDC A,R2	3A	1	CJNE R7,#d,adr	BF	2
ADDC A,R3	3B	1	CJNE @R0,#d,adr	B6	2
ADDC A,R4	3C	1	CJNE @R1,#d,adr	B7	2
ADDC A,R5	3D	1	CLR A	E4	1
ADDC A,R6	3E	1	CLR bit	C2	1
ADDC A,R7	3F	1	CLR C	C3	1
ADDC A,@R0	36	1	CPL A	F4	1
ADDC A,@R1	37	1	CPL bit	B2	1
ADDC A,#d	34	1	CPL C	B3	1
ANL A,ad	55	1	DEC A	14	1
ANL A,R0	58	1	DEC ad	15	1
ANL A,R1	59	1	DEC R0	18	1
ANL A,R2	5A	1	DEC R1	19	1
ANL A,R3	5B	1	DEC R2	1A	1
ANL A,R4	5C	1	DEC R3	1B	1
ANL A,R5	5D	1	DEC R4	1C	1
ANL A,R6	5E	1	DEC R5	1D	1
ANL A,R7	5F	1	DEC R6	1E	1
ANL A,@R0	56	1	DEC R7	1F	1

Мнемокод	КОП	Ц	Мнемокод	КОП	Ц
DEC @R0	16	1	MOV A,@R1	E7	1
DEC @R1	17	1	MOV A,#d	74	1
DIV AB	84	4	MOV ad,A	F5	1
DJNZ ad,adr	D5	2	MOV ad,R0	88	2
DJNZ R0,adr	D8	2	MOV ad,R1	89	2
DJNZ R1,adr	D9	2	MOV ad,R2	8A	2
DJNZ R2,adr	DA	2	MOV ad,R3	8B	2
DJNZ R3,adr	DB	2	MOV ad,R4	8C	2
DJNZ R4,adr	DC	2	MOV ad,R5	8D	2
DJNZ R5,adr	DD	2	MOV ad,R6	8E	2
DJNZ R6,adr	DE	2	MOV ad,R7	8F	2
DJNZ R7,adr	DF	2	MOV ad,@R0	86	2
INC A	04	1	MOV ad,@R1	87	2
INC ad	05	1	MOV ad,#d	75	2
INC DPTR	A3	2	MOV add,ads	85	2
INC R0	08	1	MOV bit,C	92	2
INC R1	09	1	MOV C,bit	A2	1
INC R2	0A	1	MOV DPTR,#d16	90	2
INC R3	0B	1	MOV R0,ad	A8	2
INC R4	0C	1	MOV R0,A	F8	1
INC R5	0D	1	MOV R0,#d	78	1
INC R6	0E	1	MOV R1,A	F9	1
INC R7	0F	1	MOV R1,ad	A9	2
INC @R0	06	1	MOV R1,#d	79	1
INC @R1	07	1	MOV R2,A	FA	1
DA A	D4	1	MOV R2,ad	AA	2
JB bit,adr	20	2	MOV R2,#d	7A	1
JBC bit,adr	10	2	MOV R3,A	FB	1
JC adr	40	2	MOV R3,ad	AB	2
JMP @A+DPTR	73	2	MOV R3,#d	7B	1
JNB bit,adr	30	2	MOV R4,A	FC	1
JNC adr	50	2	MOV R4,ad	AC	2
JNZ adr	70	2	MOV R4,#d	7C	1
JZ adr	60	2	MOV R5,A	FD	1
LCALL ad16	12	2	MOV R5,ad	AD	2
LJMP ad16	02	2	MOV R5,#d	7D	1
MOV A,ad	E5	1	MOV R6,A	FE	1
MOV A,R0	E8	1	MOV R6,ad	AE	2
MOV A,R1	E9	1	MOV R6,#d	7E	1
MOV A,R2	EA	1	MOV R7,A	FF	1
MOV A,R3	EB	1	MOV R7,ad	AF	2
MOV A,R4	EC	1	MOV R7,#d	7F	1
MOV A,R5	ED	1	MOV @R0,A	F6	1
MOV A,R6	EE	1	MOV @R0,ad	A6	2
MOV A,R7	EF	1	MOV @R0,#d	76	1
MOV A,@R0	E6	1	MOV @R1,A	F7	1

Мнемокод	КОП	Ц	Мнемокод	КОП	Ц
MOV @R1,ad	A7	2	SUBB A,R0	98	1
MOV @R1,#d	77	1	SUBB A,R1	99	1
MOVC A, @A+DPTR	93	2	SUBB A,R2	9A	1
MOVC A, @A+PC	83	2	SUBB A,R3	9B	1
MOVX A, @DPTR	E0	2	SUBB A,R4	9C	1
MOVX A, @R0	E2	2	SUBB A,R5	9D	1
MOVX A, @R1	E3	2	SUBB A,R6	9E	1
MOVX @DPTR,A	F0	2	SUBB A,R7	9F	1
MOVX @R0,A	F2	2	SUBB A,@R0	96	1
MOVX @R1,A	F3	2	SUBB A,@R1	97	1
MUL AB	A4	4	SUBB A,#d	94	1
NOP	00	1	XCH A,ad	C5	1
ORL A,ad	45	1	XCH A,R0	C8	1
ORL A,R0	48	1	XCH A,R1	C9	1
ORL A,R1	49	1	XCH A,R2	CA	1
ORL A,R2	4A	1	XCH A,R3	CB	1
ORL A,R3	4B	1	XCH A,R4	CC	1
ORL A,R4	4C	1	XCH A,R5	CD	1
ORL A,R5	4D	1	XCH A,R6	CE	1
ORL A,R6	4E	1	XCH A,R7	CF	1
ORL A,R7	4F	1	XCH A,@R0	C6	1
ORL A,@R0	46	1	XCH A,@R1	C7	1
ORL A,@R1	47	1	XCHD A,@R0	D6	1
ORL A,#d	44	1	XCHD A,@R1	D7	1
ORL ad,A	42	1	XRL A,ad	65	1
ORL ad,#d	43	2	XRL A,R0	68	1
ORL C,bit	72	2	XRL A,R1	69	1
ORL C,/bit	A0	2	XRL A,R2	6A	1
POP ad	D0	2	XRL A,R3	6B	1
PUSH ad	C0	2	XRL A,R4	6C	1
RET	22	2	XRL A,R5	6D	1
RETI	32	2	XRL A,R6	6E	1
RL A	23	1	XRL A,R7	6F	1
RLC A	33	1	XRL A,@R0	66	1
RR A	03	1	XRL A,@R1	67	1
RRC A	13	1	XRL A,#d	64	1
SETB bit	D2	1	XRL ad,A	62	1
SETB C	D3	1	XRL ad,#d	63	2
SJMP adr	80	2			
SWAP A	C4	1			
SUBB A,ad	95	1			

Один машинный цикл при частоте кварца 12МГц равен 1мкс.

Обозначение операндов:

ad – прямой 8-битовый адрес байта резидентной памяти данных, порта или регистра специальных функций;

add – прямой 8-битовый адрес приемника данных;

ads – прямой 8-битовый адрес источника данных;

adr – адрес передачи управления, определяемый как (PC)+rel;

ad16 – прямой 16-битовый адрес передачи управления;

bit – прямой 8-битовый адрес бита;

#d – данные 8-разрядные;

rel – 8-битовый байт смещения со знаком.