OP CODE DESCRIPTIONS BY FUNCTION

Arithmetic

addition

increment (none affected*)		b/c**
INC	@ R0	1/1
INC	@ R1	1/1
INC	a	1/1
INC	direct	2/1
INC	dptr	1/2
INC	R0	1/1
INC	R1	1/1
INC	R2	1/1
INC	R3	1/1
INC	R4	1/1
INC	R5	1/1
INC	R6	1/1
INC	R7	1/1

add (CY, AC, OV*)		b/c**
ADD	a, @ R0	1/1
ADD	a, @ R1	1/1
ADD	a, #data	2/1
ADD	a, direct	2/1
ADD	a, R0	1/1
ADD	a, R1	1/1
ADD	a, R2	1/1
ADD	a, R3	1/1
ADD	a, R4	1/1
ADD	a, R5	1/1
ADD	a, R6	1/1
ADD	a, R7	1/1

add with carry (CY, AC, OV*)		b/c**
ADDC	a, @ R0	1/1
ADDC	a, @ R1	1/1
ADDC	a, #data	2/1
ADDC	a, direct	2/1
ADDC	a, R0	1/1
ADDC	a, R1	1/1
ADDC	a, R2	1/1
ADDC	a, R3	1/1
ADDC	a, R4	1/1
ADDC	a, R5	1/1
ADDC	a, R6	1/1
ADDC	a, R7	1/1

decimal adjust for BCD add (CY*)		b/c**
DA	a	1/1

subtraction

decrement (None*)		b/c**
DEC	@ R0	1/1
DEC	@ R1	1/1
DEC	a	1/1
DEC	direct	2/1
DEC	R0	1/1
DEC	R1	1/1
DEC	R2	1/1
DEC	R3	1/1
DEC	R4	1/1
DEC	R5	1/1
DEC	R6	1/1
DEC	R7	1/1

subtract with borrow (CY, AC, OV*)		b/c**
SUBB	a, @ R0	1/1
SUBB	a, @ R1	1/1
SUBB	a, #data	2/1
SUBB	a, direct	2/1
SUBB	a, R0	1/1
SUBB	a, R1	1/1
SUBB	a, R2	1/1
SUBB	a, R3	1/1
SUBB	a, R4	1/1
SUBB	a, R5	1/1
SUBB	a, R6	1/1
SUBB	a, R7	1/1

multiplication (CY, OV*)		b/c**
MUL	ab	1/4

division (CY,OV*)		b/c**
DIV	ab	1/4

Logical

bit ops set/CLR/complement (as targeted*)		b/c**
CLR	bit	2/1
CLR	c	1/1
SETB	bit	2/1
SETB	c	1/1
CPL	bit	2/1
CPL	c	1/1

byte ops

ACC CLR / complement (none*)		b/c**
CLR	a	1/1
CPL	a	1/1

rotates (none*)		b/c**
RL	a	1/1
RR	a	1/1

rotates thru CY (CY*)		b/c**
RLC	a	1/1
RRC	a	1/1

AND (none*)		b/c**
ANL	a, #data	2/1
ANL	a, @ R0	1/1
ANL	a, @ R1	1/1
ANL	a, direct	2/1
ANL	direct, a	2/1
ANL	a, R0	1/1
ANL	a, R1	1/1
ANL	a, R2	1/1
ANL	a, R3	1/1
ANL	a, R4	1/1
ANL	a, R5	1/1
ANL	a, R6	1/1
ANL	a, R7	1/1
ANL	direct, # data	3/2
ANL	c,/bit	2/2
ANL	c, bit	2/2

OR (none*)		b/c**
ORL	a, # data	2/1
ORL	a, @ R0	1/1
ORL	a, @ R1	1/1
ORL	a, direct	2/1
ORL	direct, a	2/1
ORL	a, R0	1/1
ORL	a, R1	1/1
ORL	a, R2	1/1
ORL	a, R3	1/1
ORL	a, R4	1/1
ORL	a, R5	1/1
ORL	a, R6	1/1
ORL	a, R7	1/1
ORL	direct, # data	3/2
ORL	c,/bit	2/2
ORL	c, bit	2/2

XOR (none*)		b/c**
XRL	a, # data	2/1
XRL	a, @ R0	1/1
XRL	a, @ R1	1/1
XRL	a, direct	2/1
XRL	direct, a	2/1
XRL	a, R0	1/1
XRL	a, R1	1/1
XRL	a, R2	1/1
XRL	a, R3	1/1
XRL	a, R4	1/1
XRL	a, R5	1/1
XRL	a, R6	1/1
XRL	a, R7	1/1
XRL	direct, # data	3/2

* Flags affected by instruction

**b/c = bytes/cycles

JMP : Jump

CLR : Clear

ACC : Accumulator

Control Transfer

unconditional

jumps (none*)		b/c**
SJMP	rel	2/2
LJMP	addr16	3/2
AJMP	addr11	2/2
AJMP		2/2
AJMP		2/2
AJMP		2/2
AJMP		2/2
AJMP		2/2
AJMP		2/2
AJMP		2/2
calls (none*)		**b/c****
LCALL	addr16	3/2
ACALL	addr11	2/2
ACALL		2/2
ACALL		2/2
ACALL		2/2
ACALL		2/2
ACALL		2/2
ACALL		2/2
ACALL		2/2
returns (none*)		**b/c****
RET		1/2
RETI		1/2
table JMP (none*)		**b/c****
JMP	@a+dptr	1/2

conditional

JMP conditoinal on bit (none*)		b/c**
JB	bit,rel	3/2
JC	rel	2/2
JNC	rel	2/2
JNB	bit, rel	3/2
JNZ	rel	2/2
JZ	rel	2/2
JMP on bit; CLR bit (as targeted*)		**b/c****
JBC	bit, rel	3/2
compare, JMP not equal (flags : CY)		**b/c****
CJNE	@R0, #data, rel	3/2
CJNE	@R1, #data, rel	3/2
CJNE	a, #data, rel	3/2
CJNE	a, direct, rel	3/2
CJNE	R0, #data, rel	3/2
CJNE	R1, #data, rel	3/2
CJNE	R2, #data, rel	3/2
CJNE	R3, #data, rel	3/2
CJNE	R4, #data, rel	3/2
CJNE	R5, #data, rel	3/2
CJNE	R6, #data, rel	3/2
CJNE	R7, #data, rel	3/2

decrement,JMP non-zero (none)		b/c**
DJNZ	direct, rel	3/2
DJNZ	R0, rel	2/2
DJNZ	R1, rel	2/2
DJNZ	R2, rel	2/2
DJNZ	R3, rel	2/2
DJNZ	R4, rel	2/2
DJNZ	R5, rel	2/2
DJNZ	R6, rel	2/2
DJNZ	R7, rel	2/2

Data Transfer

bit <--> carry (as targeted*)		b/c**
MOV	bit,c	2/2
MOV	c,bit	2/1
immediate data (none*)		**b/c****
MOV	a,#data	2/1
MOV	@R0,#data	2/1
MOV	@R1,#data	2/1
MOV	direct,#data	3/2
MOV	dptr,#data16	3/2
MOV	R0,#data	2/1
MOV	R1,#data	2/1
MOV	R2,#data	2/1
MOV	R3,#data	2/1
MOV	R4,#data	2/1
MOV	R5,#data	2/1
MOV	R6,#data	2/1
MOV	R7,#data	2/1

ACC <--> REG, direct (none*)		b/c**
MOV	a,R0	1/1
MOV	a,R1	1/1
MOV	a,R2	1/1
MOV	a,R3	1/1
MOV	a,R4	1/1
MOV	a,R5	1/1
MOV	a,R6	1/1
MOV	a,R7	1/1
MOV	R0,a	1/1
MOV	R1,a	1/1
MOV	R2,a	1/1
MOV	R3,a	1/1
MOV	R4,a	1/1
MOV	R5,a	1/1
MOV	R6,a	1/1
MOV	R7,a	1/1

ACC <--> REG, indirect (none*)		b/c**
MOV	@R0,a	1/1
MOV	@R1,a	1/1
MOV	a,@R0	1/1
MOV	a,@R1	1/1

RAM <--> REG, indirect (flags:none)		b/c**
MOV	@R0,direct	2/2
MOV	@R1,direct	2/2
MOV	direct,@R0	2/2
MOV	direct,@R1	2/2

ACC <--> RAM, direct (flags:none)		b/c**
MOV	a,direct	2/1
MOV	direct,a	2/1

RAM to RAM (flags : none)		b/c**
MOV	direct,direct	3/2

REG <--> data MEM (flags : none)		b/c**
MOV	R0,direct	2/2
MOV	R1,direct	2/2
MOV	R2,direct	2/2
MOV	R3,direct	2/2
MOV	R4,direct	2/2
MOV	R5,direct	2/2
MOV	R6,direct	2/2
MOV	R7,direct	2/2
MOV	direct,R0	2/2
MOV	direct,R1	2/2
MOV	direct,R2	2/2
MOV	direct,R3	2/2
MOV	direct,R4	2/2
MOV	direct,R5	2/2
MOV	direct,R6	2/2
MOV	direct,R7	2/2

exchange ACC with RAM (none)		b/c**
XCH	a,@R0	1/1
XCH	a,@R1	1/1
XCH	a,direct	2/1
XCH	a,R0	1/1
XCH	a,R1	1/1
XCH	a,R2	1/1
XCH	a,R3	1/1
XCH	a,R4	1/1
XCH	a,R5	1/1
XCH	a,R6	1/1
XCH	a,R7	1/1

REG : Register MEM : Memory EXT : External <--> : To/from 1 cyc = 12 oscillator periods

ACC <--> REG, indirect (none*)		b/c**
MOV	@R0,a	1/1
MOV	@R1,a	1/1
MOV	a,@R0	1/1
MOV	a,@R1	1/1

RAM <--> REG, indirect (flags:none)		b/c**
MOV	@R0,direct	2/2
MOV	@R1,direct	2/2
MOV	direct,@R0	2/2
MOV	direct,@R1	2/2

ACC <--> RAM, direct (flags:none)		b/c**
MOV	a,direct	2/1
MOV	direct,a	2/1

RAM to RAM (flags : none)		b/c**
MOV	direct,direct	3/2

REG <--> data MEM (flags : none)		b/c**
MOV	R0,direct	2/2
MOV	R1,direct	2/2
MOV	R2,direct	2/2
MOV	R3,direct	2/2
MOV	R4,direct	2/2
MOV	R5,direct	2/2
MOV	R6,direct	2/2
MOV	R7,direct	2/2
MOV	direct,R0	2/2
MOV	direct,R1	2/2
MOV	direct,R2	2/2
MOV	direct,R3	2/2
MOV	direct,R4	2/2
MOV	direct,R5	2/2
MOV	direct,R6	2/2
MOV	direct,R7	2/2

exchange ACC with RAM (none)		b/c**
XCH	a,@R0	1/1
XCH	a,@R1	1/1
XCH	a,direct	2/1
XCH	a,R0	1/1
XCH	a,R1	1/1
XCH	a,R2	1/1
XCH	a,R3	1/1
XCH	a,R4	1/1
XCH	a,R5	1/1
XCH	a,R6	1/1
XCH	a,R7	1/1

program MEM table lookup (none)		b/c**
MOVC	a,@a+dptr	1/2
MOVC	a,@a+pc	1/2

ACC <--> EXT data MEM (Flags : none)		b/c**
MOVX	@R0,a	1/2
MOVX	@R1,a	1/2
MOVX	a,@R0	1/2
MOVX	a,@R1	1/2
MOVX	@dtpr,a	1/2
MOVX	a,@dptr	1/2

Misc stack operations (flags : none)		b/c**
POP	direct	2/2
PUSH	direct	2/2

nibble swaps (flags : none)		b/c**
SWAP	a	1/1
XCHD	a,@R0	1/1
XCHD	a,@R1	1/1

no-operation (flags : none)		b/c**
NOP		1/1

reserved (flags : none)	b/c**
(reserved)	-/-

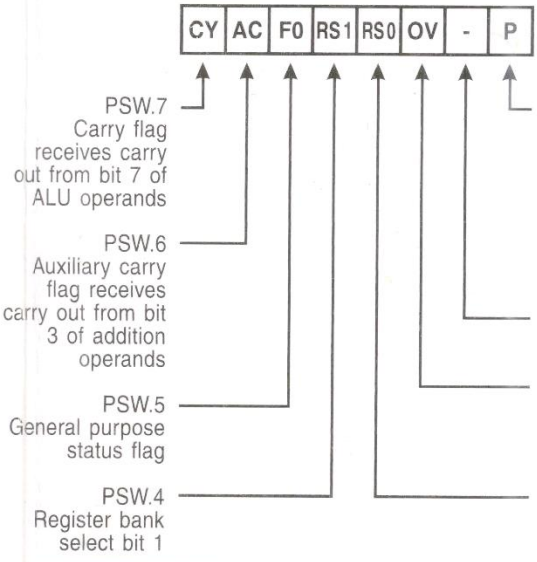
Key :

addr11 = lower 11 bits of 16-bit addresses; upper 5 bits come from current PC (after incrementing twice)
addr16 = code address in program memory
bit = bit address (0-FF)
/bit = bit address (0-FF) (bit is complemented before use)
direct = data address in internal RAM memory
im8 = immediate byte value (0-FF)
im16 = immediate word value (0-FFFF)
rel = signed 1 byte program memory displacement

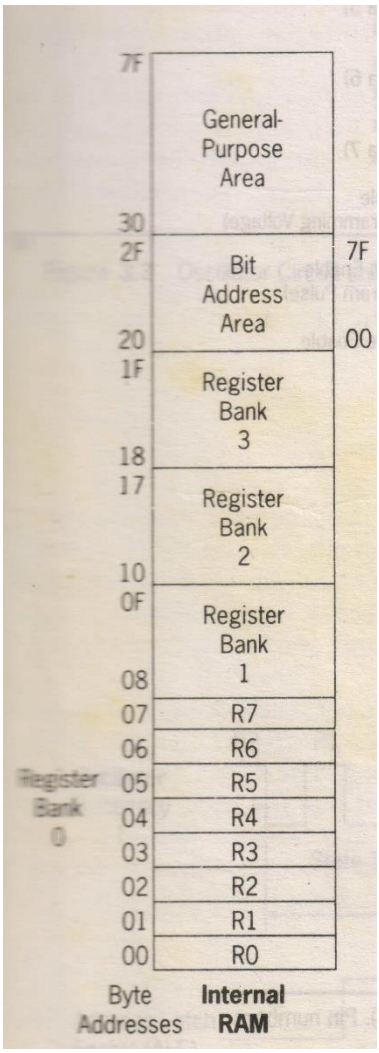
Notes :

1. The flags effects notation describes PSW bits changed by the CPU. Data moves and pops to PSW will directly affect the flags.

Program Status Word (PSW) Bit Definitions



1 cyc = 12 oscillator periods



2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00

Bit Addressed RAM