

Команди передачі управління

JMP a	aaa00100 aaaaaaaa	Безумовний перехід $PC[10..0] := a[10..0]$; $PC[11] := MB$
JMPP @A	10110011	Безумовний перехід в межах поточної сторінці; $PC[7..0] := (A)$
JC a	11110110 aaaaaaaa	Перехід, якщо $C=1$ $PC[7..0] := a$ інакше $PC := PC + 2$
JNC a	11100110 aaaaaaaa	Перехід, якщо $C=0$
DJNZ Rr, a	11101rrr aaaaaaaa	Декремент вмісту регістру і перехід, якщо вміст регістру не дорівнює нулю
JZ a	11000110 aaaaaaaa	Перехід, якщо вміст A дорівнює нулю
JNZ a	10010110 aaaaaaaa	Перехід, якщо вміст A не дорівнює нулю
1	2	3
JF0 a	10110110 aaaaaaaa	Перехід, якщо $F0=1$
JF1 a	01110110 aaaaaaaa	Перехід, якщо $F1=1$
JT0 a	00110110 aaaaaaaa	Перехід, якщо $T0=1$
JNT0 a	00100110 aaaaaaaa	Перехід, якщо $T0=0$
JT1 a	01010110 aaaaaaaa	Перехід, якщо $T1=1$
JNT1 a	01000110 aaaaaaaa	Перехід, якщо $T1=0$
JTF a	00010110 aaaaaaaa	Перехід, якщо $TF=1$
JNI a	10000110 aaaaaaaa	Перехід, якщо $INT=0$
JBb a	bbb10010 aaaaaaaa	Перехід, якщо розряд Bb акумулятора встановлений в одиницю
CALL a	aaa10100 aaaaaaaa	Виклик підпрограми; $SP := SP + 1$; $(SP) := PSW[7..4]$; $PC[11] := MB$; $PC[10..0] := a[10..0]$
RET	10000011	Повернення із підпрограми; $SP := SP - 1$; $PC := (SP[11..0])$
1	2	3
RETR	10010011	Повернення із підпрограми з встановленням стану; $SP := SP - 1$; $PC := SP[11..0]$; $PSW[7..4] := (SP[15..12])$

<i>Команди звернення до акумулятора</i>		
CLR A	00100111	Встановлення вмісту акумулятора в нуль $A := 0$
CPL A	00110011	Інвертування вмісту A; $A := NOT A$
INC A	00010111	Інкремент вмісту A; $A := A + 1$
DEC A	00000111	Декремент вмісту A; $A := A - 1$
RR A	01110111	Циклічний зсув вмісту A вправо; $A[7] := A[0]$; $A[i] := A[i + 1]$; $i = \overline{6, 0}$
RL A	11110011	Циклічний зсув вмісту A вліво; $A[0] := A[7]$; $A[i] := A[i + 1] := A[i]$; $i = \overline{6, 0}$
RRC A	01100111	Циклічний зсув вмісту A з бітом переносу вправо; $A[7] := C$; $C := A[0]$; $A[i] := A[i + 1]$; $i = \overline{6, 0}$
RLC A	11110111	Циклічний зсув вмісту A з бітом переносу вліво; $A[0] := C$; $C := A[1]$; $A[i + 1] := A[i]$; $i = \overline{6, 0}$
SWAP A	01000111	Обмін тетрадами A; $A[7..0] \leftrightarrow A[3..0]$
DA A	01010111	Десяткова корекція вмісту A

1	2	3
MOV A, Rr ; $r = (7 - 0)$	11111rrr	Пересилка вмісту регістру в A; $A := Rr$
MOV Rr, A ; $r = (7 - 0)$	10101rrr	Пересилка вмісту A в регістр; $Rr := A$
XCH A, Rr ; $r = (7 - 0)$	00101rrr	Обмін вмісту A і регістру; $A \leftrightarrow Rr$
ANL A, Rr ; $r = (7 - 0)$	01011rrr	Логічне І вмісту A і регістру; $A := A AND Rr$
ORL A, Rr ; $r = (7 - 0)$	01001rrr	Логічне АБО вмісту A і регістру; $A := A OR Rr$
XRL A, Rr ; $r = (7 - 0)$	11011rrr	Виключне АБО вмісту A і регістру; $A := A XOR Rr$
ADD A, Rr ; $r = (7 - 0)$	01101rrr	Сума вмісту A і регістру; $A := A + Rr$
ADDC A, Rr ; $r = (7 - 0)$	01111rrr	Сума вмісту A, регістру і переносу C; $A := A + Rr + C$
DEC Rr ; $r = (7 - 0)$	11001rrr	Декремент вмісту регістру; $Rr := Rr - 1$
INC Rr ; $r = (7 - 0)$	00011rrr	Інкремент вмісту регістру; $Rr := Rr + 1$

<i>Команди звернення до внутрішньої пам'яті даних</i>		
MOV A, @Rr ; $r = 0, 1$	1111000r	Пересилка із внутрішньої пам'яті даних в A; $A := (Rr)$
MOV @Rr, A ; $r = 0, 1$	1010000r	Пересилка вмісту A до внутрішньої пам'яті даних; $(Rr) := A$
XCH A, @Rr ; $r = 0, 1$	0010000r	Обмін вмістом A і комірки внутрішньої пам'яті даних; $A \leftrightarrow Rr$
XCHD A, @Rr ; $r = 0, 1$	0011000r	Обмін молодшими тетрадами A і комірки внутрішньої пам'яті даних; $A[3..0] \leftrightarrow (Rr)[3..0]$
ANL A, @Rr ; $r = 0, 1$	0101000r	Логічне І вмісту A і комірки внутрішньої пам'яті даних; $A := A AND (Rr)$

1	2	3
ORL A, @Rr ; $r = 0, 1$	0100000r	Логічне АБО вмісту A і комірки резидентної пам'яті даних; $A := A OR (Rr)$
XRL A, @Rr ; $r = 0, 1$	1101000r	Виключення АБО вмісту A і комірки резидентної пам'яті даних; $A := A XOR (Rr)$
ADD A, @Rr ; $r = 0, 1$	0110000r	Сума вмісту A і комірки резидентної пам'яті даних; $A := A + (Rr)$
ADDC A, @Rr ; $r = 0, 1$	0111000r	Сума вмісту A, комірки резидентної пам'яті даних і переносу C; $A := A + (Rr) + C$
INC @Rr ; $r = 0, 1$	0001000r	Інкремент комірки резидентної пам'яті даних; $(Rr) := (Rr) + 1$

<i>Команди роботи з зовнішньою пам'яттю даних</i>		
MOVX A, @Rr ; $r = 0, 1$	1000000r	Пересилка із ЗПД в A; $A := (Rr)$
MOVX @Rr, A ; $r = 0, 1$	1001000r	Пересилка вмісту A до ЗПД; $(Rr) := A$

Команди звернення до пам'яті програми

MOV Rr, #d ; r = (7-0)	10111rrr ddddddddd	Пересилка безпосереднього операнда до регістру; $(Rr) := d$
MOV A, #d	00100011 ddddddddd	Пересилка безпосередньої адреси до A $A := d$
MOV @Rr, #d ; r = 0,1	1011000r ddddddddd	Пересилка безпосереднього операнда до внутрішньої пам'яті даних $(Rr) := d$

Продовження табл. 3.3

1	2	3
ANL A, #d	01010011 ddddddddd	Логічне І вмісту A з безпосереднім операндом; $A := A \text{ AND } d$
ORL A, #d	01000011 ddddddddd	Логічне АБО вмісту A з безпосереднім операндом; $A := A \text{ OR } d$
XRL A, #d	11010011 ddddddddd	Виключне АБО вмісту A з безпосереднім операндом; $A := A \text{ XOR } d$
ADD A, #d	00000011 ddddddddd	Сума вмісту A та безпосереднього операнду; $A := A + d$
ADDC A, #d	00010011 ddddddddd	Сума вмісту A, безпосереднього операнду та переносу C; $A := A + d + C$
MOVP A, @A	10100011	Пересилка даних із поточної сторінки пам'яті програм до A; $A := (PC[11..8]..A)$
MOVP3 A, @A	11100011	Пересилка даних із сторінки 3 пам'яті програм до A; $A := (0011..A)$

Команди звернення до регістру PSW

MOV PSW, A	11010111	Пересилка вмісту A до регістру PSW; $PSW := A$
MOV A, PSW	11000111	Пересилка вмісту регістру PSW до A; $A := PSW$
MOV A, T	01000010	Пересилка вмісту TCNT в A; $A := TCNT$
MOV T, A	01100010	Пересилка вмісту A в TCNT; $TCNT := A$

<u>Команди встановлення ознак</u>		
CLR C	10010111	Встановлення в нуль ознаки C; $C := 0$
CPL C	10100111	Інвертування ознаки C; $C := NOT C$
CLR F0	10000101	Встановлення в нуль ознаки F0; $F0 := 0$
CLR F1	10100101	Встановлення в нуль ознаки F1; $F1 := 0$
CPL F0	10010101	Інвертування ознаки F0; $F0 := NOT F0$
CPL F1	10110101	Інвертування ознаки F1; $F1 := NOT F1$
<u>Команди звернення до портів P1 і P2</u>		
ANL Pp, #d ; p=1,2	100110pp ddddddddd	Логічне І порту P1(P2) з безпосереднім операндом; $Pp := Pp AND d$
ORL Pp, #d ; p=1,2	100010pp ddddddddd	Логічне АБО порту P1(P2) з безпосереднім операндом; $Pp := Pp OR d$
IN A, Pp	000010pp	Введення даних із порту P1(P2) в A; $A := Pp$
OUTL Pp, A	001110pp	Виведення вмісту A в порт P1(P2) $Pp := A$
<u>Команди звернення до портів P4, P5, P6, P7</u>		
ANLD Pp, A ; p=(7-4)	100111pp	Логічне І порту P4(P5, P6, P7) з A; $Pp := Pp AND A[3..0]$
ORLD Pp, A ; p=(7-4)	100011pp	Логічне АБО порту P4(P5, P6, P7) з A; $Pp := Pp OR A[3..0]$
MOVD A, Pp ; p=(7-4)	000011pp	Ввід із порту P4(P5, P6, P7) в A; $A[7..4] := 0$; $A[3..0] := Pp$
MOVD Pp, A ; p=(7-4)	001111pp	Вивід молодшої тетради із A в порт P4(P5, P6, P7); $Pp := A[3..0]$
<u>Команди звернення до порту BUS</u>		
ANL BUS, #d	10011000 ddddddddd	Логічне І порту BUS з безпосереднім операндом; $BUS := BUS AND d$
ORL BUS, #d	10001000 ddddddddd	Логічне АБО порту BUS з безпосереднім операндом; $BUS := BUS OR d$
INS A, BUS	00001000	Введення даних із порту BUS в A; $A := BUS$
OUTL BUS, A	00000010	Виведення вмісту A в порт BUS; $BUS := A$
<u>Команди управління режимами роботи</u>		
ENTO CLK	01110101	Дозвіл видачі імпульсів синхрон. на TO
SEL MBO	11100101	Вибір нульового банку пам'яті програм; $MB := 0$
SEL MB1	11110101	Вибір першого банку пам'яті програм; $MB := 1$
SEL RBO	11000101	Вибір нульового банку регістрів пам'яті даних; $RB := 0$
SEL RB1	11010101	Вибір першого банку регістрів пам'яті даних; $RB := 1$
NOP	00000000	Немає операції
EN I	00000101	Дозвіл зовнішніх переривань
DIS I	00010101	Заборона зовнішніх переривань
EN TCNTI	00100101	Дозвіл переривань від таймера/лічильника
DIS TCNTI	00110101	Заборона переривань від таймера/лічильника
STRT T	01010101	Запускання таймера/лічильника в режимі таймера
STRT CNT	01000101	Запускання таймера/лічильника в режимі лічильника
STOP TCNT	01100101	Зупинка таймера/лічильника