

2507/302
MICROCONTROLLER TECHNOLOGY
June/July 2023
Time: 3 hours



THE KENYA NATIONAL EXAMINATIONS COUNCIL
DIPLOMA IN AERONAUTICAL ENGINEERING
(AVIONICS OPTION)

MODULE III

MICROCONTROLLER TECHNOLOGY

3 hours

INSTRUCTIONS TO CANDIDATES

You should have the following for this examination:

Answer booklet;

Non-programmable scientific calculator;

Intel 8051 Instruction set.

This paper consists of EIGHT questions.

Answer any FIVE of the EIGHT questions in the answer booklet provided.

All questions carry equal marks.

Maximum marks for each part of a question are as indicated.

Candidates should answer the questions in English.

This paper consists of 11 printed pages.

Candidates should check the question paper to ascertain that all the pages are printed as indicated and that no questions are missing.

1. (a) Perform the following number system conversions:

(i) 1110101_2 to decimal;

(ii) 237_8 to decimal;

(iii) A_3FE_{16} to binary.

(6 marks)

(b) Explain the following Intel 8051 microcontroller addressing modes citing one example in each case:

(i) indirect;

(ii) indexed.

(4 marks)

(c) Explain what each of the following Intel 8051 microcontroller instructions accomplishes:

(i) `MOV A, 65H;`

(ii) `ANL A, #F2H`

(4 marks)

(d) **Table 1** shows a program segment of Intel 8051 microcontroller.

Table 1

INSTRUCTION	T-states
ORG 0000H	
MOV A, 60H	1
ADD A, 61 H	1
DAA	3
MOV 52H, A	1
MOV A, #00H	1
ADDC A, # 00H	2
MOV 53H, A	1
END	

The oscillator frequency of the microcontroller is 12 MHz. Determine the:

(i) execution time of the program;

(ii) length of the program in bytes.

(6 marks)

2. (a) (i) State the function of each of the following Intel 8051 microcontroller CPU registers:

- I. Accumulator;
- II. B register.

(ii) Draw the structure of intel 8051 microcontroller flag register. (6 marks)

(b) Table 2 shows two intel microcontroller series and their features. Complete the table. (4 marks)

Table 2

Series	I/O pins	chip pins	RAM size (bytes)	ROM size (bytes)
8048	27	—	64	—
8051	—	40	—	4k

(c) Write an intel 8051 microcontroller assembly language program to perform the following sequentially:

- Load data 2CH to register A.
 - Logically OR the content of A with data 50 H.
 - Logically AND content of A with content of memory location pointed by register R_0 .
 - Store results in memory location 45 H.
- (4 marks)

(d) Table 3 shows an Intel 8051 program listing:

- (i) draws its trace table;
 - (ii) determine the size of program in bytes.
- (6 marks)

Table 3

No	Instructions
1	MOV R5, #25H
2	MOV A, #55H
3	ADD A, R5

3. (a) With the aid of diagrams, describe each of the following microcontroller input/output devices:
- (i) single pole double throw (SPDT) relay;
 - (ii) capacitive proximity switch. (8 marks)
- (b) An aircraft warning light indicator ON condition is represented by the following Boolean expression.

$$F = A.B + \overline{A.B}$$

Draw a programmable logic controller (PLC) ladder diagram to implement the expression. (4 marks)

- (c) **Figure 1** shows a PLC ladder logic diagram program.

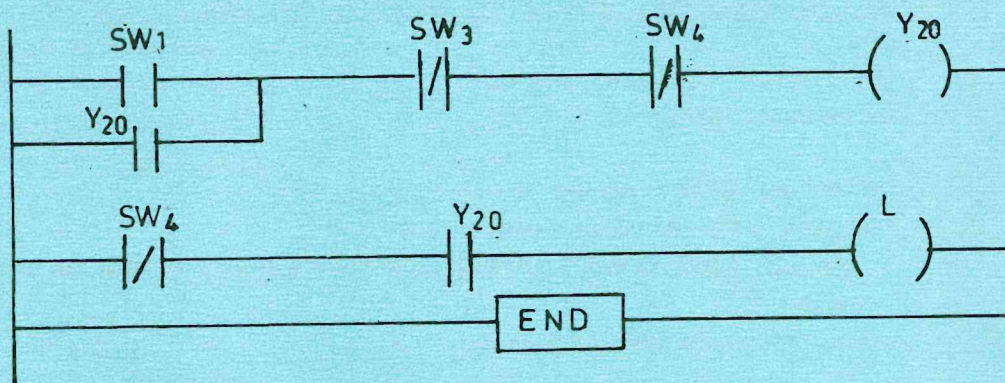


Fig.1

- (i) Write the Boolean equations for the output Y20 and L;
 - (ii) Write the program instruction listing. (8 marks)
4. (a) Describe each of the following with respect to supervisory control and data acquisition (SCADA) system.
- (i) Remote terminal unit (RTU);
 - (ii) central host. (4 marks)
- (b) Describe the three generations of a SCADA system. (6 marks)
- (c) With the aid of a block schematic diagram, describe the operation of a successive approximation analog-to-digital converter (ADC). (7 marks)
- (d) An 8 bit analog-to-digital converter has a reference voltage of 7 V. The analogue input is 2.5 V. Determine the binary output of the converter. (3 marks)

5. (a) Define the following with respect to robot:
- (i) work space;
 - (ii) work envelope. (2 marks)
- (b) State **three** types of proximity sensors used by robots in aircraft manufacturing industry. (3 marks)
- (c) Describe **three** drive systems used in landing gears. (6 marks)
- (d) Describe **three** robot programming methods. (9 marks)
6. (a) State **three** functions of interfacing with respect to microcontrollers. (3 marks)
- (b) (i) Define each of the following with respect to Digital to Analog converters (DAC):
- I. resolution;
 - II. linearity.
- (ii) A 5-bit DAC has a full scale output of 10 V. Determine the:
- I. resolution in volts;
 - II. output voltage when the input is $(10101)_2$. (8 marks)
- (c) Distinguish between maskable and non-maskable interrupts. (2 marks)
- (d) A VDU, keyboard and printer are assigned descending interrupt priorities 1, 2 and 3 respectively in a microcontroller based system.
- (i) Draw a block diagram of these devices daisy chain hardware connection.
 - (ii) Describe the connection in (d)(i). (7 marks)

7. (a) (i) Describe the operation of a microprogrammed control unit. (7 marks)
- (ii) State **two** merits of the design in (a)(i). (3 marks)
- (b) (i) Define a port with respect to I/O interfacing. (3 marks)
- (ii) State **one** merit and one demerit of parallel ports. (5 marks)
- (c) **Figure 2** shows a circuit diagram of a pulse width measurement timer used in microcontrollers. Describe its operation. (5 marks)

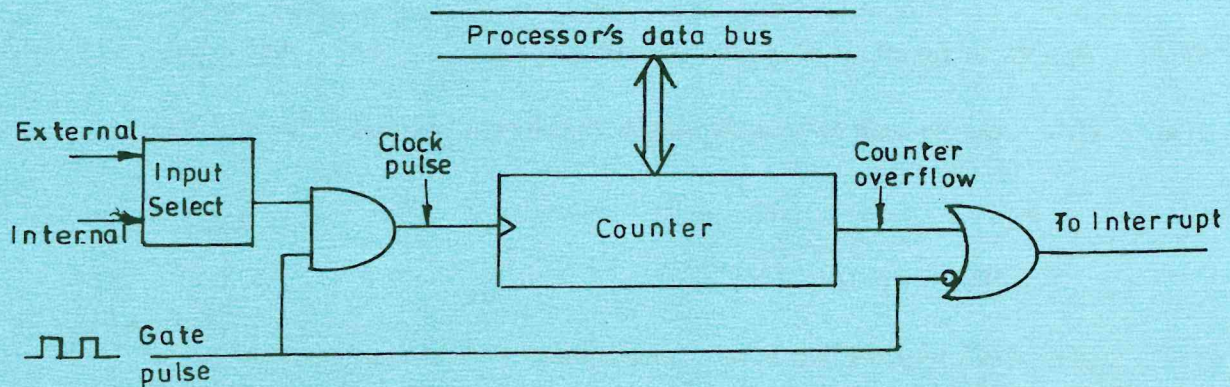


Fig. 2

- (d) Write an intel 8051 assembly language program to subtract 40 H from 9 FH and exchange the digits of the result. (5 marks)
8. (a) Describe any **three** instruction groups available in microcontroller processor instruction set. (6 marks)
- (b) (i) Highlight **three** differences between a data logger and data acquisition system. (8 marks)
- (ii) A data logging system must take 10 samples from each of 40 variables every second. Determine the maximum signal acquisition time. (6 marks)
- (c) An aircraft lamp is controlled by three switches A, B and C. The lamp lights when any two switches are closed or when switch C is closed. For the lighting system:
- (i) draw the truth table;
- (ii) write the simplified Boolean Expression for the lamp control;
- (iii) draw the corresponding ladder diagram. (6 marks)

Appendix A: Instruction Set of 8051 Microcontroller

Mnemonics, Arranged Alphabetically

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ACALL addr11	PC + 2 → (SP); addr11 → PC	2	2	
ADD A, direct	A + (direct) → A	2	1	C OV AC
ADD A, @Ri	A + (Ri) → A	1	1	C OV AC
ADD A, #data	A + #data → A	2	1	C OV AC
ADD A, Rn	A + Rn → A	1	1	C OV AC
ADDC A, direct	A + (direct) + C → A	2	1	C OV AC
ADDC A, @Ri	A + (Ri) + C → A	1	1	C OV AC
ADDC A, #data	A + #data + C → A	2	1	C OV AC
ADDC A, Rn	A + Rn + C → A	1	1	C OV AC
AJMP addr11	Addr11 → PC	2	2	
ANL A, direct	A AND (direct) → A	2	1	
ANL A, @Ri	A AND (Ri) → A	1	1	
ANL A, #data	A AND #data → A	2	1	
ANL A, Rn	A AND Rn → A	1	1	
ANL direct, A	(direct) AND A → (direct)	2	1	
ANL direct, #data	(direct) AND #data → (direct)	3	2	
ANL C, bit	C AND bit → C	2	2	C
ANL C, bit	C AND bit → C	2	2	C
CJNE A, direct, rel	[A <> (direct)]; PC + 3 + rel → PC	3	2	C
CJNE A, #data, rel	[A <> data]; PC + 3 + rel → PC	3	2	C
CJNE @Ri, #data, rel	[(Ri) <> data]; PC + 3 + rel → PC	3	2	C
CJNE Rn, #data, rel	[Rn <> data]; PC + 3 + rel → PC	3	2	C
CLR A	0 → A	1	1	
CLR bit	0 → bit	2	1	
CLR C	0 → C	1	1	0
CPL A	\bar{A} → A	1	1	
CPL bit	$\bar{\text{bit}}$ → bit	2	1	
CPL C	\bar{C} → C	1	1	C
DA A	A bin → A dec	1	1	C
DEC A	A - 1 → A	1	1	
DEC direct	(direct) - 1 → (direct)	2	1	
DEC @Ri	(Ri) - 1 → (Ri)	1	1	
DEC Rn	Rn - 1 → Rn	1	1	
DIV AB	A/B → AB	1	4	0 OV
DJNZ direct, rel	[(direct) - 1 <> 00]; PC + 3 + rel → PC	3	2	
DJNZ Rn, rel	[Rn - 1 <> 00]; PC + 2 + rel → PC	2	2	
INC A	A + 1 → A	1	1	
INC direct	(direct) + 1 → (direct)	2	1	
INC DPTR	DPTR + 1 → DPTR	1	2	
INC @Ri	(Ri) + 1 → (Ri)	1	1	
INC Rn	Rn + 1 → Rn	1	1	
JB bit, rel	[b=1]; PC + 3 + rel → PC	3	2	
JBC bit, rel	[b=1]; PC + 3 + rel → PC; 0 → bit	3	2	
JC rel	[C=1]; PC + 2 + rel → PC	2	2	
JMP @A + DPTR	DPTR + A → PC	1	2	
JNB bit, rel	[b=0]; PC + 3 + rel → PC	3	2	
JNC rel	[C=0]; PC + 2 + rel → PC	2	2	
JNZ rel	[A > 00]; PC + 2 + rel → PC	2	2	
JZ rel	[A = 00]; PC + 2 + rel → PC	2	2	
LCALL addr16	PC + 3 → (SP); addr16 → PC	3	2	
LJMP addr16	Addr16 → PC	3	2	
MOV A, direct	(direct) → A	2	1	
MOV A, @Ri	(Ri) → A	1	1	
MOV A, #data	#data → A	2	1	
MOV A, Rn	Rn → A	1	1	
MOV direct, A	A → (direct)	2	1	
MOV direct, direct	(direct) → (direct)	3	2	
MOV direct, @Ri	(Ri) → (direct)	2	2	
MOV direct, #data	#data → (direct)	3	2	
MOV direct, Rn	Rn → (direct)	2	2	
MOV bit, C	C → bit	2	2	C
MOV C, bit	bit → C	2	1	
MOV @Ri, A	A → (Ri)	1	1	
MOV @Ri, direct	(direct) → (Ri)	2	2	

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
MOV Rn, #data	#data → Rn	2	1	
MOVC A, @A+DPTR	(A+DPTR) → A	1	2	
MOVC A, @A+PC	(A+PC) → A	1	2	
MOVX A, @DPTR	(DPTR) [^] → A	1	2	
MOVX A, @Ri	(Ri) [^] → A	1	2	
MOVX @DPTR, A	A → (DPTR) [^]	1	2	
MOVX @Ri, A	A → (Ri) [^]	1	2	
NOP	PC + 1 → PC	1	1	
MUL AB	A × B → AB	1	4	0 OV
ORL A, direct	A OR (direct) → A	2	1	
ORL A, @Ri	A OR (Ri) → A	1	1	
ORL A, #data	A OR #data → A	2	1	
ORL A, Rn	A OR Rn → A	1	1	
ORL direct, A	(direct) OR A → (direct)	2	1	
ORL direct, #data	(direct) OR #data → (direct)	3	2	
ORL C, bit	C OR bit → C	2	2	C
ORL C, $\overline{\text{bit}}$	C OR $\overline{\text{bit}}$ → C	2	2	C
POP direct	(SP) → (direct)	2	2	
PUSH direct	(direct) → (SP)	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	
RL A	A0 ← A7 ← A6... ← A1 ← A0	1	1	
RLC A	C ← A7 ← A6... ← A0 ← C	1	1	C
RR A	A0 → A7 → A6... → A1 → A0	1	1	
RRC A	C → A7 → A6... → A0 → C	1	1	C
SETB bit	1 → bit	2	1	
SETB C	1 → C	1	1	1
SJMP rel	PC + 2 + rel → PC	2	2	
SUBB A, direct	A - (direct) - C → A	2	1	C OV AC
SUBB A, @Ri	A - (Ri) - C → A	1	1	C OV AC
SUBB A, #data	A - #data - C → A	2	1	C OV AC
SUBB A, Rn	A - Rn - C → A	1	1	C OV AC
SWAP A	Alsn ↔ Amsn	1	1	
XCH A, direct	A ↔ (direct)	2	1	
XCH A, @Ri	A ↔ (Ri)	1	1	
XCH A, Rn	A ↔ Rn	1	1	
XCHD A, @Ri	Alsn ↔ (Ri)lsn	1	1	
XRL A, direct	A XOR (direct) → A	2	1	
XRL A, @Ri	A XOR (Ri) → A	1	1	
XRL A, #data	A XOR #data → A	2	1	
XRL A, Rn	A XOR Rn → A	1	1	
XRL direct, A	(direct) XOR A → (direct)	2	1	
XRL direct, #data	(direct) XOR #data → (direct)	3	2	

ACRONYMS

addr11	Page address of 11 bits, which is in the same 2K page as the address of the following instruction.
addr16	Address for any location in the 64K memory space.
bit	The address of a bit in the internal RAM bit address area or a bit in an SFR.
C	The carry flag.
#data	An 8-bit binary number from 00 to FFh.
#data16	A 16-bit binary number from 0000 to FFFFh.
direct	An internal RAM address or an SFR byte address.
lsn	Least significant nibble.
msn	Most significant nibble.
rel	Number that is added to the address of the next instruction to form an address +127d to -128d from the address of the next instruction.
Rn	Any of registers R0 to R7 of the current register bank.
@Ri	Indirect address using the contents of R0 or R1.
[]	IF the condition inside the brackets is <i>true</i> , THEN the action listed will occur; ELSE go to the next instruction.
[^]	EXTERNAL memory location.
()	Contents of the location inside the parentheses.

Note that flags affected each instruction are shown where appropriate; any operations which affect the PSW address may also affect the flags.

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOF	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6

2F	1	ADD	A,R7
30	3	JNB	bit addr, code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr

8051 OpCodes en Hexadecimal.

62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data

95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1

8051 OpCodes en Hexadecimal.

C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A

FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Instruction Opcodes in Hexadecimal Order

8051 OpCodes en Hexadecimal.

THIS IS THE LAST PRINTED PAGE.