2507/302
MICROCONTROLLER TECHNOLOGY
Oct./Nov. 2019
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.
*Answer **FIVE** of the **EIGHT** questions in the answer booklet provided.*
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) Describe each of the following with respect to microcontrollers:

(i) internal bus width;
(ii) speed;
(iii) external memory.

(6 marks)

(b) Convert:

(i) 427.32_8 into hexadecimal;

(ii) 1101011.10001_2 into octal.

(6 marks)

(c) (i) Work out $10001_2 \div 101_2$ (to 3 decimal places). (3 marks)

(ii) Perform the following using 8-bit two's complement arithmetic:

$(10010_2 - 100011_2) \times (10)_2$. (5 marks)

2. (a) State:

(i) **two** types of photoelectric sensors;
(ii) **three** manufacturers of PLC.

(5 marks)

(b) An analogue signal is passed through an ADC then processed through a microcontroller and converted back to analogue before it is displayed on a CRO. Draw a schematic diagram of the system. (6 marks)

(c) Table 1 shows an Intel 8051 assembly language program.

Table 1

MOV R5, # 30 H
MOV R 7, # 24 H
MOV A, OOH
ADD A; R 5
ADD A, R 7
ADD, A
ADD, A
END

(i) draw a trace table for the program.

(ii) explain what the program accomplishes.

(9 marks)

3. (a) State:

(i) the function of a limit switch;

(ii) **three** methods of actuating a limit switch.

(4 marks)

(b) Explain **three** differences between reduced instruction set (RISC) and complex instruction set (CIS). (6 marks)

(c) A conveyor is controlled by a PLC using three switches. Switch 1 controls the motor, switch 2 presence or absence of the load, and switch 3 indicates presence or absence of an object along the operation area. For proper operation, the conveyor motor should be ON, load present and no obstruction.

(i) draw a truth table for the system operation;

(ii) draw a ladder diagram for the truth table in c (i);

(iii) write down the PLC program listing.

(10 marks)

4. (a) Draw a block diagram of a typical microcontroller and describe the main parts.

(9 marks)

(b) Write an assembly language program to perform the following operations:

(i)
```
  5 F H
x 3 D H
--------
```

(ii)
```
  80 H
÷ 40 H
--------
```

(8 marks)

(c) State the effect of the following microcontroller instructions:

(i) MOV A, R7;

(ii) ADDC A, @ RO;

(iii) MOV R 3, #45 H.

(3 marks)

5. (a) Describe each of the following PLC designs:

(i) unitary;
(ii) modular.

(4 marks)

(b) In an earoplane dashboard, a RED warning light (R) comes ON when the ignition (I) is switched ON and the Door (D) is not closed properly, OR the seat belt (S) is not fastened. The system is under PLC control.

(i) draw a functional block diagram;
(ii) write down a PLC program listing to realize the system.

(10 marks)

(c) With the aid of a ONE line instruction, describe the following microcontroller instructions:

(i) single-operand operations;
(ii) two-operand operations.

(6 marks)

6. (a) State **three** features of a typical microcontroller. (3 marks)

(b) Figure 1 shows a memory map of a typical microcontroller.

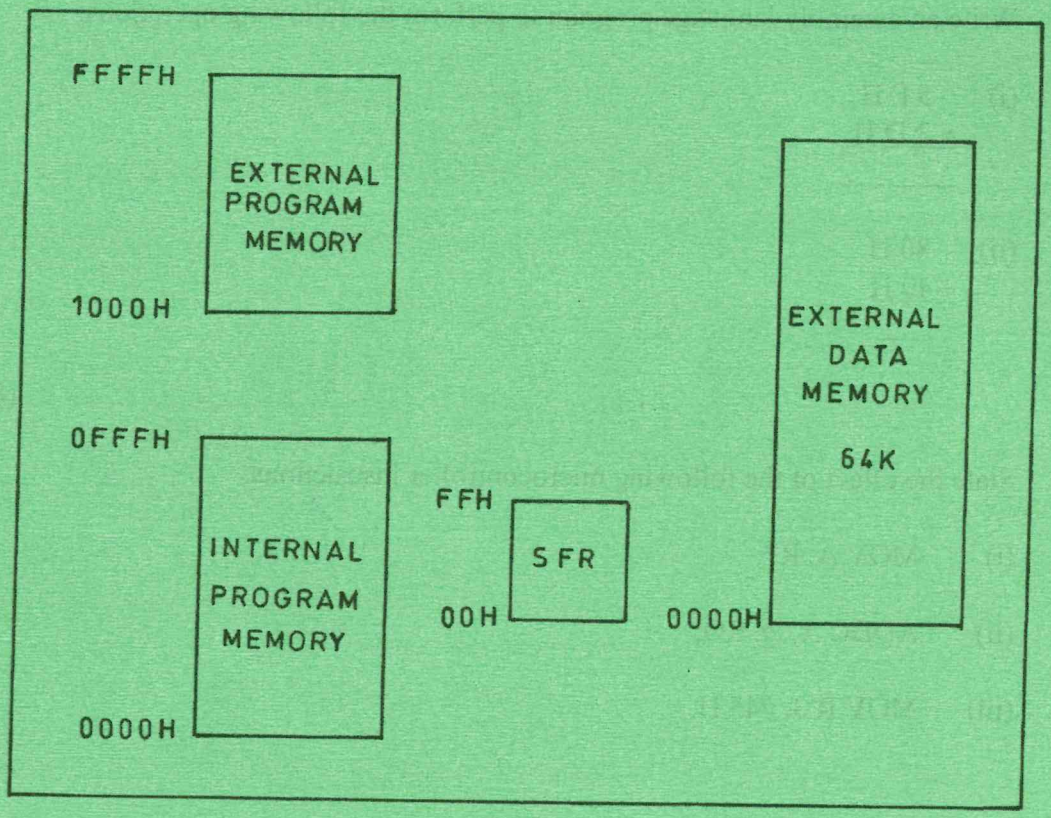

Fig. 1

Determine the following:

(i) size in kB of external program memory;

(ii) end address of external data memory;

(iii) state **three** functions of the SFR.

(9 marks)

(c) Table 2 represents an Intel 8051 program listing.

Table 2

MOV A, 78 H
ADD A, # 34 H
DAA
ADD CA, # 12 H
DAA
MOV R3, A
RET

(i) Explain what the program does.

(ii) Hand code the program into 8051 hexadecimal machine code.

(8 marks)

7. (a) State **four** main sources of interrupt in a microcontroller. (4 marks)

(b) Describe the following as used in SCADA systems:

(i) data acquisition;

(ii) logging.

(4 marks)

(c) (i) Define the following ADC/DAC specifications:

(I) linearity;
(II) speed.

(ii) With the aid of a schematic-block diagram, describe the operation of a counter-based analogue to digital converter (ADC).

(12 marks)

8. (a) Outline the main steps in programming a microcontroller for serial data transfer. (4 marks)

(b) (i) Differentiate between MOD 1 and MOD 2 of the 8051 microcontroller timer operations.

(ii) Table 3 shows a program listing of a microcontroller. Given that the clock frequency is 12 MHz, determine the total delay time. Assume **one** execution clock cycle = $\frac{1}{12}$ of crystal frequency.

Table 3

MOV A, # 10
MOV RO, # 05 H
MOV R1, # 05 H
Loop: ADD A, A
DEC RO
JNZ LOOP
MOV@ R1, A
RET

(12 marks)

(c) Explain **two** ways of implementing the microcontroller control unit. (4 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, $\overline{bit}$	C AND $\overline{bit}$ → 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	$\overline{A}$ → A	1	1	
CPL bit	$\overline{bit}$ → bit	2	1	
CPL C	$\overline{C}$ → C	1	1	C
DA A	Abin → Adec	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 x 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, $\overline{bit}$	C OR $\overline{bit}$ → 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 *true*, 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	Mnemo-nic	Operands
00	1	NOP	
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 Hexadecinal.

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 Hexadecinal.

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 Hexadecinal.

THIS IS THE LAST PRINTED PAGE.