B.E. Part-III 5th Semester (ETC) Final Examination, Dec. 2012

Subject: Microprocessor (ET-503)

Full marks: 70 **Time: 3hrs**

Answer Question no. 1 and any five from rest taking at least 2 from each group.
Answer should be brief and to the point.
*Unnecessary lengthy answers may result in **loss of marks**.*

Q1. Fill up the blanks with appropriate word assuming the μP is 8085.

i) Instruction ADI, 5FH Consists of _____ M/c cycles and _____ T-states.

ii) Maximum _____ numbers of address lines can be used to select memory based I/O ports.

iii) _____ nos of address bits are necessary to address A 256x8bit memory.

iv) RST 6.5 pin is a _____ interrupt.

v) Opcode Fetch M/c. cycle has maximum ____ T-state.

vi) To interrupt the μP through INTR interrupt signal should remain high for ____ T-state.

vii) ____ pin goes to high state during T1 state of each M/c cycle.

viii) Opcode Fetch is a special _____ _____ cycle and data moves to _____ _____ during Opcode Fetch. (10)

GROUP- I

Q2. Draw the transition state diagram of 8085 with suitable explanation. (12)

Q3. Design a memory system of 4KByte EPROM and one 16KByte RAM for an 8085 μP. Each RAM memory chip contains 4Kbyte memory. Draw the memory map of the system. Assume necessary control & data pins are available on the memory chips. (12)

Q4. a) What is addressing mode in microprocessors?

b) With suitable examples, explain the addressing modes of 8085 μP. (2+10)

Q5. Interface 4 Nos. 7 segment LEDs using PPI 8255 to a 8085 microprocessor to display decimal numbers up to 99. Draw the circuit diagram and write assembly language programme for implementation. Assumption (if any) must be mentioned. (12)

Q6. Draw timing diagrams for the following instruction with appropriate control and status signal. Explain in brief. CALL 2000 (12)

GROUP - II

Q7. a) Explain what happens when INTR in 8085 μP goes high.

b) A user wants to implement a hardwired interrupt through RST5. Draw the circuit diagram and explain steps involved in it. (4+8)

Q8. a) Write an 8085 assembly language program to multiply two BCD numbers and store the result in the memory locations. Draw flow chart for the program.

b) Write an 8085 assembly language program to subtract two BCD numbers in registers and result to be placed in HL register pair. (6+6)

Q9. Sketch the interface of any 8-bit A/D Converter having necessary μP compatible control signals to an 8085 μP. Write an assembly language program to read and store the 100 data bytes in the microcomputer system. Assumption if any must be mentioned. (6+6)

Q10. What is key debouncing? How it can be eliminated? How a 4x4 matrix keyboard can be interfaced using interrupt to give input to a μP based system? (2+2+8)

Move

MOV	A,A	7F
	A,B	78
	A,C	79
	A,D	7A
	A,E	7B
	A,H	7C
	A,L	7D
	A,M	7E
MOV	B,A	47
	B,B	40
	B,C	41
	B,D	42
	B,E	43
	B,H	44
	B,L	45
	B,M	46
MOV	C,A	4F
	C,B	48
	C,C	49
	C,D	4A
	C,E	4B
	C,H	4C
	C,L	4D
	C,M	4E
MOV	D,A	57
	D,B	50
	D,C	51
	D,D	52
	D,E	53
	D,H	54
	D,L	55
	D,M	56

Move (cont)

MOV	E,A	5F
	E,B	58
	E,C	59
	E,D	5A
	E,E	5B
	E,H	5C
	E,L	5D
	E,M	5E
MOV	H,A	67
	H,B	60
	H,C	61
	H,D	62
	H,E	63
	H,H	64
	H,L	65
	H,M	66
MOV	L,A	6F
	L,B	68
	L,C	69
	L,D	6A
	L,E	6B
	L,H	6C
	L,L	6D
	L,M	6E
MOV	M,A	77
	M,B	70
	M,C	71
	M,D	72
	M,E	73
	M,H	74
	M,L	75

XCHG EB

Move Immediate

MVI	A, byte	3E
	B, byte	06
	C, byte	0E
	D, byte	16
	E, byte	1E
	H, byte	26
	L, byte	2E
	M, byte	36

Load Immediate

LXI	B, dble	01
	D, dble	11
	H, dble	21
	SP, dble	31

Load / Store

LDAX B		0A
LDAX D		1A
LHLD adr		2A
LDA adr		3A
STAX B		02
STAX D		12
SHLD adr		22
STA adr		32

byte = constant, or logical / arithmetic expression that evaluates to an 8-bit data quantity (Second byte of 2-byte instructions)
dble = constant, or logical / arithmetic expression that evaluates to an 16-bit data quantity (Second and Third bytes of 3-byte instructions)
adr = 16-bit address (Second and Third bytes of 3-byte instructions)
* = all flags (C, Z, S, P, AC) affected.
** = all flags except CARRY affected.
(exception : INX and DCX affect no flags)
† = only CARRY affected.

Add*

ADD	A	87
	B	80
	C	81
	D	82
	E	83
	H	84
	L	85
	M	86
ADC	A	8F
	B	88
	C	89
	D	8A
	E	8B
	H	8C
	L	8D
	M	8E

Subtract *

SUB	A	97
	B	90
	C	91
	D	92
	E	93
	H	94
	L	95
	M	96
SBB	A	9F
	B	98
	C	99
	D	9A
	E	9B
	H	9C
	L	9D
	M	9E

Double Add †

DAD	B	09
	D	19
	H	29
	SP	39

Increment**

INR	A	3C
	B	04
	C	0C
	D	14
	E	1C
	H	24
	L	2C
	M	34
INX	B	03
	D	13
	H	23
	SP	33

Decrement **

DCR	A	3D
	B	05
	C	0D
	D	15
	E	1D
	H	25
	L	2D
	M	35
DCX	B	0B
	D	1B
	H	2B
	SP	3B

Specials

DAA *		27
CMA		2F
STC†		37
CMC†		3F

Rotate†

RLC		07
RRC		0F
RAL		17
RAR		1F

Logical*

ANA	A	A7
	B	A0
	C	A1
	D	A2
	E	A3
	H	A4
	L	A5
	M	A6
XRA	A	AF
	B	A8
	C	A9
	D	AA
	E	AB
	H	AC
	L	AD
	M	AE
ORA	A	B7
	B	B0
	C	B1
	D	B2
	E	B3
	H	B4
	L	B5
	M	B6
CMP	A	BF
	B	B8
	C	B9
	D	BA
	E	BB
	H	BC
	L	BD
	M	BE

Arith & Logical Immediate

ADI byte		C6
ACI byte		CE
SUI byte		D6
SBI byte		DE
ANI byte		E6
XRI byte		EE
ORI byte		F6
CPI byte		FE

Jump

JMP adr	C3
JNZ adr	C2
JZ adr	CA
JNC adr	D2
JC adr	DA
JPO adr	E2
JPE adr	EA
JP adr	F2
JM adr	FA
PCHL	E9

Call

CALL adr	CD
CNZ adr	C4
CZ adr	CC
CNC adr	D4
CC adr	DC
CPO adr	E4
CPE adr	EC
CP adr	F4
CM adr	FC

Return

RET	C9
RNZ	C0
RZ	C8
RNC	D0
RC	D8
RPO	E0
RPE	E8
RP	F0
RM	F8

Restart

0	C7
1	CF
2	D7
3	DF
4	E7
5	EF
6	F7
7	FF

Stack Ops

PUSH	B	C5
	D	D5
	H	E5
	PSW	F5
POP	B	C1
	D	D1
	H	E1
	PSW*	F1

XTHL	E3
SPHL	F9

Input/Output

OUT byte	D3
IN byte	DB

Control

DI	F3
EI	FB
NOP	00
HLT	76

New Instructions (8085 Only)

RIM	20
SIM	30

ASSEMBLER REFERENCE

Operators

Q
NUL
LOW, HIGH
*,/, MOD, SHL, SHR
+, -
NOT
AND
OR, XOR

Pseudo Instruction

General :

ORG
END
EQU
SET
DS
DB
DW

Macros :

MACRO
ENDM
LOCAL
REPT
IRP
IRPC
EXITM

Relocation :

ASEG	NAME
DSEG	STKLN
CSEG	STACK
PUBLIC	MEMORY
EXTRN	

Conditional Assembly :

IF
ELSE
ENDIF

Constant Definition

0BDH, 1AH	Hex
105D, 105	Decimal
720, 720Q	Octal
11011B, 00110B	Binary
'TEST', 'A', 'B'	ASCII