BIEE 3rd Year Examination, 2019
(1st Semester)

Microprocessors & Microcontrollers – Theory & Application

Time: Three hours | Full Marks: 100

Sl. No	**Answer all questions** **(Assume that the 8085 microprocessor board uses a 6.25 MHz crystal)**	Marks
1.	What are power-on and manual resets of the 8085 µp. With a neat circuit diagram explain how you implement power-on and manual reset of the 8085 µp. How does the 8085 µp enter and exit a HALT state?	3 2 5
2.	Write down the content of the accumulator and flag register as one steps through the following program: ORG 0000_H XRA A MVI B, $7F_H$ ACI 88_H ADD A SUB B ADD B HLT	10
3.	a) Write an 8085 assembly language program that divides the 16-bit unsigned integer stored in HL register pair by 8. Calculate the execution time of the code. b) Write an 8085 assembly language program that when executed generates a time delay of 120 µs. Calculate the % error in timing due to rounding-off of the count value.	8 2 8 2

BIEE 3rd Year Examination, 2019
(1st Semester)

Microprocessors & Microcontrollers – Theory & Application

Time: Three hours | Full Marks: 100

Sl. No	(Assume that the 8051 microcontroller board uses a 12 MHz crystal)	Marks
4.	Name the 8051 interrupt inputs and their vector addresses. How does the microcontroller resolves the following conflicts: • An interrupt request comes when an ISR in running. • Two interrupt requests come at the same instant of time.	4 2 4
5.	Write a stretch of code that converts an unsigned binary data stored in accumulator (in the range of 00_H to 63_H) to its packed BCD form.	5
6.	Briefly explain: • Why an FF_H is written on the port latch before every read operation? • What is the utility of the port latch output read buffer? • Why is the start bit of a serially transmitted/received data for the on-chip UART set to '0'?	 3 3 4
7.	Use timer-0 in interrupt driven mode to generate a 20% duty cycle 2 kHz pulse train at P1.2.	15
8.	An ICM7218B display controller is interfaced with an AT89C51 microcontroller and the connection description is as follows: • P1 is used as the data bus • P0.0 and P0.1 are used to drive the MODE and WR pins of ICM7218B. a) Draw the complete connection diagram of the microcontroller with the display controller. b) Write an 8051 assembly language program that dumps the display data stored in on-chip RAM locations 30_H to 37_H to the display controller.	 5 15

8085 Instruction Set

Instruction	Bytes	T-states	Instruction	Bytes	T-states
ACI data8	2	7	LXI RP, data16	3	10
ADC M	1	7	MOV M,REG	1	7
ADC REG	1	4	MOV REG,M	1	7
ADD M	1	7	MOV REG,REG	1	4
ADD REG	1	4	MVI M,data8	2	10
ADI data8	2	7	MVI REG,data8	2	7
ANA M	1	7	NOP	1	4
ANA REG	1	4	ORA M	1	7
ANI data8	2	7	ORA REG	1	4
CALL add16	3	18	ORI data8	2	7
CC add16	3	9/18	OUT add8	2	10
CM add16	3	9/18	PCHL	1	6
CMA	1	4	POP RP	1	10
CMC	1	4	PUSH RP	1	12
CMP M	1	7	RAL	1	4
CMP REG	1	4	RAR	1	4
CNC add16	3	9/18	RC	1	6/12
CNZ add16	3	9/18	RET	1	10
CP add16	3	9/18	RIM	1	4
CPE add16	3	9/18	RLC	1	4
CPI data8	2	7	RM	1	6/12
CPO add16	3	9/18	RNC	1	6/12
CZ add16	3	9/18	RNZ	1	6/12
DAA	1	4	RP	1	6/12
DAD RP	1	10	RPE	1	6/12
DCR M	1	10	RPO	1	6/12
DCR REG	1	4	RRC	1	4
DCX RP	1	6	RST n	1	12
DI	1	4	RZ	1	6/12
EI	1	4	SBB M	1	7
HLT	1	5	SBB REG	1	4
IN add8	2	10	SBI data8	2	7
INR M	1	7	SHLD add16	3	16
INR REG	1	4	SIM	1	4
INX RP	1	6	SPHL	1	6
JC add16	3	7/10	STA add16	3	13
JM add16	3	7/10	STAX RP	1	7
JMP add16	3	10	STC	1	4
JNC add16	3	7/10	SUB M	1	7
JNZ add16	3	7/10	SUB REG	1	4
JP add16	3	7/10	SUI data8	2	7
JPE add16	3	7/10	XCHG	1	4
JPO add16	3	7/10	XRA M	1	7
JZ add16	3	7/10	XRA REG	1	4
LDA add16	3	13	XRI data8	2	7
LDAX RP	1	7	XTHL	1	16
LHLD add16	3	16			

89C51 Instruction Set

Arithmatic operations		
mnemonic	byte	m/c cycle
ADD A, Rn/@Ri	1	1
ADD A, direct,#data	2	1
ADDC A, Rn/@Ri	1	1
ADDC A, direct/#data	2	1
SUBB A, Rn	1	1
SUBB A, direct	2	1
SUBB A, @Ri	1	1
SUBB A, #data	2	1
INC A/Rn/@Ri	1	1
INC direct	2	1
DEC A/Rn/@Ri	1	1
DEC direct	2	1
INC DPTR	1	2
MUL AB	1	4
DIV AB	1	4
DA A	1	1
Logical operations		
mnemonic	byte	cycle
ANL A, Rn/@Ri	1	1
ANL A, direct/#data	1	1
ANL A, #data	2	1
ANL direct, A	2	1
ANL direct, #data	3	2
ORL A, Rn/@Ri	1	1
ORL A, direct/#data	2	1
ORL direct, A	2	1
ORL direct, #data	3	2
XRL A, Rn/@@Ri	1	1
XRL A, direct/#data	2	1
XRL direct, A	2	1
XRL direct, #data	3	2
RL A	1	1
RLC A	1	1
RR A	1	1
RRC A	1	1
SWAP A	1	1
Program branching		
mnemonic	byte	m/c cycle
ACALL addr11	2	2
LCALL addr16	3	2
RET	1	2
RETI	1	2
AJMP addr11	2	2
LJMP addr16	3	2
SJMP add8	2	2
JZ rel	2	2
JNZ rel	2	2
CJNE A, direct, rel	3	2
CJNE A, #data, rel	3	2
CJNE Rn, #data, rel	3	2
CJNE @Ri, #data, rel	3	2
DJNZ Rn, rel	3	2
DJNZ direct, rel	3	2
NOP	1	1
Data transfer		
mnemonic	byte	m/c cycle
MOV A, Rn/@Ri	1	1
MOV A, direct/#data	2	1
MOV Rn, A	1	1
MOV Rn, direct	2	2
MOV Rn, #data	2	1
MOV direct, A	2	1
MOV direct, Rn	2	2
MOV direct, direct	3	2
MOV direct, @Ri	2	2
MOV direct, #data	3	2
MOV @Ri, A	1	1
MOV @Ri, direct	2	2
MOV @Ri, #data	2	1
PUSH direct	2	2
POP direct	2	2
XCH A, Rn	1	1
XCH A, @Ri	1	1
XCHD A, @Ri	1	1
Boolean variable manipulation		
mnemonic	byte	m/c cycle
CLR C	1	1
CLR bit	2	1
SETB C	1	1
SETB bit	2	1
CPL C	1	1
CPL bit	2	1
ANL C, bit	2	2
ANL C, /bit	2	2
ORL C, bit	2	2
ORL C, /bit	2	2
MOV C, bit	2	1
MOV bit, C	2	2
JC rel	2	2
JNC rel	2	2
JB bit, rel	3	2
JNB bit, rel	3	2
JBC bit, rel	3	2