TECHNICAL UNIVERSITY OF MOMBASA

School of Engineering and Technology

Department of Electrical and Electronic Engineering

UNIVERSITY EXAMINATION FOR:

Bachelor of Science in Electrical and Electronic Engineering

EEE 4407: MICROPROCESSOR II

END OF SEMESTER EXAMINATION

SERIES: AUGUST 2024

TIME: 2 HOURS

DATE: Pick DateSelect MonthPick Year

Instructions to Candidates

You should have the following for this examination

-Answer Booklet, examination pass and student ID

This paper consists of **five** Questions; Answer any THREE Questions.

Do not write on the question paper.

QUESTION ONE

a) For the program Table Q1(a),

i) Determine the number of times the instruction DCR C is executed

ii) Determine the number of times the program jumps to LOOP

iii) Explain what can be done to change the program to loop 200 times

iv) Determine the time delay of the program assume the clock frequency for the system is 2 MHz

Table Q1(a) [5 Marks]

Label	Mnemonic	T-states
	MVI C, 78H	7
LOOP	DCR C	4
	JNZ LOOP	10/7
	HLT	5

b)

i) Write a program that multiplies decimal 15 and 10 including a JZ instruction.

ii) Modify the program in (i) by using a JNZ instead of a JZ instruction

iii) Hand-assemble the program in (ii) starting at address 2000H.

iv) Change the multiplication part of the program in (iii) into a subroutine memory located at starting at address F006H.

v) Write a program that utilizes the subroutine to multiply any two 8-bit numbers.

[15 Marks]

QUESTION TWO

a)

(i) State any **THREE** needs for modular programming

(ii) State the **THREE** techniques used for the status saving and status restoring, during interrupts

[6 Marks]

b) Explain why DMA is a more efficient method of transferring blocks of data than either the programmed or the interrupt I/O and why it saves CPU time. [6 Marks]

c) Using appropriate equate statements, write a commented section of PIC16F84 assembly code to set 1 pin of Port B to be an output and the other 7 pins to be inputs. Figure Q2(c) shows the status register details from the PIC16F84 datasheet. [8 Marks]

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **IRP**: Register Bank Select bit (used for indirect addressing)
0 = Bank 0, 1 (00h - FFh)
1 = Bank 2, 3 (100h - 1FFh)
The IRP bit is not used by the PIC16F8X. IRP should be maintained clear.

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h - 7Fh)
01 = Bank 1 (80h - FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)
Each bank is 128 bytes. Only bit RP0 is used by the PIC16F8X. RP1 should be maintained clear.

bit 4: $\overline{TO}$: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3: $\overline{PD}$: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions) (For $\overline{borrow}$ the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred
Note: For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

Figure Q2(c)

QUESTION THREE

a) Outline the process for the Intel 8085 non-vectored interrupt process [8 Marks]

b) Table Q3 (b) is an Intel 8085 assembly language program.

(i) Determine the vector location for the second instruction in the program

(ii) State the op code that will most likely be found in this location

Table Q3(b)

LXI H, 4000H
RST 5
MOV C, A

[2 Marks]

c) Figure Q1(c) shows a peripheral device connected to the RST5.5 interrupt. After the CPU receives a data word in port 12H, it can send a high ACKNOWLEDGE bit (bit 7 of port 11H) back to the peripheral device. The starting address in the RST 5.5 vector location is F100H. Write a service subroutine that inputs 128 bytes of data from the peripheral device and store the data at address starting at 4000H.

[6 Marks]

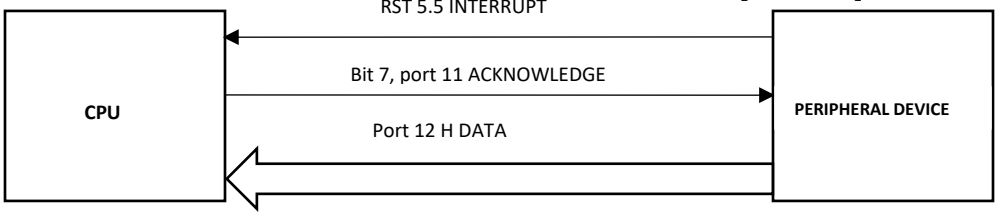

Fig. Q1(c)

d) Identify (in both binary and hexadecimal) the resulting values in the working register for each of the following:

i) MOVLW B'00011001'
SUBLW B'01001001'

ii) MOVLW B'10110011'
XORLW B'10001010'

iii) MOVLW B'11010010'
ANDLW B'00011011'

[6 Marks]

QUESTION FOUR

(a) Explain the purpose of the following Intel 8085 microprocessor control signals:

i) $\overline{WR}$
ii) ALE
iii) $CLK\ OUT$
iv) $\overline{RESET} \in ¿ ¿$
v) $IO/\overline{M}$

[5 Marks]

(b) The following 4 lines of assembly code will set 4 particular outputs from PortB to be high, without affecting the other 4 outputs.

```
BSF 0x06, 7
BSF 0x06, 6
BSF 0x06, 3
BSF 0x06, 0
```

Write two lines (only) of assembly code which achieve the same result [5 Marks]

(c) Explain the difference between the instructions CALL and GOTO and provide an example of when using each would be appropriate [4 Marks]

(d) Using the register file map diagram included as an appendix to this paper, explain the key functions of the registers located in data memory locations:

i) 0x01
ii) 0x85
iii) 0x10

[6 Marks]

QUESTION FIVE

a)

i) Write a program to exchange the higher and lower nibble of ten 8-bit numbers stored from location 3300H, using subroutine and parameter passing technique using memory location

ii) Explain any **FOUR** merits of subroutine use

[10 Marks]

b) In the system shown in Figure Q5 (b), a 3 MHz clock is connected to the TIMER IN input of the Intel 8156 timer. Write an assembly language program segment to configure the 8155 to divide the system clock by a factor of 2500 so as to produce a pulse train at the TIMER OUT pin. Include a start Timer Command, disable all port interrupts, make ports A and C input and port B output.

[10 Marks]

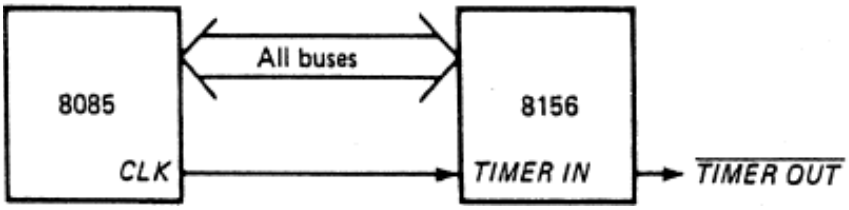

Fig Q5 (b)

8085A CPU INSTRUCTIONS IN OPERATION CODE SEQUENCE

Table 5-2

OP CODE	MNEMONIC		OP CODE	MNEMONIC		OP CODE	MNEMONIC		OP CODE	MNEMONIC		OP CODE	MNEMONIC		OP CODE	MNEMONIC	
00	NOP		28	DCX	H	56	MOV	D,M	81	ADD	C	AC	XRA	H	D7	RST	2
01	LXI	B,D16	2C	INR	L	57	MOV	D,A	82	ADD	D	AD	XRA	L	D8	RC	
02	STAX	B	2D	DCR	L	58	MOV	E,B	83	ADD	E	AE	XRA	M	D9	–	
03	INX	B	2E	MVI	L,D8	59	MOV	E,C	84	ADD	H	AF	XRA	A	DA	JC	Adr
04	INR	B	2F	CMA		5A	MOV	E,D	85	ADD	L	B0	ORA	B	DB	IN	D8
05	DCR	B	30	SIM		5B	MOV	E,E	86	ADD	M	B1	ORA	C	DC	CC	Adr
06	MVI	B,D8	31	LXI	SP,D16	5C	MOV	E,H	87	ADD	A	B2	ORA	D	DD	–	
07	RLC		32	STA	Adr	5D	MOV	E,L	88	ADC	B	B3	ORA	E	DE	SBI	D8
08	–		33	INX	SP	5E	MOV	E,M	89	ADC	C	B4	ORA	H	DF	RST	3
09	DAD	B	34	INR	M	5F	MOV	E,A	8A	ADC	D	B5	ORA	L	E0	RPO	
0A	LDAX	B	35	DCR	M	60	MOV	H,B	8B	ADC	E	B6	ORA	M	E1	POP	H
0B	DCX	B	36	MVI	M,D8	61	MOV	H,C	8C	ADC	H	B7	ORA	A	E2	JPO	Adr
0C	INR	C	37	STC		62	MOV	H,D	8D	ADC	L	B8	CMP	B	E3	XTHL	
0D	DCR	C	38	–		63	MOV	H,E	8E	ADC	M	B9	CMP	C	E4	CPO	Adr
0E	MVI	C,D8	39	DAD	SP	64	MOV	H,H	8F	ADC	A	BA	CMP	D	E5	PUSH	H
0F	RRC		3A	LDA	Adr	65	MOV	H,L	90	SUB	B	BB	CMP	E	E6	ANI	D8
10	–		3B	DCX	SP	66	MOV	H,M	91	SUB	C	BC	CMP	H	E7	RST	4
11	LXI	D,D16	3C	INR	A	67	MOV	H,A	92	SUB	D	BD	CMP	L	E8	RPE	
12	STAX	D	3D	DCR	A	68	MOV	L,B	93	SUB	E	BE	CMP	M	E9	PCHL	
13	INX	D	3E	MVI	A,D8	69	MOV	L,C	94	SUB	H	BF	CMP	A	EA	JPE	Adr
14	INR	D	3F	CMC		6A	MOV	L,D	95	SUB	L	C0	RNZ		EB	XCHG	
15	DCR	D	40	MOV	B,B	6B	MOV	L,E	96	SUB	M	C1	POP	B	EC	CPE	Adr
16	MVI	D,D8	41	MOV	B,C	6C	MOV	L,H	97	SUB	A	C2	JNZ	Adr	ED	–	
17	RAL		42	MOV	B,D	6D	MOV	L,L	98	SBB	B	C3	JMP	Adr	EE	XRI	D8
18	–		43	MOV	B,E	6E	MOV	L,M	99	SBB	C	C4	CNZ	Adr	EF	RST	5
19	DAD	D	44	MOV	B,H	6F	MOV	L,A	9A	SBB	D	C5	PUSH	B	F0	RP	
1A	LDAX	D	45	MOV	B,L	70	MOV	M,B	9B	SBB	E	C6	ADI	D8	F1	POP	PSW
1B	DCX	D	46	MOV	B,M	71	MOV	M,C	9C	SBB	H	C7	RST	0	F2	JP	Adr
1C	INR	E	47	MOV	B,A	72	MOV	M,D	9D	SBB	L	C8	RZ		F3	DI	
1D	DCR	E	48	MOV	C,B	73	MOV	M,E	9E	SBB	M	C9	RET	Adr	F4	CP	Adr
1E	MVI	E,D8	49	MOV	C,C	74	MOV	M,H	9F	SBB	A	CA	JZ		F5	PUSH	PSW
1F	RAR		4A	MOV	C,D	75	MOV	M,L	A0	ANA	B	CB	–		F6	ORI	D8
20	RIM		4B	MOV	C,E	76	HLT		A1	ANA	C	CC	CZ	Adr	F7	RST	6
21	LXI	H,D16	4C	MOV	C,H	77	MOV	M,A	A2	ANA	D	CD	CALL	Adr	F8	RM	
22	SHLD	Adr	4D	MOV	C,L	78	MOV	A,B	A3	ANA	E	CE	ACI	D8	F9	SPHL	
23	INX	H	4E	MOV	C,M	79	MOV	A,C	A4	ANA	H	CF	RST	1	FA	JM	Adr
24	INR	H	4F	MOV	C,A	7A	MOV	A,D	A5	ANA	L	D0	RNC		FB	EI	
25	DCR	H	50	MOV	D,B	7B	MOV	A,E	A6	ANA	M	D1	POP	D	FC	CM	Adr
26	MVI	H,D8	51	MOV	D,C	7C	MOV	A,H	A7	ANA	A	D2	JNC	Adr	FD	–	
27	DAA		52	MOV	D,D	7D	MOV	A,L	A8	XRA	B	D3	OUT	D8	FE	CPI	D8
28	–		53	MOV	D,E	7E	MOV	A,M	A9	XRA	C	D4	CNC	Adr	FF	RST	7
29	DAD	H	54	MOV	D,H	7F	MOV	A,A	AA	XRA	D	D5	PUSH	D			
2A	LHLD	Adr	55	MOV	D,L	80	ADD	B	AB	XRA	E	D6	SUI	D8			

D8 = constant, or logical/arithmetic expression that evaluates to an 8-bit data quantity.

D16 = constant, or logical/arithmetic expression that evaluates to a 16-bit data quantity.

Adr = 16-bit address.