COMSATS University Islamabad, Lahore Campus
Department of Electrical and Computer Engineering

☐Sessional-1 ☐ Sessional-II ■ Terminal Examination – SPRING 2021

Course Title:	Microprocessor System and Interfacing			Course Code:	CPE/EEE 342	Credit Hours:	4(3,1)
Course Instructor:	Dr. Abbas Javed/Dr. Naeem Shehzad			Program Name:	BCE, BEE		
Semester:	6th	Batch:	FA18	Section:	A & B	Date:	July 13, 2021
Time Allowed:	**180 Minutes**			**Maximum Marks:**		**50**	
Student's Name:				Reg. No.	CIIT/		/LHR

Important Instructions / Guidelines:

- This is an open book examination.
- The exam is online. The time limit to solve this exam is 3-hour and extra 15 mins are given to upload your exam.
- Write your name and registration number in the above-mentioned space using Microsoft Word.
- Solve the questions on your notebook and insert the images of your solution at the end of the question paper.
- Save your exam file in the following format and submit your exam file on CUonline:
- XXE342-S-I-FA1X-00X.PDF

Question 1: **[CLO1-C4-PLO3] [5]**

Analyze the following assembly language code using the knowledge of 8086 programming model, addressing mode and assembly language programming concepts. You are required to show the value of updated registers after execution of each instruction:

```
MOV DX, 02000H
MOV SP, DX
MOV DX, 0A000H
MOV SS, DX
MOV AX, 01D00H
MOV BX, 2000H
PUSH AX
PUSH BX
ADD AX, BX
PUSH AX
POP CX
POP DX
```

MST Termind Exam Solution

Q.1
1) DX 20 00
2) SP 20 00
3) DX A0 00
4) SS A0 00
5) AX 1D 00
6) BX 20 00
7) A000: 1FFE 1D00 SP 1FFE
A000: 1FFC 2000 SP 1FFC
AX -> 3D 00
A000: 1FFA 3D00 SP 1FFA
CX · 3D 00 SP: 1FFC
DX 20 00 SP: 1FFE

Question 2: **[CLO2-C5-PLO3] [4+6]**

Integrate total of 32KB SRAM and total of 16 KB EPROM with 8086 CPU. Available SRAM memory circuits are 4KB chips only and EPROM memory circuits are 2 KB only. Starting address for EPROM is X0000 H and choose suitable starting address for SRAM. Where X is the non-zero least significant digit of your registration number. Draw completely labelled schematic diagram of memory integration with CPU and mention starting and ending address of each chip.

2) Integrate 32 KB SRAM → available chip 4 KB

16 KB EPROM → available 2 KB chip

2K → A1 — A11

	A19 A18 A17 A16	A15 A14 A13 A12	A11 A10 A9 A8	A7 A6 A5 A4	A3 A2 A1	A0
90000	1 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0	0
90FFE	1 0 0 1	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1	0
90FFF	1 0 0 1	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1	1
91000	1 0 0 1	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0	0
11FFE	1 0 0 1	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1	0
11FFF	1 0 0 1	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1	1
12000	1 0 0 1	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0	0
12FFE	1 0 0 1	0 0 1 0	1 1 1 1	1 1 1 1	1 1 1	0
12FFF	1 0 0 1	0 0 1 0	1 1 1 1	1 1 1 1	1 1 1	1
13000	1 0 0 1	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0	0
13FFE	1 0 0 1	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1	0
13FFF	1 0 0 1	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1	1

A14 — A19 → CS.

A12 — A13 → select line of 2-4 decoder

A1 — A11 → 2 KB EPROM.

32 kB SRAM.

	A_{19} A_{18} A_{17} A_{16}	A_{15} A_{14} A_{13} A_{12}	A_{11} A_{10} A_9 A_8	A_7 A_6 A_5 A_4	A_3 A_2 A_1	A_0
A 0000	1 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0	0
	1 0 1 0	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1	0
A 1FFF	1 0 1 0	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1	1
	1 0 1 0	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0	0
	1 0 1 0	0 0 1 0	1 1 1 1	1 1 1 1	1 1 1	0
	1 0 1 0	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1	1
	1 0 1 0	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0	0
	1 0 1 0	0 1 0 1	1 1 1 1	1 1 1 1	1 1 1	0
	1 0 1 0	0 1 0 1	1 1 1 1	1 1 1 1	1 1 1	1
	1 0 1 0	0 1 1 0	0 0 0 0	0 0 0 0	0 0 0	0
	1 0 1 0	0 1 1 1	1 1 1 1	1 1 1 1	1 1 1	0
	1 0 1 0	0 1 1 1	1 1 1 1	1 1 1 1	1 1 1	1

A_{15} - A_{19} → CS

A_{13}, A_{14} → Select lines of Decoder

A_1 - A_{12} → Address inputs of Memory

Question 3: **[CLO2-C5-PLO3] [10]**

Integrate 8254 timer with 8086 microprocessor and program counter 1 so that it generates a continuous wave of X00 KHz at OUT1. Program counter 2 so that it generates an interrupt after every X00 micro-seconds where X is non zero least significant digit of your registration number. (For a student with registration number Fa18-BCE-107, X is 7). You are free to choose any suitable address for 8254 timer and suitable input clock frequency.

Q.3

For Roll number with last digit 9.

```
MOV  DX, 706H
MOV  AL, 01110101
OUT  DX, AL
MOV  DX, 706H
MOV  AL, 10110001
OUT  DX, AL
MOV  DX, 702H
MOV  AL, 90
out  DX, AL
MOV  AL, 00
Out  DX, AL
```

clock = input from device here

count = 9MH / 900kHz

count = 90.

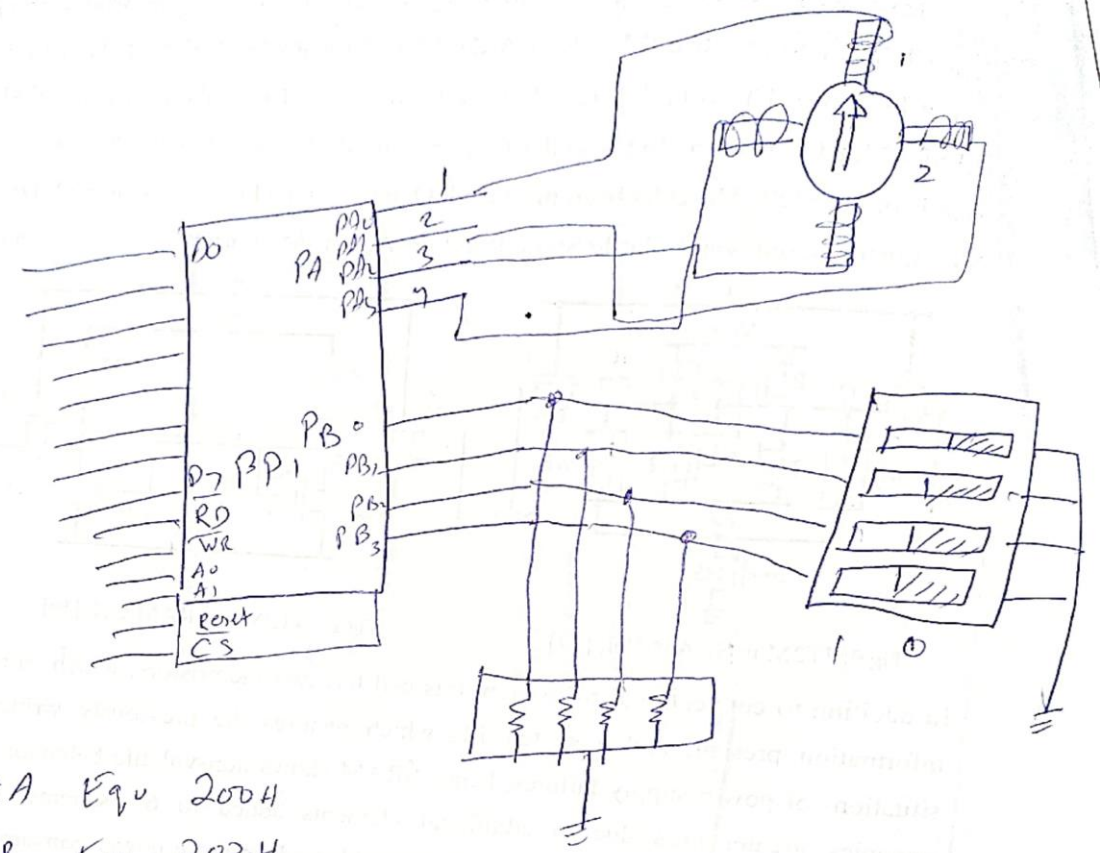
Count = 900 clock =

```
MOV  DX, 704H
MOV  AL, 00
OUT  DX, AL
MOV  AL, 09
OUT  DX, AL.
```

Question 4: **[CLO2-C5-PLO3] [6+4]**

Integrate 4 coil stepper motor in wave drive mode with 8086 microprocessor using programmable peripheral interface (PPI). Stepper motor should rotate 'n' steps in clockwise direction, where n will be entered by the user through 4 slide switches. Integrate 4 slide switches with PPI to input 4-bit data. Write an assembly program to stepper motor and slide switches and draw a schematic diagram. Choose suitable addresses for peripheral interfacing.

Q.4.



PortA Equ 200H
PortB Equ 202H
PortC Equ 204H

Bit:

1	0	0	0	0	0	1	0

→ 82H

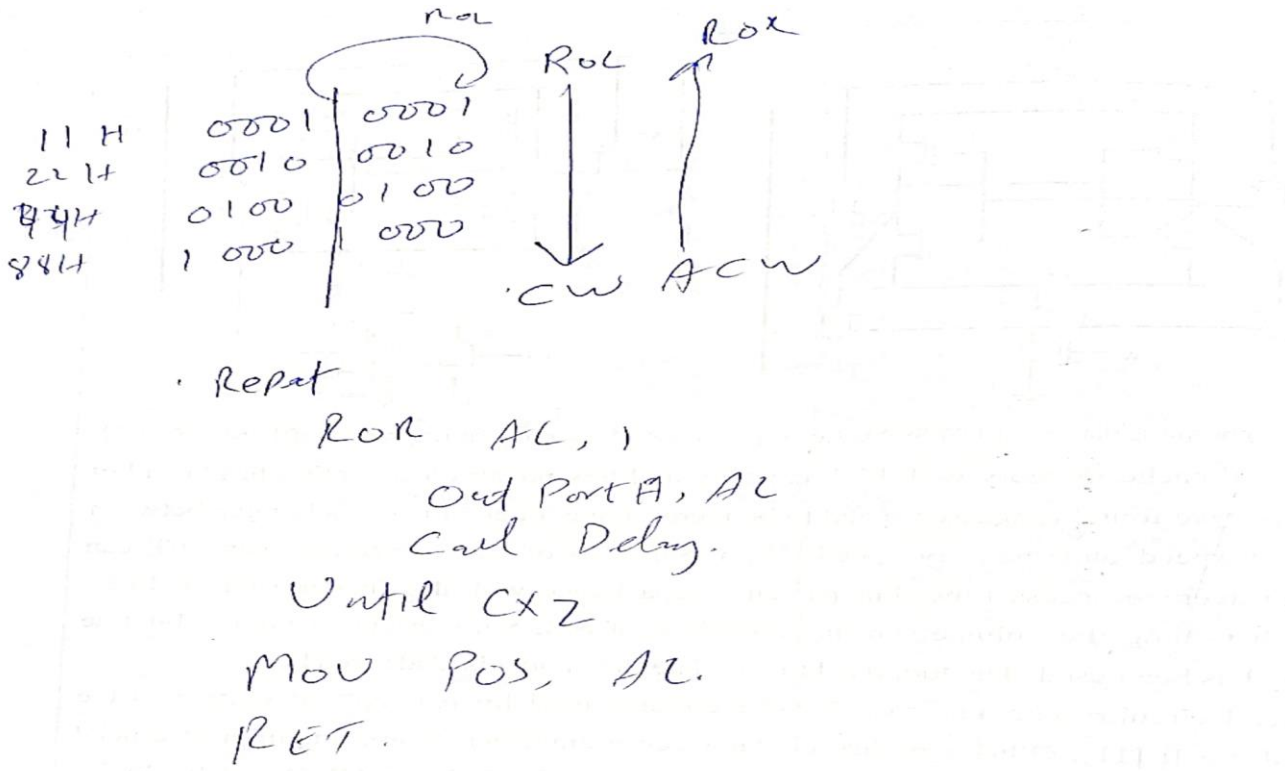
```
MOV AL, 82H
MOV DX, 206 H
Out DX, AL
```

	C_4	C_3	C_2	C_1	C_4	C_3	C_2	C_1
1	0	0	0	1	0	0	0	1
2								

```
IN AL, PortB  ⟶  MOV Slide Switch input to AL
MOV CX, AX
MOV AL, POS
```

Current angular location stored at location POS (in memory)

POS | 11 H, 22H, 44H 88 H |



Question 5: **[CLO3-C5-PLO3][5+10]**

Design a 8088 microprocessor based temperature controller for maintaining indoor thermal comfort. Temperature controller should control the HVAC to maintain the cooling setpoint entered by the occupant. Temperature controller should turn OFF the HVAC when indoor room temperature is equal to or less than cooling setpoint and turn ON the HVAC when the room temperature is greater than cooling setpoint. As a design engineer, you are required to implement the following tasks:

1. Interface a LM 35 temperature sensor using Analog to Digital Converter (ADC) with 8088 microprocessor. LM35 is a temperature sensor which can measure temperature in the range of -55°C to 150°C. The output voltage of this 3 pin temperature sensor is directly proportional to the ambient temperature and is given by the formula:

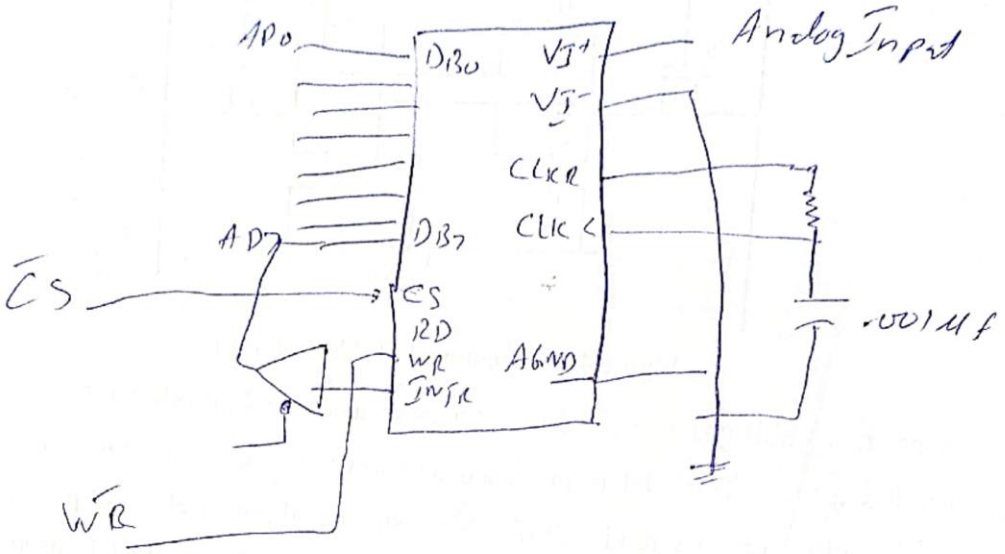
   $$V_{out} = K * T$$

   Where
   K = 10 mv/^{o}C
   T = ambient temperature
   For a temperature range of 0-100 ^{o}C, the output varies from 0-1 V with step of 10 mv. Starting address of the ADC is your registration number. (For a student with registration number Fa18-BCE-**107**, the starting address will be **107**H)
2. Integrate a suitable peripheral for allowing the occupant to enter the required cooling setpoint. Starting address of PPI is 0X000 H where X is least significant digit of your registration number. (For a student with registration number Fa18-BCE-107, the starting address will be 07000H)
3. Integrate a suitable peripheral for displaying the current room temperature.
4. Control the HVAC using digital switch connected to the digital output of the microprocessor.

a) Draw a completely labelled circuit diagram of temperature controller

b) Write an Assembly code for implementing the temperature controller.

Q.5



For Register ID F8H_BCE_0

```
ADCX  Proc  Near.
      Out   90H, AL
ADCX1:
      IN    AL, 92H
      Test  AL, 80H
      JNZ   ADCX1
      IN    AL, 90H
      RET
ADCX  ENDP
```

Resolution $\frac{5.1}{255} = 0.02V = 20mv$

For 0 - 100°C

ADC Value changes from 0 - 50

Multiply by 2 to get actual temperature

Lets suppose 26°C is ambient set point

ADC value is 13 for 26°C temperature

TempSet EQU 13

Call ADCX

Call key

CMP AL, TempSET

~~JE~~ HVACOFF

~~JGE~~ JG HVACON

JLE HVACOFF

HVACON:

OUT 100H, 01H.

Call Write

HVACOFF

Out 100H, 00H.

Call Write

Assume HVAC control Switch interfaced at 100H.

2) Open Ended- Student may use keypad or slide switches/buttons

Buttons

IN AL, DX

Keypads
assembly language version;

```
;KEY scans the keyboard and returns the key code in AL.
COLS EQU 4
ROWS EQU 4
PORTA EQU 50H
PORTB EQU 51H
KEY PROC NEAR USES CX BX
MOV BL,FFH ;compute row mask
SHL BL,ROWS
MOV AL,0
OUT PORTB,AL ;place zeros on Port B
.REPEAT ;wait for release
.REPEAT
  CALL SCAN
.UNTIL ZERO?
CALL DELAY10
CALL SCAN
.UNTIL ZERO?
.REPEAT ;wait for key
        .REPEAT
          CALL SCAN
        .UNTIL !ZERO?
        CALL DELAY10
        CALL SCAN
.UNTIL !ZERO?
MOV CX,00FEH
.WHILE 1 ;find column
        MOV AL,CL
        OUT PORTB,AL
        CALL SHORTDELAY ;see text
        CALL SCAN
        .BREAK !ZERO?
        ADD CH,COLS
        ROL CL,1
.ENDW
.WHILE 1 ;find row
        SHR AL,1
        .BREAK .IF !CARRY?
        INC CH
.ENDW
MOV AL,CH ;get key code
        RET
```

```
KEY ENDP
SCAN PROC NEAR
        IN AL,PORTA ;read rows
        OR AL,BL
        CMP AL,0FFH ;test for no keys
        RET
SCAN ENDP


3)  USE LCD to Display
WRITE PROC NEAR
MOV AL,BL ;BL to Port A
MOV DX,PORTA_ADDRESS
OUT DX,AL
MOV AL,0 ;write ASCII
MOV DX,PORTB_ADDRESS
OUT DX,AL
OR AL,00000100B ;Set E bit
OUT DX,AL ;send to Port B
AND AL,11111011B ;Clear E bit
NOP ;a small delay
NOP
OUT DX,AL ;send to Port B
CALL BUSY ;wait for completion
RET
WRITE ENDP
```