INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR

B.Tech. 5th SEMESTER MID-TERM EXAMINATIONS, OCTOBER 2021

MICROPROCESSOR BASED DESIGN (CS 3101)

[Answer script header should have the i) Name, Examination Roll No, G-suit-id]

Answer as many questions as you can. Answer should be brief and to the point. Full marks is 50. The sum of the total marks exceeding the full marks will be curtailed to 50.

Time allowed is 45 minutes

You may submit machine printed answer script by writing the answers after a question right on this file. For diagrams you may draw them separately and import them at appropriate locations. Don't forget to import your signature at the end of the answer script.

1. What is a flag? What is a flag register? Explain the use of the flags, namely, Carry and Zero, using small program segments (2 to 5 lines of appropriate code in assembly). [5]

ANS: A flag is a single bit used to reflect the effect of the last arithmetic or logic operation using which a decision is usually taken. For example, the zero flag (Z) is set if the preceding instruction produces 0 (zero) in a register (or ALU). A number of such flags (like, carry C, Sign S etc.) are available in a single register; known as the FLAG register.

```
Ex:     DCR  C
        JNZ  loop;  loop control using Z flag
```

```
        RAL
        JC   done;  bifurcation using Carry flag
```

2. What is bus contention? Explain bus contention with a suitable diagram. [3]

ANS: When two or more devices try to write on the same bus we get BUS CONTENTION – this happens when an output device is writing 0 (logic low) the other may write 1 (logic high) leading to low impedance path between power supply and ground.

<diagram here>

BUS contention may damage the circuit due to high current flow moreover the logic state at that point is indeterminate.

3. Draw the timing diagram of the machine cycles (showing the content of the address and data buses in each cycle) needed to execute the following return instruction (OF m/c cycle of RET instruction is 4T). [8]

```
ORG 3000H
```

LXI SP, 0A0F0H

SUBX EQU 0F0BH

CALL SUBX

:

SUBX: ; some instructions

RET

<diagram here>

The RET instruction takes 10T (OF[4]+MR[3]+MR[3])

	Address	Data
OF	NOT KNOWN	C9
MR (1)	A0EF	30
MR (2)	A0EE	06

Note: i) That the exact assembled location of RET is not know as it depends on the number and type of the instructions in the subroutine body. Ii) Return point after CALL SUBX is 3006H. And this is stored in locations 0A0EFH and 0A0EEH.

4. What is wait state? Why is it provided? Draw the IF (Instruction fetch) machine cycle with two wait states. [5]

ANS: Wait state is a pseudo T-state injected in the machine cycle to stretch it to cope up with the low speed memory or peripherals.

<diagram here>

5. Write a subroutine in assembly language that converts an 8-bit binary number to corresponding Gray code. The subroutine gets the 8-bit binary in Acc register and produces output in Acc. register. Show the calling of this conversion function from the main routine as well. [8]

ANS: For n-bit binary number Gn-1 = Bn-1 = 0 XOR Bn-1 ; Gn-2 = Bn-1 XOR Bn-2 … G0 = B1 XOR B0.

Bin2gray:

; this subroutine gets a binary number (8-bit) in Acc and returns the Gray in Acc

```
PUSH  B      ; B is saved
MOV B, A     ; keep a copy
```

```
    STC
    CMC  ; Carry is set to 0
RAR          ; MOVING 0 TO b7 of acc, b7 to b6 and so on
XOR  B       ; XOR converts the binary to gray as explained above
POP  B       ; B is restored
RET
```

6. Write a subroutine that generates a delay using a 16-bit delay count made available through the caller calling this subroutine. Also, calculate the maximum delay possible for a clock speed of 3 MHz.

[8]

ANS:

```
Delay16:
; this subroutine uses DE as the delay value – supposedly loaded by the caller
; calls : Noe
; destroys : A, B

DCX   D       ; 6T
MOV   A, E    ; 4T
ORA   D       ; 4T
JNZ   Delay16         ; 7T/10T
RET                   ; 10T
```

Max. delay for 0FFFFH (or 65535) in DE:

= 64534 x (6+4+4+10) + (14 + 7 + 10) [it loops through n-1 times] T

Delay in sec. = (1/3000000) x [(65535 x 24) + 7] s = 0.52428 s

7. Using a standard 3-8 decoder (say 74LS138) where you have 3-inputs, 3 enable lines (active low, active low, active high) and 8- active low outputs show the decoding to accommodate one 1 x 2K

EPROM, 2 x 2K RAM and 1 input and 1 output port (memory mapped with the same address A000H). Fold back is allowed.

[8]

<diagram here>

8. Explain the assembler directives ORG, EQU, DB, DW and $ with appropriate examples. [5]

ORG stands for Origin and used to set the PLC (assembly starting point for a block of code)

ORG 2000H; the next instruction would be assembled from 2000H

EQU stands for Equate and used to define a constant as shown

PORTX EQU 10H; assembler would replace the symbolic constant PORTX by 10H

DB means Define Byte and used to define a (or a number of) bytes as shown

ORG 5000H

DB 5, 1, 0AH; this would generate 5, 1 and 0A in location 5000, 5001 and 5002

DB is also used to define a string like

DB "CST" ; ASCII code for C, S and T would be assembled

DW stands for define a word – similar to DB but generates 16 bit data like

DW 5020H; here 20H and 50H would be assembled in two consecutive memory locations following the little endian form.

Signature: AD