MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

III SEMESTER B.TECH. (CSE)

IN SEMESTER EXAMINATION

SUBJECT: COMPUTER ORGANIZATION AND ARCHITECTURE (CSE 2151)

Time:10.30AM- 12.00 NOON   Date: 15/12/2021   MAX.MARKS: 20

Scheme of Evaluation Set2

1 Calculate the product of 13.125 and 17.5 in IEEE 32-bit format representation. Show all the steps clearly and represent the final answer in both IEEE 32-bit format and decimal representation 4

Step i: Convert 13.125 to binary representation **(0.5)**

1101.001→ After normalizing: 1.101001 ×2^3

E'=3+127= 130= $10000010_{(2)}$

In IEEE 32-bit format: 0 10000010 10100100000000…….

Step ii: Convert 17.5 to binary representation **(0.5)**

10001.1→ After normalizing: 1.00011×2^4

E'=4+127= 131= $10000011_{(2)}$

In IEEE 32-bit format: 0 10000011 000110000000000…….

Step 1 : **(0.5)**

Add exponents and subtract 127.

130+131-127= 134

Step 2:

Multiply the mantissa **(1)**

```
          1.101001 × 1.00011
          1 101001
         11 01001
        000 0000
       0000 000
      00000 00
 _____110100 1
      111001 011011
```

1.101001 × 1.00011= 1.11001 011011 **(0.5)**

The result is already normalized.

Step 3:

Adjust the exponent if required: Not applicable

In 32-bit representation: **(0.5)**

0 10000110 11001011011000…

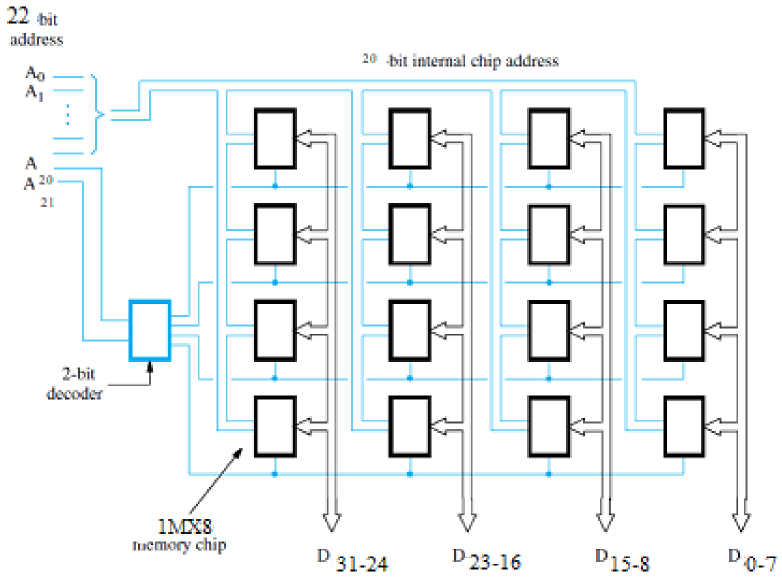
= 1. 11001011011000 X 2^7

=11100101.1011000…

=128+64+32+4+1+ 0.5+0.125+.0625

In decimal representation =229.6875 **(0.5)**

2 A. Construct a large memory of capacity 4Mx32 using 1M x 8 chips. Draw the structure and indicate clearly, the number of address lines and the data lines in the diagram 3

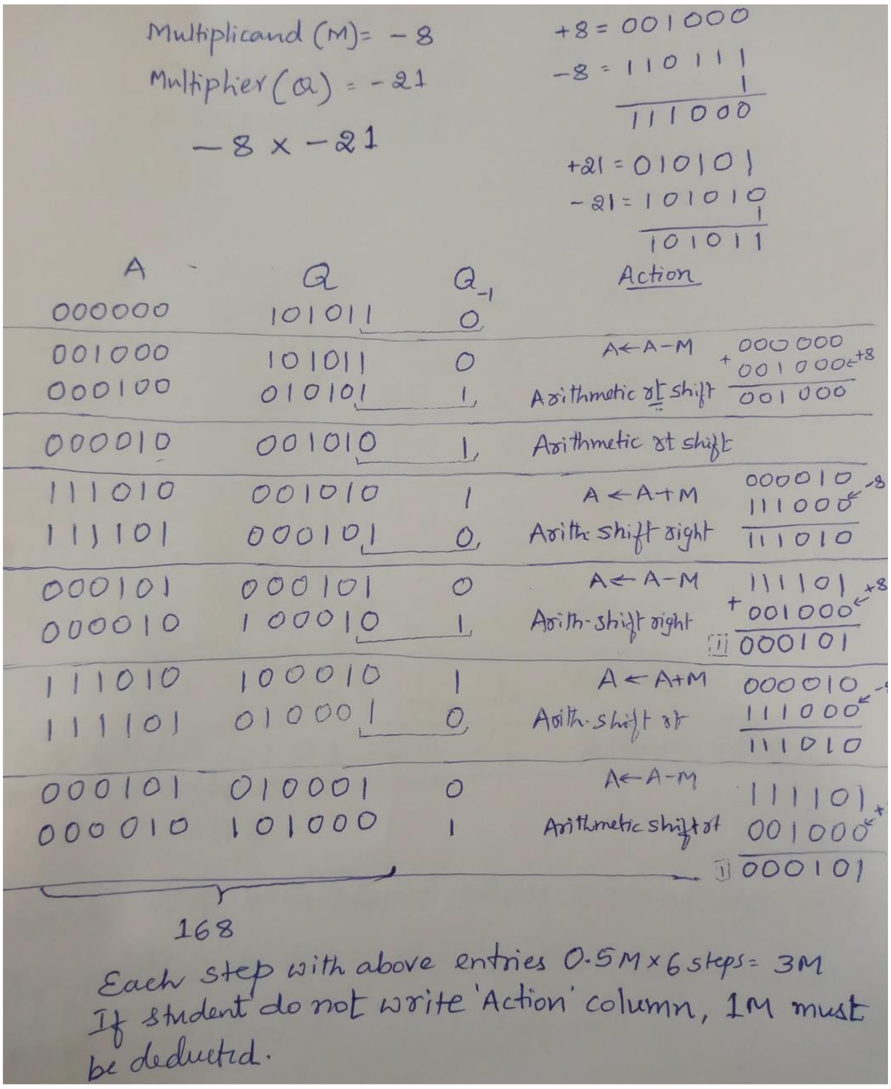


Identifying correct number of rows and colums-> 0.5m
Identifying no. of address lines for Large memory-0.5M
Identifying no. of address lines for small memory-0.5M
Writing decoder with proper inputs and outputs-0.5M
Writing proper data lines : 0.5M
Overall diagram: 0.5M

B. Explain the factors on which the length of the microinstruction is dependent on. 1

The length of the μinstruction is dependent on factors:
- How many μoperations will have to be activated simultaneously.
- The method by which the address of next μinstruction is specified.

3 Using Booth’s algorithm for 2’s complement multiplication, show how to multiply the multiplicand (-8) by the multiplier (-21) 3



Multiplicand (M) = −8

Multiplier (Q) = −21

−8 × −21

+8 = 001000
−8 = 110111
           1
111000

+21 = 010101
−21 = 101010
            1
101011

A	Q	Q_{-1}	Action	
000000	101011	0		
001000	101011	0	A←A−M	000 000 + 001 000 ←+8 001 000
000100	010101	1	Arithmetic rt shift	
000010	001010	1	Arithmetic rt shift	
111010	001010	1	A←A+M	000010 ←−8 111000 111010
111101	000101	0	Arith shift right	
000101	000101	0	A←A−M	111101 ←+8 + 001000 $\boxed{1}$ 000101
000010	100010	1	Arith-shift right	
111010	100010	1	A←A+M	000010 ←− 111000 111010
111101	010001	0	Arith. shift rt	
000101	010001	0	A←A−M	111101 ←+ 001000 $\boxed{1}$ 000101
000010	101000	1	Arithmetic shift rt	

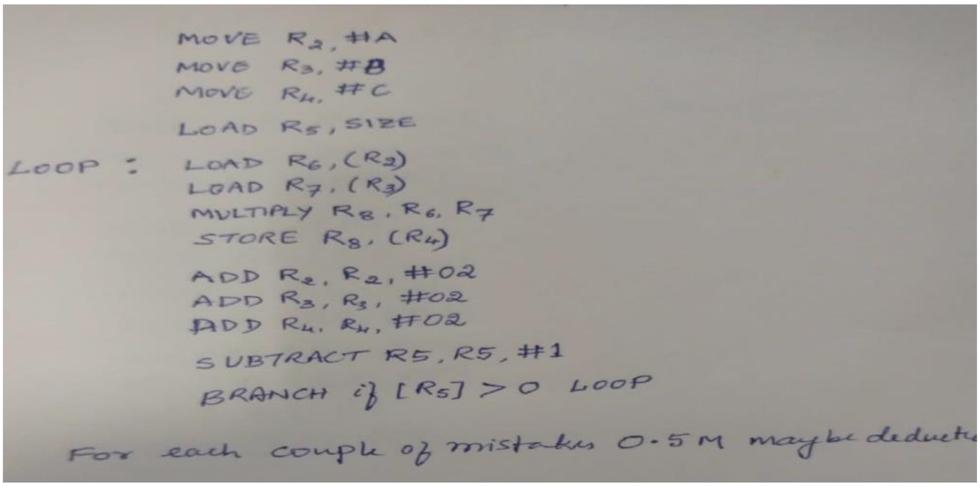
168

Each step with above entries 0.5M × 6 steps = 3M

If student do not write 'Action' column, 1M must be deducted.

If student do not write the ‘Action steps’ then 1M may be deducted.
0.5M for each step X 6 = 3M

4 Assume that an array A is stored in consecutive word locations starting from address A and an array B is stored in consecutive word locations starting from address B. Write a RISC-style program to find the product of the corresponding elements of two arrays and store it in the third array C starting at address C. Assume uniform sized arrays with the size of the array stored in memory location named SIZE. Assume a byte addressable memory and word length of two bytes. 3



```
          MOVE  R2, #A
          MOVE  R3, #B
          MOVE  R4, #C
          LOAD  R5, SIZE                          1
LOOP :    LOAD  R6, (R2)
          LOAD  R7, (R3)
          MULTIPLY R8, R6, R7
          STORE R8, (R4)                          1
          ADD R2, R2, #02
          ADD R3, R3, #02
          ADD R4, R4, #02
          SUBTRACT R5, R5, #1                     1
          BRANCH if [R5] > 0  LOOP
```

For each couple of mistakes 0.5M maybe deducte

5 Divide 25 by 3 using Restoring division method. Indicate all the steps clearly 3

A	Q	
00000	11001	Initial values M=00011 -M=11101
00001 11101 11110 00011 00001	10010 10010	Shift A,M to the left A=A-M Set Q0=0, Restore A I Cycle
00011 11101 00000	00100 00101	Shift A,M to the left A=A-M Set Q0=0, Restore A II Cycle
00000 11101 11101 00011 00000	01010 01010	Shift A,M to the left A=A-M Set Q0=1 III Cycle
00000 11101 11101 00011 00000	10100 10100	Shift A,M to the left A=A-M Set Q0=0, Restore A IV Cycle
00001 11101 11110 00011 00001	10100 01000	Shift A,M to the left A=A-M Set Q0=0, Restore A V Cycle
Remainder	Quotient	

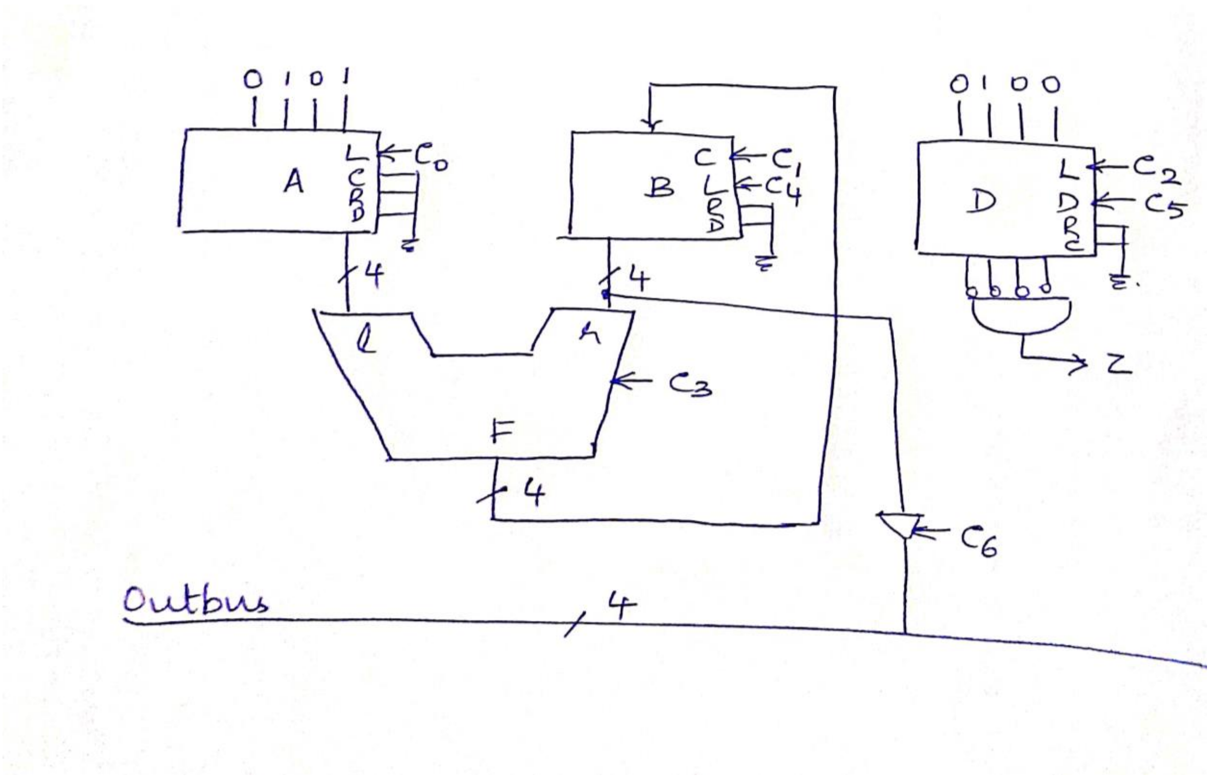
Each step : 0.5 X 5 = 2.5M ;Indicating quotient and remainder 0.5M

6 Consider the Register Transfer Description given below: 3

```
        Declare Registers A[4],B[4],D[4]
        Declare buses Outbus[4]
          Start: A←5, B←0, D←4;

Loop:            B←B+A
                   D←D-1;
        If D<> 0 then go to Loop
        Outbus = B;
        Halt: go to Halt
```

Design a neat processing section identifying all the control points



Writing 4 blocks: 0.5M

Showing width of data path and inputs: 0.5M

Identifying control signals and output: 2M

For any mistake : 0.5M deducted