



Course:(Code) CS234AI	Applied digital logic design and computer organisation		Semester : 3 <sup>rd</sup> BE
Date : Oct 2024		Duration : 120 minutes	Staff : DD/PSB/KS/KBR/KB
Name :	USN :	Section :	A/B/C/D/E/CD/CY

### PART A

Sl. No	Quiz questions	Mark s	* L1- L6	*CO
1.1	What are the minimum number of 2 input NAND gates required to implement the function $A \oplus B \oplus C$	1	L2	CO2
1.2	Draw the truth table for a system which accepts 4 inputs and gives a high output when more than two inputs are high.	1	L2	CO2
1.3	Subtract the following numbers using 2's complement. 15 from 13	1	L2	CO2
1.4	Construct the Booth recording of a multiplier for the following sequence.  1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0	1	L2	CO2
1.4	Simplify the following expression and write the prime implicants and essential prime implicants. $F(a,b,c,d) = \pi M(0,1,4,5,7,8,9,11,14,15) - dc(2,3)$	2	L2	CO2
1.5	With example, specify the significance of the overflow in binary addition and binary subtraction?	2	L2	CO1
1.6	Represent the number + 35.25 using 32 bit(single precision) floating point representation.	2	L2	CO1

### PART B

Sl. No	Answer all the questions	Mark s	* L1- L6	*CO
1a	A temperature control system has four inputs: A (Temperature High), B (Temperature Low), C (Humidity High), and D (System Active). The system should activate under these conditions: When the temperature is high and humidity is low ( $A=1, B=0, C=0$ ). When the system is active regardless of temperature and humidity ( $D=1$ ). When the temperature is low and humidity is high ( $B=1, C=1$ ).  Create a truth table, fill in the K-map and find the simplified expression for the system activation condition.	4	L3	CO2
b	Design a single circuit for BCD to excess 3 and excess 3 to BCD converter using 4 bit parallel adder and appropriate gates)	6	L3	CO2
2a	Using Modified Booth's multiplication algorithm, multiply +9 with -6. (Note: +9 is the multiplicand and (-6) is the multiplier)	4	L3	CO1
b	Illustrate the design of a single decade BCD adder(decimal adder) with correction network and show the truth table	6	L2	CO2
3	Explain the process for calculating carry in a Carry Look Ahead adder, design a four bit carry look ahead adder(derive the equations) How does	10	L3	CO2

PTO

	propagation delay in a Carry Look ahead Adder compare to that of a parallel adder?			
4	A home security system has four sensors: A (Garage Door Open), B (Front Door Open), C (Back Door Open), and D (Alarm Active). The alarm activates in the following cases: If any door is open (at least one of A,B, or C is 1). If the alarm is already active (D=1).  Task: Develop a truth table, and using Quine Mc-cluskey method, obtain the essential prime implicants for generating the alarm.	10	L4	CO2
5	Draw the flow chart for restoring division algorithm and perform the division operation on the following unsigned integers:Dividend:11 Divisor:5	10	L3	CO1

**Course Outcomes: After completing the course, the students will be able to:-**

CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyze the models used for designing various Combinational and Sequential circuits
CO 3	Develop applications of synchronous sequential networks using flip flops, registers and counters
CO 4	Design optimized modern processors and memories for given specifications
CO 5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test	Max Marks	18	42					16	34	10		



R V College of Engineering  
Department of Computer Science and Engineering(data science and cyber security)  
ADLDCO CIE - I : Question Paper and scheme and solution

Course: (Code)CS234 AI	Applied digital logic design and computer organisation	Semester : 3 <sup>rd</sup> BE
Date : Oct 2024	Duration : 120 minutes	Staff :DD/PSB/KS/KBR/KB
Name :	USN :	Section : A/B/C/D/E/CD/CY

**PART A**

Sl. No	Quiz questions	Mark s	* L1- L6	*CO																																																
1.1	<p>What are the minimum number of 2 input NAND gates required to implement the function <math>A \oplus B \oplus C</math> 8 Nand gates</p>	1	L2	CO2																																																
1.2	<p>Draw the truth table for a system which accepts 4 inputs and gives a high output when more than two inputs are high.</p> <p><input type="checkbox"/> The output Y is set to 1 when there are more than two inputs that are high (1).</p> <p><input type="checkbox"/> Specifically, the combinations that yield a high output are:</p> <ul style="list-style-type: none"> <li>• A=0, B=1, C=1, D=1 (3 high)</li> <li>• A=1, B=0, C=1, D=1 (3 high)</li> <li>• A=1, B=1, C=0, D=1 (3 high)</li> <li>• A=1, B=1, C=1, D=0 (3 high)</li> <li>• A=1, B=1, C=1, D=1 (4 high)</li> </ul> <p><b>A B C D Y (Output)</b></p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td></td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td></td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td></td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td></td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td></td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td></td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> </table>	0	0	0	0	0		0	0	0	1	0		0	0	1	0	0		0	0	1	1	0		0	1	0	0	0		0	1	0	1	0		0	1	1	0	0		0	1	1	1	1		1	L2	CO2
0	0	0	0	0																																																
0	0	0	1	0																																																
0	0	1	0	0																																																
0	0	1	1	0																																																
0	1	0	0	0																																																
0	1	0	1	0																																																
0	1	1	0	0																																																
0	1	1	1	1																																																

	1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1		
1.3	<p>Subtract the following numbers using 2's complement. 15 from 13</p>	1	L2 CO2
1.4	<p>Construct the Booth recording of a multiplier for the following sequence.</p> <p>1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0</p> <p>Soln: -1 +1 -1 +1 0 0 0 0 0 0 0 -1 +1 -1 +1 -1 +1 0 -1 0</p>	1	L2 CO2
1.5	<p>Simplify the following expression and write the prime implicants and essential prime implicants. <math>F(a,b,c,d) = \pi M(0,1,4,5,7,8,9,11,14,15) - dc(2,3)</math></p> <p>Ploting on kmap-1m and EPI-1m</p>	2	L2 CO2
1.6	<p>With example, specify the significance of the overflow in binary addition and binary subtraction?</p> <p>Overflow occurs in binary addition when the result of adding two binary numbers exceeds the range that can be represented with a fixed number of bits.</p> <p><b>Example:</b> consider an 8-bit binary system:</p> <ul style="list-style-type: none"> <li>• <b>Binary Numbers:</b> <ul style="list-style-type: none"> <li>○ A = 01111111 (127 in decimal)</li> </ul> </li> </ul>	2	L2 CO1

	<ul style="list-style-type: none"> <li>○ B = 00000001 (1 in decimal)</li> </ul> <p><b>Addition:</b></p> $  \begin{array}{r}  01111111 \\  + 00000001 \\  \hline  10000000  \end{array}  \quad =1\text{mark}  $ <p>Overflow in binary subtraction occurs when the result of subtracting two numbers yields a result that is outside the representable range.</p> <p><b>Example:</b> Consider the same 8-bit system:</p> <ul style="list-style-type: none"> <li>● <b>Binary Numbers:</b> <ul style="list-style-type: none"> <li>○ A = 00000000 (0 in decimal)</li> <li>○ B = 00000001 (1 in decimal)</li> </ul> </li> </ul> <p>A = 00000000 (0 in decimal) B = 01111111 (127 in decimal)</p> $  \begin{array}{r}  00000000 \\  - 01111111 \\  \hline  10000001  \end{array}  \quad \text{(which is -127 in decimal)}  $ <p>If A (0) is less than B (127) and the result appears to be a large positive number (10000001), it indicates that an underflow occurred because the expected result (negative value) cannot be represented correctly in the range. ==1mark</p>			
1.7	<p>Represent the number + 35.25 using 32 bit(single precision) floating point representation.</p> <p>35 in decimal = 100011 in binary. <b>Convert the fractional part (0.25) to binary:</b> 0.25 in decimal = 0.01 in binary</p> <p>35.25 in binary is 100011.01<sub>(2)</sub>. We need to express 100011.0100011.01100011.01 in normalized form:</p> <p>Move the binary point 5 places to the left:</p> <ul style="list-style-type: none"> <li>○ 1.0001101×2<sup>5</sup> bias is 127, so 127+5=132</li> <li>○ 132 in decimal = 100001001000010010000100</li> </ul> <p>Mantissa: 000110100000000000000000</p> <p>0 10000100 000110100000000000000000</p>	2	L2	CO1

## PART B

Sl. No	Answer all the questions	Mark s	* L1-L6	*CO
1a	A temperature control system has four inputs: A (Temperature High), B (Temperature Low), C (Humidity High), and D (System Active). The system should activate under these conditions: When the temperature is high and humidity is low (A=1,B=0,C=0). When the system is active regardless of temperature and humidity (D=1). When the temperature is low and humidity is high (B=1,C=1). Create a truth table, fill in a K-map,	4	L3	CO2

and find the simplified expression for the system activation condition.  
 Conditions for Activation

1. **Condition 1:** A = 1, B = 0, C = 0 (Temperature High, Humidity Low)
2. **Condition 2:** D = 1 (System Active)
3. **Condition 3:** B = 1, C = 1 (Temperature Low, Humidity High)

**A B C D      Y (System Active)**

0    0    0    0    0

0    0    0    1    1

0    0    1    0    0

0    0    1    1    1

0    1    0    0    0

0    1    0    1    1

0    1    1    0    1

0    1    1    1    1

1    0    0    0    1

1    0    0    1    1

1    0    1    0    0

1    0    1    1    1

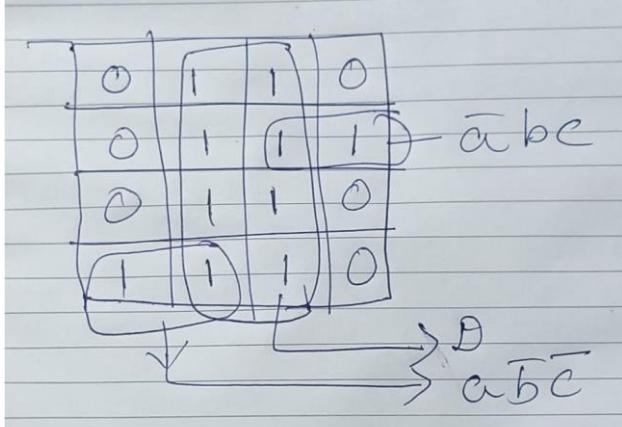
1    1    0    0    0

1    1    0    1    1

1    1    1    0    0

1    1    1    1    1

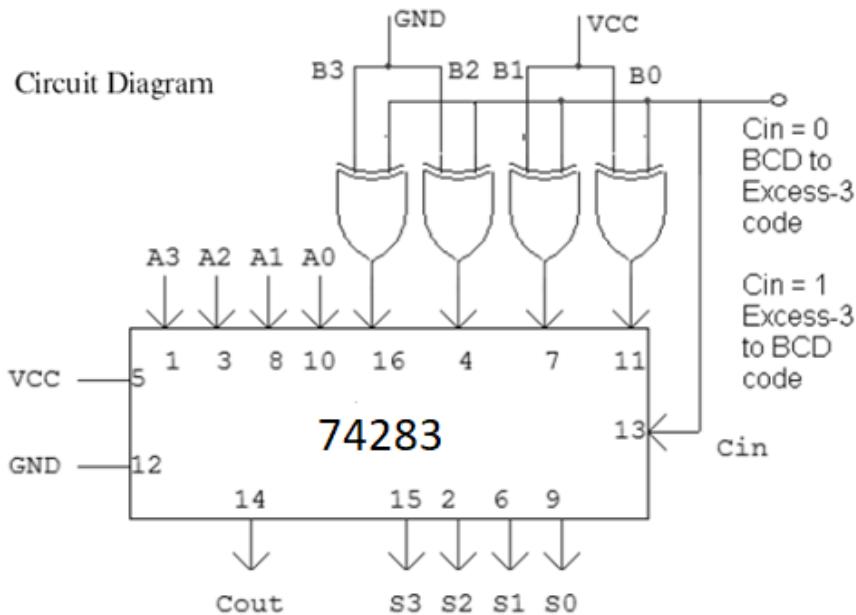
$$Y = D + AB'C' + A'BC$$



- b Design a single circuit for BCD to excess 3 and excess 3 to BCD converter using 4 bit parallel adder and appropriate gates)

BCD	EXCESS-3	EXCESS-3	BCD
0 0 0 0	0 0 1 1	0 0 1 1	0 0 0 0
0 0 0 1	0 1 0 0	0 1 0 0	0 0 0 1
0 0 1 0	0 1 0 1	0 1 0 1	0 0 1 0
0 0 1 1	0 1 1 0	0 1 1 0	0 0 1 1
0 1 0 0	0 1 1 1	0 1 1 1	0 1 0 0
0 1 0 1	1 0 0 0	1 0 0 0	0 1 0 1
0 1 1 0	1 0 0 1	1 0 0 1	1 0 0 0
0 1 1 1	1 0 1 0	1 0 1 0	1 0 0 1
1 0 0 0	1 0 1 1	1 0 1 1	1 0 0 0
1 0 0 1	1 1 0 0	1 1 0 0	1 0 0 1

Truth table -2 marks, circuit with brief explanation-4 marks



- 2a Using Modified Booth's multiplication algorithm, multiply +9 with -6. (Note: +9 is the multiplicand and (-6) is the multiplier)

6 L3 CO2

4 L3 CO1

2a)

$$\begin{array}{r}
 +9 \rightarrow 01001 \rightarrow 10111 \\
 -6 \rightarrow 00110 \\
 \hline
 11001 \\
 \hline
 11010 \rightarrow -6
 \end{array}$$

0-1+1-10 - recoding of bits

$$\begin{array}{r}
 11010 \\
 \hline
 10111 \\
 \hline
 10111
 \end{array}$$

bitpair recoding

$$\begin{array}{r}
 01001 \times 0-1-2 \\
 \hline
 111101110 \\
 111101110xx \\
 0000000xxxx \\
 \hline
 11111001010 \rightarrow -54 \\
 110101 \\
 \hline
 110110 \rightarrow 54
 \end{array}$$

- b Illustrate the design of a single decade BCD adder(decimal adder) and show the truth table

### Comparing binary and BCDsums

Decimal Sum	Binary sum					Required BCD sum				
	K	P <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>	C <sub>out</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

=2m

		$P_1 P_0$			
		00	01	11	10
		00	01	00	00
		01	0	0	0
		11	1	1	1
		10	0	1	1
$P_3 P_2$					
00	0	0	0	0	0
01	0	0	0	0	0
11	1	1	1	1	1
10	0	0	1	1	1

KMap to detect the combinations

**1010, 1011, 1100, 1101, 1110, 1111**

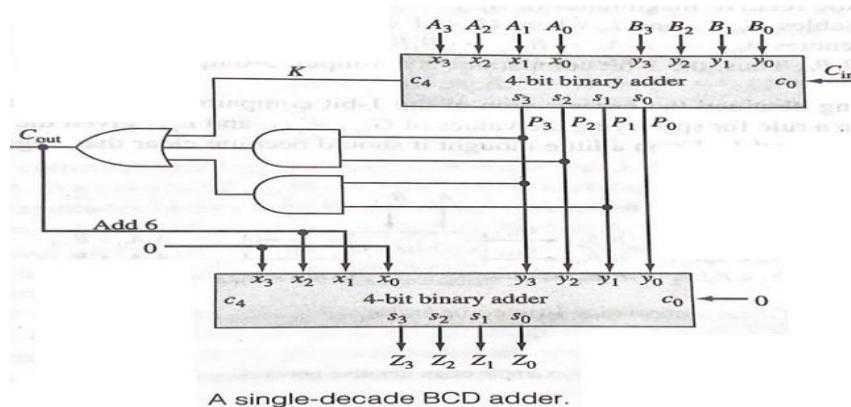
from the Kmap

$$\text{ADD}_6 = P_3 P_2 + P_3 P_1$$

Including the Carry output (K), in the expression, the Boolean expression for adding the correction is

$$\text{ADD\_6} = K + P_3 P_2 + P_3 P_1$$

=2m



=2m

3 Explain the process for calculating carry in a Carry Look Ahead adder, design a four bit carry look ahead adder(derive the equations) How does propagation delay in a Carry Look ahead Adder compare to that of a parallel adder?

$$S_i = X_i \cdot \text{xor. } Y_i \cdot \text{xor. } C_i$$

$$\begin{aligned} C_{i+1} &= X_i Y_i + X_i C_i + Y_i C_i \\ &= X_i Y_i + C_i (X_i + Y_i) \\ &= X_i Y_i + C_i (X_i \cdot \text{or. } Y_i) \end{aligned}$$

Let us represent,  $X_i Y_i$  by  $G_i$  (also called as Carry generate function) &  $X_i \cdot \text{or. } Y_i$  by  $P_i$  (also called carry propagate function)

$$= G_i + C_i P_i$$

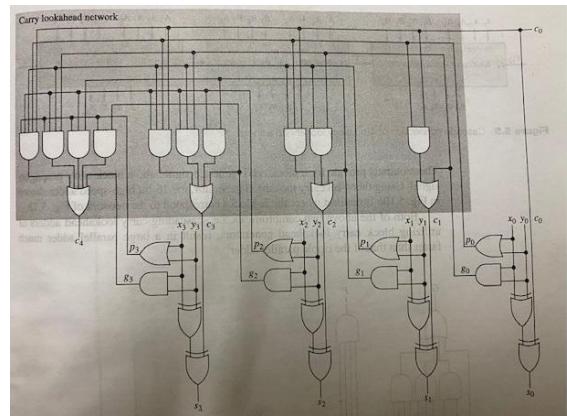
$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

**P0 C0 = 5 marks**



carry look ahead adder for 4 bit operands is shown in the figure. Generalizing from the figure, The path length from the generation of carry to its appearance as an input at any higher order stage, i.e the path length through any stage of the carry look ahead network, is two levels of logic.

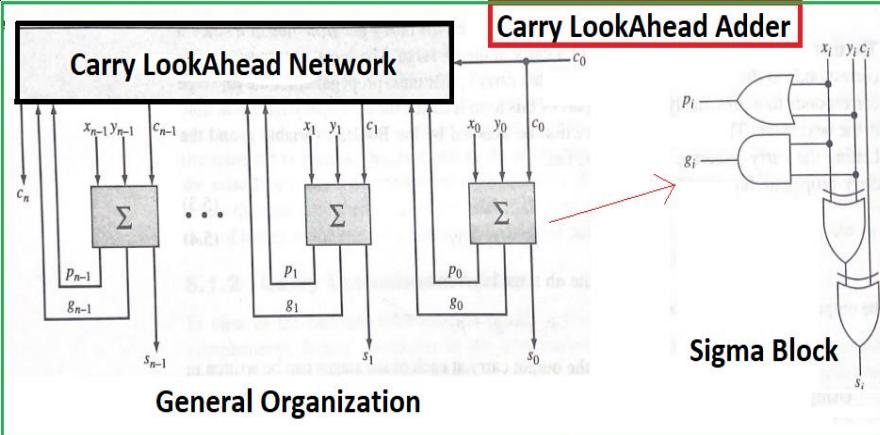
Thus with one level of logic to from  $G_i$ , two levels of logic for the carry to propagate between any two stages, and one level of logic to have the carry effect a sum output, the maximum propagation delay for CLA is 4 Units of time, assuming each gate introduces a unit of time of propagation delay.

<- 4Bit CLA Design

Circuit-3marks Propagation delay comparision-2 marks

Or

10 L3 CO2



- 4 A home security system has four sensors: A (Garage Door Open), B (Front Door Open), C (Back Door Open), and D (Alarm Active). The alarm activates in the following cases: If any door is open (at least one of A,B, or C is 1). or If the alarm is already active (D=1).

Consider for both if conditions also

**Task:** Develop a truth table, and using Quine Mc-cluskey method, obtain the essential prime implicants for generating the alarm.

To create a truth table for the home security system based on the revised conditions, we need to clarify how the alarm activates under the specified conditions: Inputs:

- A: Garage Door Open (1 if open, 0 if closed)
  - B: Front Door Open (1 if open, 0 if closed)
  - C: Back Door Open (1 if open, 0 if closed)
  - D: Alarm Active (1 if active, 0 if inactive) Conditions for Alarm Activation: The alarm activates if:
    1. At least one door is open: ( A or B or C) (true if any of A, B, or C is 1)
    2. Or if the alarm is already active: ( D = 1)
- Final Logical Expression:

Thus, the new alarm state (  $D = (A \text{ or } B \text{ or } C) \text{ or } D$  )

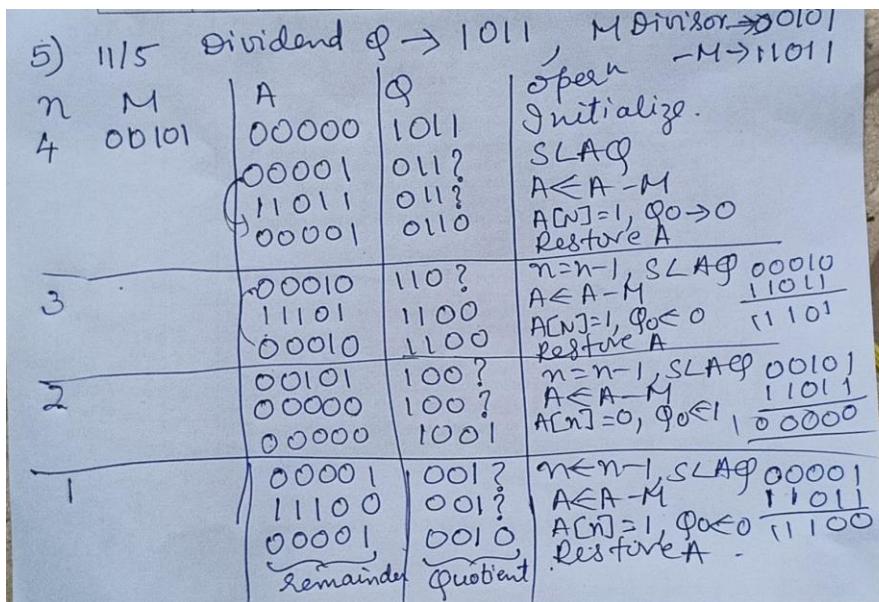
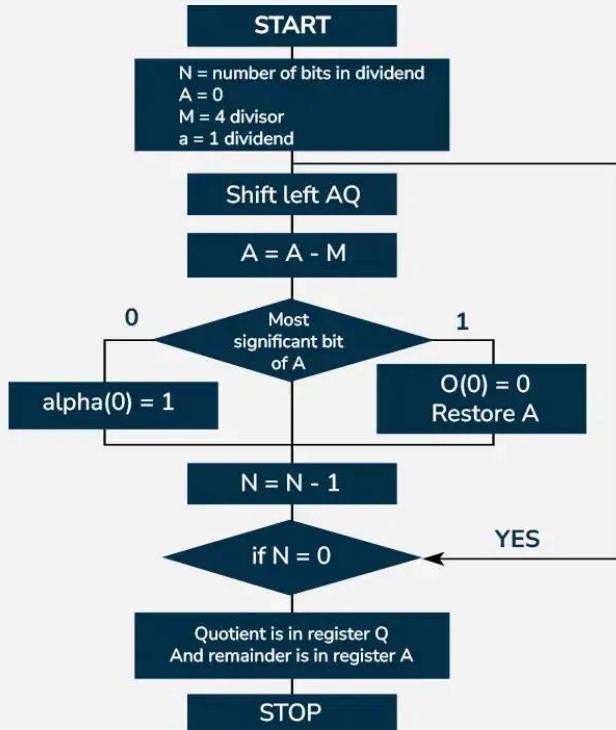
This means that the alarm will be active if:

- Any door (A, B, or C) is open
- The alarm is already active

10 L4 CO2

	<p>Truth Table:</p> <table border="1"> <thead> <tr> <th>A</th><th>B</th><th>C</th><th>D</th><th>Dactive</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>System activates when A+B+C+D is active</p> <p>For And logic=<math>m(3,5,7,9,11,13,15)</math> System activates when A+B+C+D is active</p> <p><math>ad+bd+cd</math></p> <p>  A   B   C   D   Dactive</p>	A	B	C	D	Dactive	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1		
A	B	C	D	Dactive																																																																																				
0	0	0	0	0																																																																																				
0	0	0	1	1																																																																																				
0	0	1	0	1																																																																																				
0	0	1	1	1																																																																																				
0	1	0	0	1																																																																																				
0	1	0	1	1																																																																																				
0	1	1	0	1																																																																																				
0	1	1	1	1																																																																																				
1	0	0	0	1																																																																																				
1	0	0	1	1																																																																																				
1	0	1	0	1																																																																																				
1	0	1	1	1																																																																																				
1	1	0	0	1																																																																																				
1	1	0	1	1																																																																																				
1	1	1	0	1																																																																																				
1	1	1	1	1																																																																																				

	--- --- --- --- -----			
	0 0 0 0  0			
	0 0 0 1  1			
	0 0 1 0  1			
	0 0 1 1  1			
	0 1 0 0  1			
	0 1 0 1  1			
	0 1 1 0  1			
	0 1 1 1  1			
	1 0 0 0  1			
	1 0 0 1  1			
	1 0 1 0  1			
	1 0 1 1  1			
	1 1 0 0  1			
	1 1 0 1  1			
	1 1 1 0  1			
	1 1 1 1  1			
5	Draw the flow chart for restoring division algorithm and perform the division operation on the following unsigned integers:Dividend:11 Divisor:5	10	L3	CO1



Flow chart 4+ solving 6 marks

Course Outcomes: After completing the course, the students will be able to:-	
<b>CO 1</b>	Apply design requirements for digital systems and Computer organization
<b>CO 2</b>	Analyze the models used for designing various Combinational and Sequential circuits
<b>CO 3</b>	Develop applications of synchronous sequential networks using flip flops, registers and counters
<b>CO 4</b>	Design optimized modern processors and memories for given specifications

<b>CO 5</b>	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools										
-------------	--	--	--	--	--	--	--	--	--	--	--

Marks Distribution	Particulars		CO1	CO2	CO 3	CO4	CO5	L1	L2	L3	L 4	L 5	L6
	Test	Max Mark s	18	42					16	34	10		



**R V College of Engineering**  
**Department of Computer Science and Engineering(data science and cyber security)**  
**ADLDCO CIE - II : Question Paper**

Course: (Code)CS234 AI	Applied digital logic design and computer organisation	Semester: 3 <sup>rd</sup> BE
Date : Dec 2024	Duration : 120 minutes	Staff :DD/PSB/KS/KBR/KB
Name :	USN :	Section: A/B/C/D/E/CD/CY

**PART A**

Sl. No	Quiz questions	Mark s	* L1- L6	*CO
1.1	Develop a logic circuit to implement the functionality of an XOR gate utilizing a 2:1 multiplexer, demonstrating the integration of combinational logic with minimal hardware resources.	1	L3	CO2
1.2	Calculate the maximum achievable operating frequency of a MOD-32 ripple counter, given that the propagation delay per flip-flop is 50 ns, while considering the cumulative effect of sequential flip-flop delays on the circuit's performance.	1	L3	CO3
1.3	What techniques can be employed to mitigate the issue of "0" and "1" catching in digital circuits, ensuring reliable signal integrity and robust operation?	1	L3	CO3
1.4	Derive the Characteristic Equation for JK Flip-Flop	1	L2	CO3
1.5	A D-Flip Flop can be converted into T-Flip Flop by using additional logic circuit, Obtain the Boolean expression for additional logic circuit	1	L3	CO3
1.6	Why are asynchronous inputs referred to as overriding inputs, and how do they directly affect the state of a flip-flop irrespective of clock signals?	1	L3	CO3
1.7	Draw the circuit diagram of a 4:16 Decoder using 3:8 decoder	2	L2	CO3
1.8	Draw the Logic diagram for Master-Slave Flip-Flop with Data Lockout and How can a data lock-out flip-flop be symbolically distinguished from a pulse-triggered flip-flop in digital circuit representations?	2	L2	CO3

**PART B**

Sl. No	Answer all the questions	Mark s	* L1- L6	*CO
1a	Design and implement an asynchronous Mod-6 up counter, detailing the circuit configuration, flip-flop connections, and state transitions, and provide the corresponding clock timing diagram that illustrates the sequential counting behavior	5	L3	CO3
b	Elucidate the operational principles of a synchronous 2N twisted ring counter, detailing the sequential state	5	L3	CO3

	transitions, feedback mechanism, and the role of flip-flops in generating the desired counting sequence			
2a	Design a digital logic circuit to implement the Boolean functions $F1(a,b,c)=\pi M(0,3,5,6,7)$ and $F2(a,b,c)=\pi M(2,3,4,5,7)$ using a 3-to-8 decoder, ensuring minimal usage of input gates. Provide a step-by-step analysis of how the decoder outputs can be utilized to construct the functions efficiently, while optimizing the number of gates used for combining the relevant terms.	5	L3	CO2
b	Design and implement the Boolean function $f(A,B,C,D)=\sum m(0,1,2,4,6,9,12,14)$ using a 4-to-1 multiplexer. Provide a detailed explanation of how to map the min terms to the MUX inputs, and outline the necessary logic for selecting the appropriate input lines based on the function's truth table.	5	L3	CO2
3a	Design a 4bit Universal shift register for the operations specified in the table  S1 S0 Register Operation 0 0 No Change 0 1 Shift Right 1 0 Shift Left 1 1 Load the parallel data	5	L2	CO3
b	Detail the working of 1:4 Demultiplexer and with this, Design and implement a 1:8 demultiplexer utilizing two 1:4 demultiplexers, ensuring correct signal routing through hierarchical selection lines. Provide a detailed circuit configuration that demonstrates how to cascade the two 1:4 demultiplexers, managing the selection logic to achieve the required output channels.	5	L3	CO2
4a	Explain the phenomenon of a race-around condition in sequential circuits, particularly in JK flip-flops, and describe how this issue is mitigated using a master-slave JK flip-flop configuration. Additionally, illustrate the use of a gated SR latch to control the input signals, providing a detailed logic diagram and timing waveforms to demonstrate the elimination of the race-around condition in the master-slave configuration.	5	L3	CO3
b	Propose the block diagram to realize a 4-bit Comparator, using 1 bit comparator with iterative approach, with all the connections and indicating clearly the meaning of the inputs and outputs. Demonstrate, the working of the circuit with example data of 4 bit numbers. (No truth table and derivations required)	5	L3	CO3
5a	Design a synchronous counter using positive edge triggered JK flip flops to count $0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0 \rightarrow 1 \rightarrow 2 \dots$	5	L2	CO3
b	Design and implement an edge-triggered D flip-flop, detailing the circuit configuration, input-output	5	L2	CO3

	relationships, and the control mechanisms that ensure synchronization with the clock signal.			
--	--	--	--	--

<b>Course Outcomes: After completing the course, the students will be able to:-</b>	
<b>CO 1</b>	Apply design requirements for digital systems and Computer organization
<b>CO 2</b>	Analyze the models used for designing various Combinational and Sequential circuits
<b>CO 3</b>	Develop applications of synchronous sequential networks using flip flops, registers and counters
<b>CO 4</b>	Design optimized modern processors and memories for given specifications
<b>CO 5</b>	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test	Max Marks		16	44			20	40				

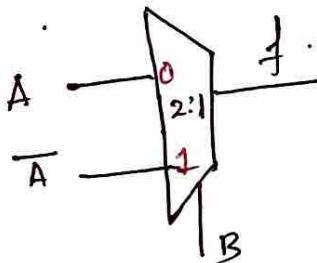
Date: 03-12-2024

ADLD & CO  
CIE-II Scheme and Solution

PART-A QUIZ

1.1

NOR gate using 2:1 MUX.



① Marks.

1.2

MOD 32 need 5 flip flop  $\therefore 2^5 = 32$

$$f_{MAC} = \frac{1}{50\text{ns} \times 5} = 4 \text{MHz}$$

1.3 0's and 1's catching problem can be resolved with the help of Edge Triggered D-Flipflop: this flipflop consider the Data at the time of Edge and it won't consider the entire clock level. ① Marks.

1.4

JK flip flop equation

$$Q^+ = J\bar{Q} + \bar{K}Q$$

① Marks.

1.5

$$D = \bar{T}Q + T\bar{Q}$$

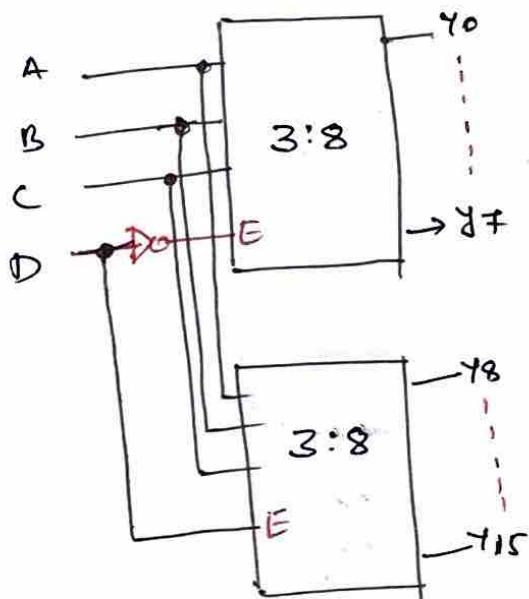
$$= T \oplus Q$$

① Marks.

1.6 Asynchronous inputs are directly connected to a bistable element; hence these inputs are overriding other inputs and also impact the state of a flip flop irrespective of clock signal.

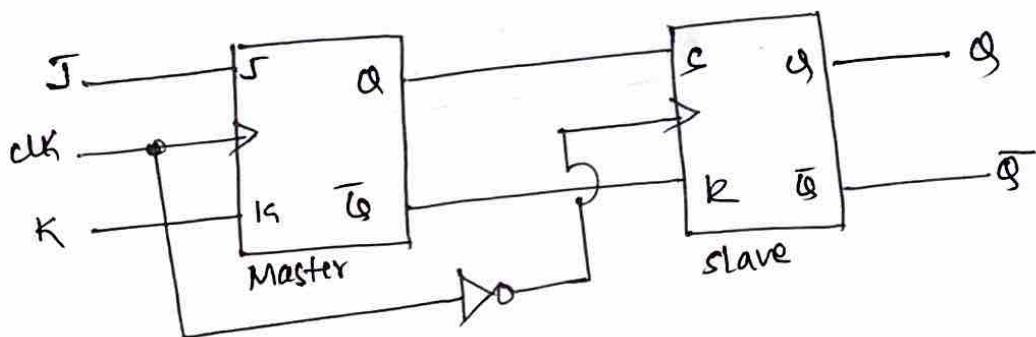
→ (1) Marks

1.7

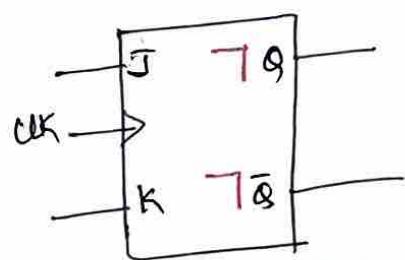


→ (2) Marks

1.8



→ (2) Marks



Data lockout representation

PART-B TEST

1.a MOD6 up counter (Asynchronous counter)

$$2^n \geq N$$

$$2^3 \geq 6$$

$2^3 \geq 6 \therefore$  we need 3 flip flops

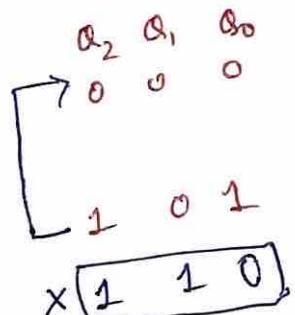
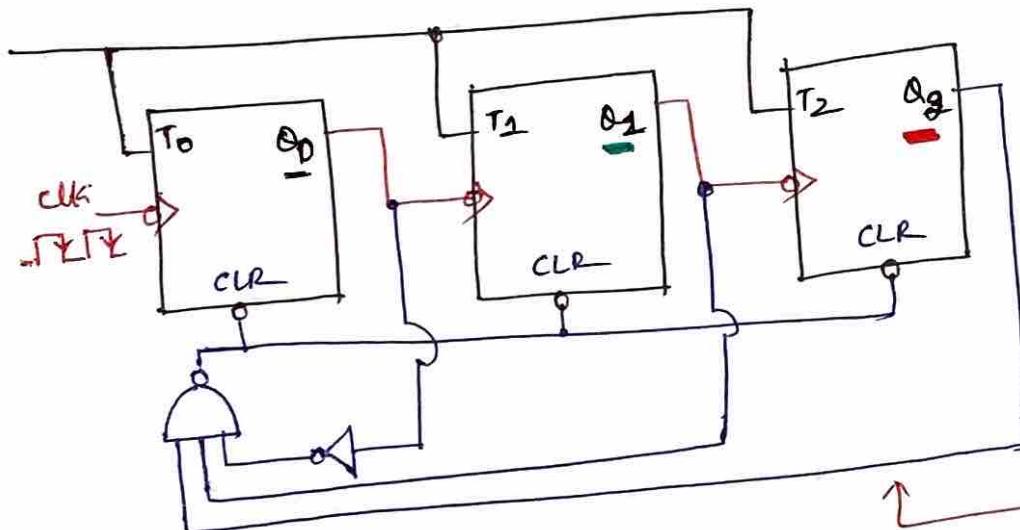
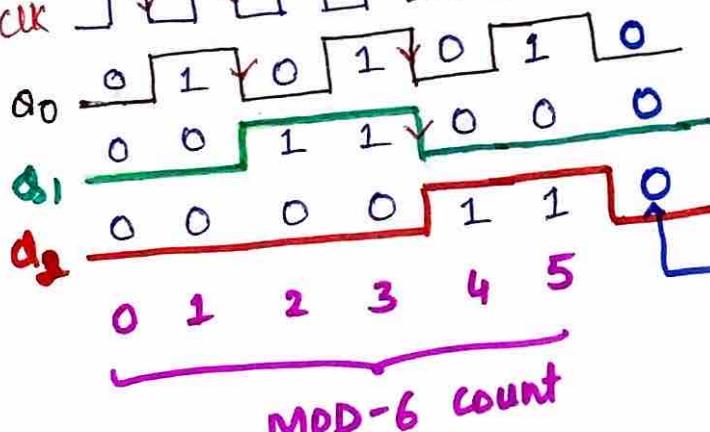
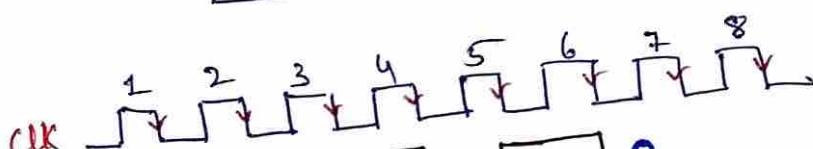


Diagram - ② marks.



waveform - ⑤ marks

CLR signal takes place.

1.b

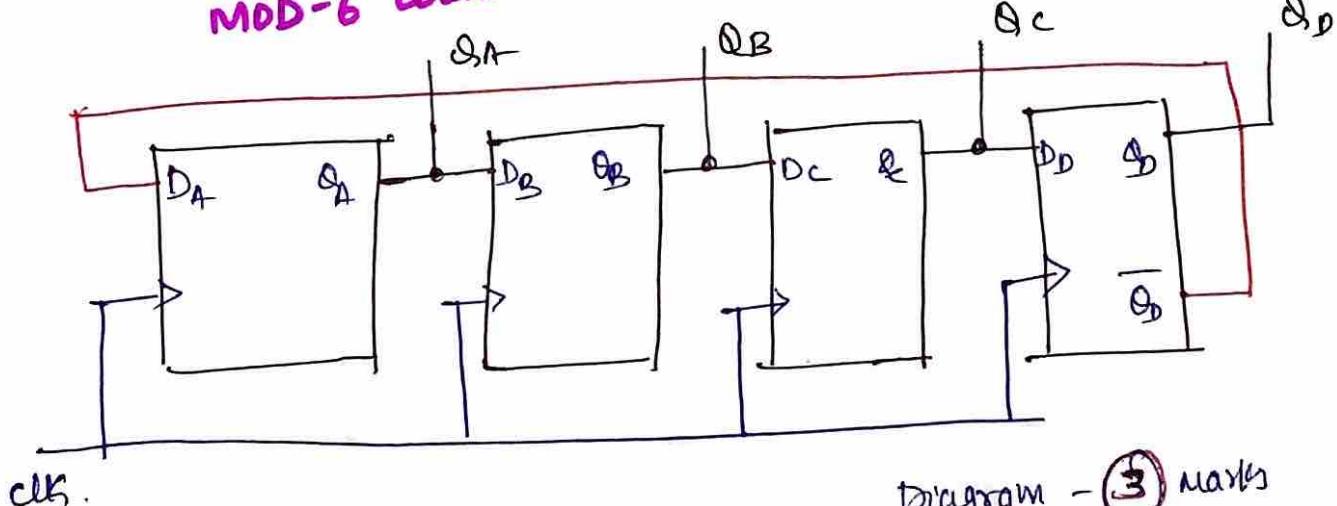


Diagram - ③ marks

clk	$Q_A$	$Q_B$	$Q_C$	$Q_D$	Initial state
-	0	0	0	0	← Initial state
①	1	0	0	0	
②	1	1	0	0	
③	1	1	1	0	
④	1	1	1	1	
⑤	0	1	1	1	
⑥	0	0	1	1	
⑦	0	0	0	1	
⑧	0	0	0	0	← Initial state

N bit Johnson counter

$$\text{so } N = 4 \text{ :-}$$

$$2N = 2^4 = 8 \text{ counts.}$$

Sequence table  
Marks.

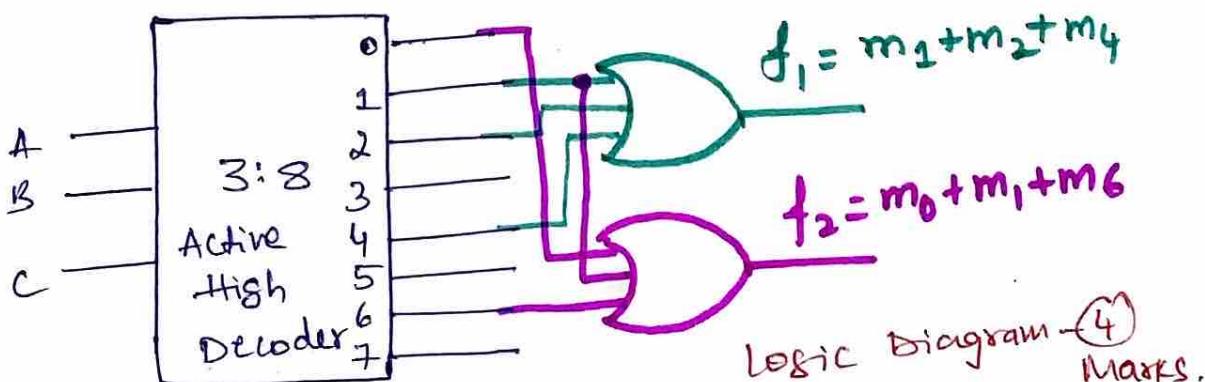
clk	$Q_A$	$Q_B$	$Q_C$	$Q_D$	Initial state
-	0	0	0	0	← Initial state
①	1	0	0	0	
②	1	1	0	0	
③	1	1	1	0	
④	1	1	1	1	
⑤	0	1	1	1	
⑥	0	0	1	1	
⑦	0	0	0	1	
⑧	0	0	0	0	← Initial state

2. a Implement the following function using 3:8 decoder & minimal usage of input gates.

$$F_1(a, b, c) = \overline{\Sigma M(0, 3, 5, 6, 7)} \Rightarrow \Sigma_m(1, 2, 4)$$

$$F_2(a, b, c) = \overline{\Sigma M(2, 3, 4, 5, 7)} \Rightarrow \Sigma_m(0, 1, 6)$$

① mark

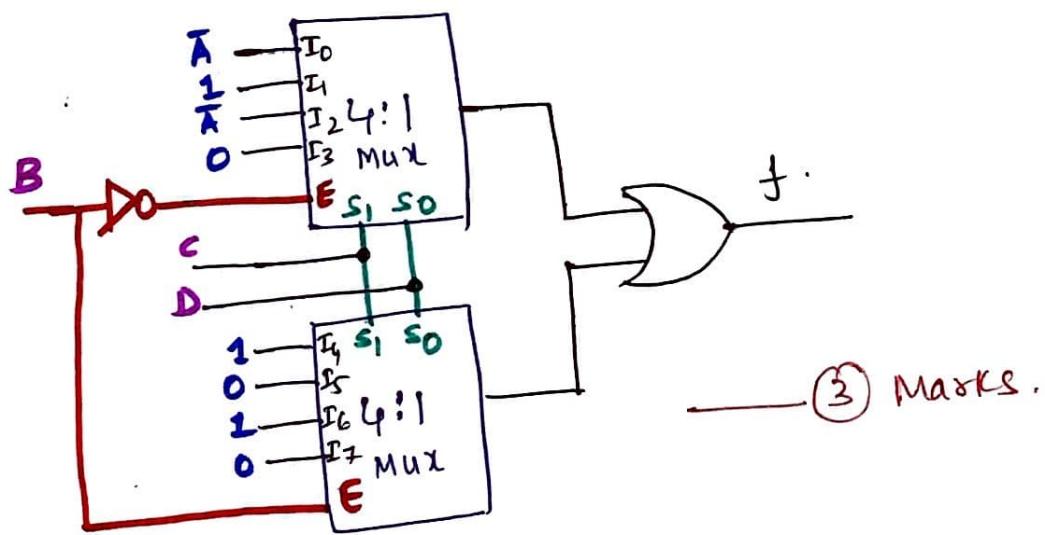


Note: To achieve minimal usage of input gates, first we need to convert POS form to SOP form. Then realize the equation using Active-High Decoder & OR gate.

2.b  $F(A, B, C, D) = \sum_m (0, 1, 2, 4, 6, 9, 12, 14)$  using 4:1 mux.

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	$\bar{A}$	1	$\bar{A}$	0	1	0	1	0

(2) Marks

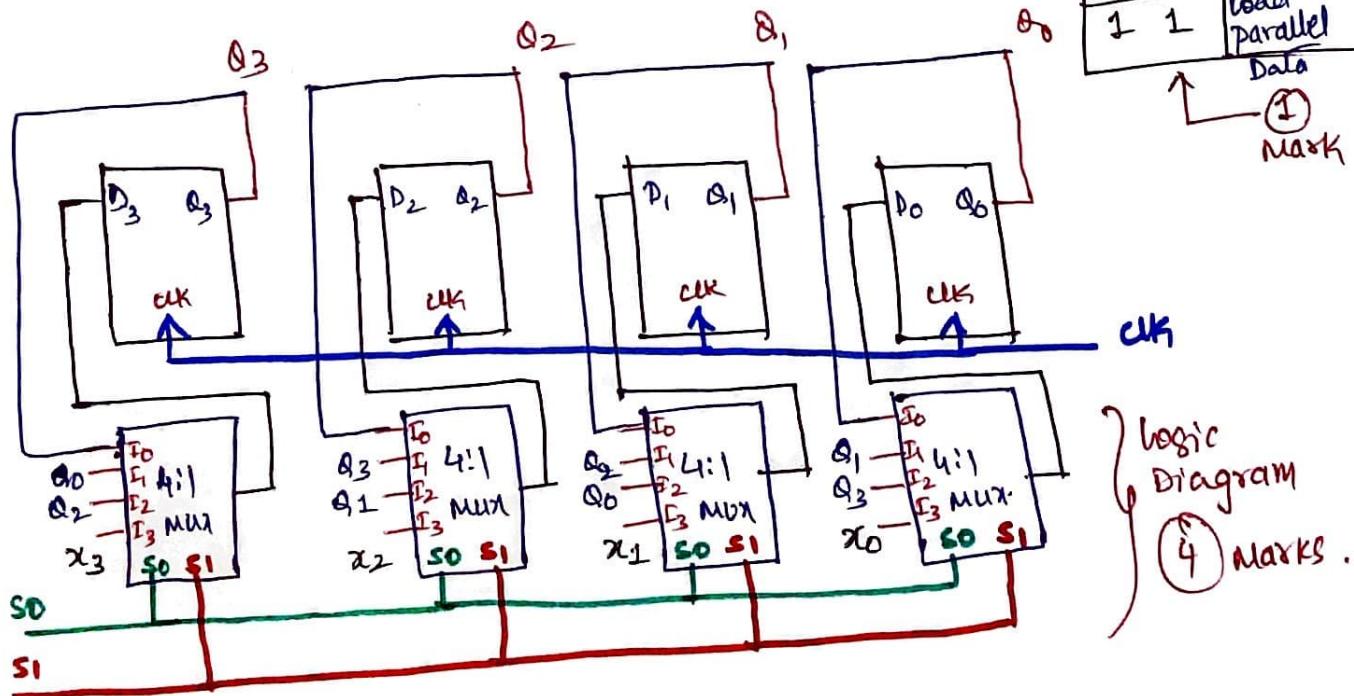


A ← for Mapping

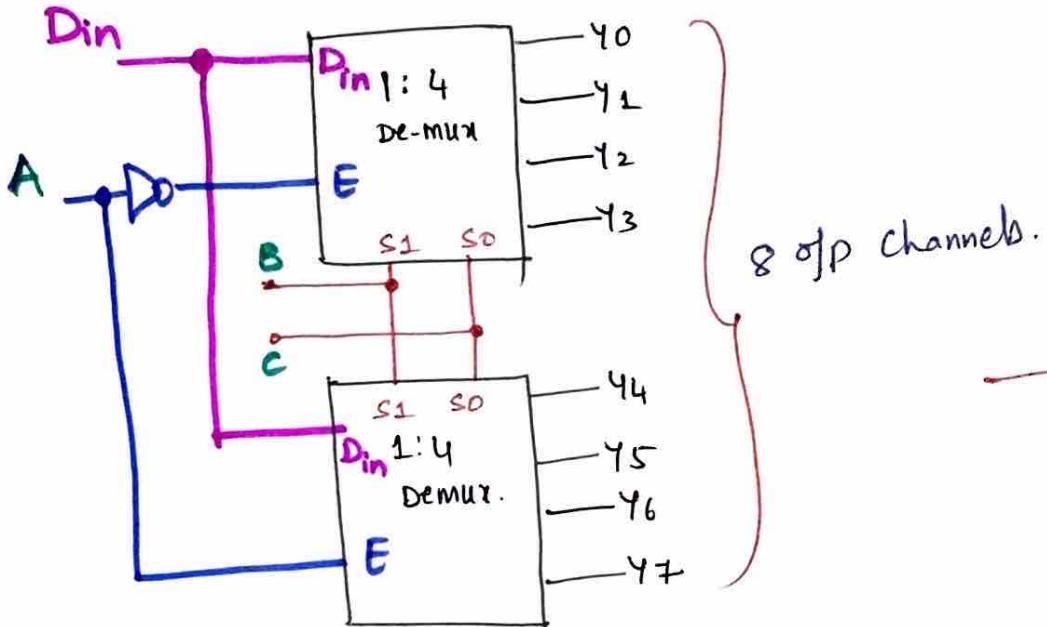
B ← to activate the Enable pin

C & D ← used for select line.

3.a



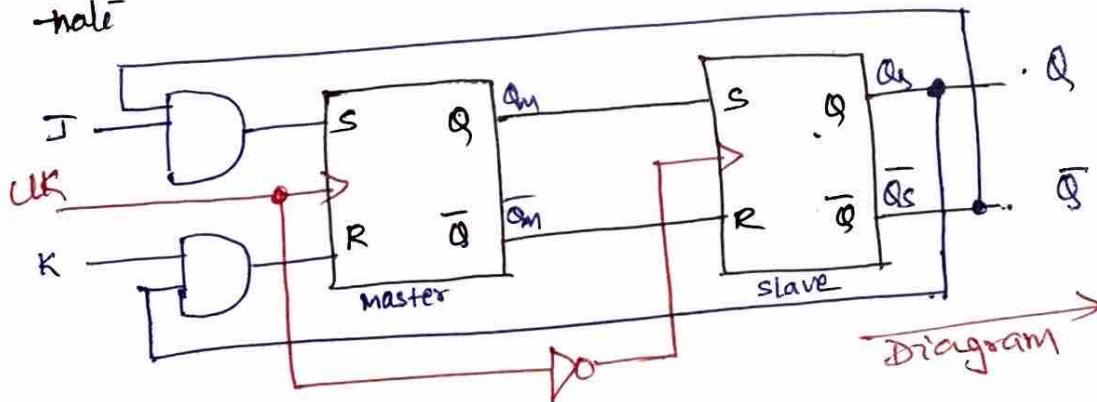
3.b



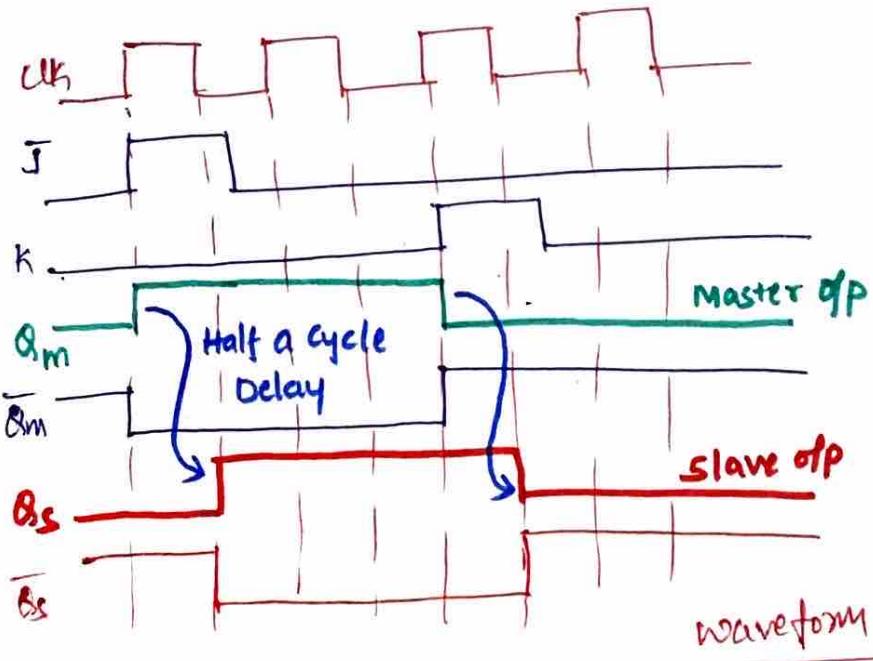
(5) Marks.

A. a

Race around means continuous toggling. The clock pulse is HIGH for a longer duration than the flip-flop propagation delay, this causes the output to toggle continuously within the same clock, making it indeterminate state.



(2) Marks

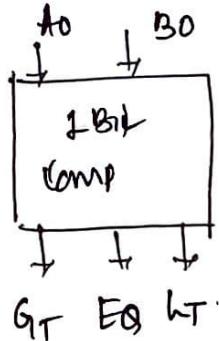


clk	J	K	Q	Q-bar	Status
1	0	0	0	1	No change
1	0	1	0	1	Reset-set
1	1	0	1	0	Toggle
1	1	1	0	1	No change
0	x	x	Q	Q-bar	

Truth table (1) Marks

(2) Marks

#### 4.b) 1 Bit comparator

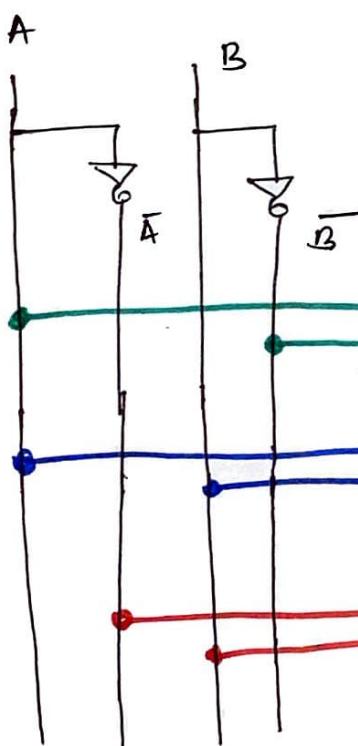


A <sub>0</sub>	B <sub>0</sub>	GT	EQ	LT
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$GT = A \bar{B}$$

$$EQ = \bar{A}B + AB \\ = (\overline{A \oplus B})$$

$$LT = \bar{A}B$$

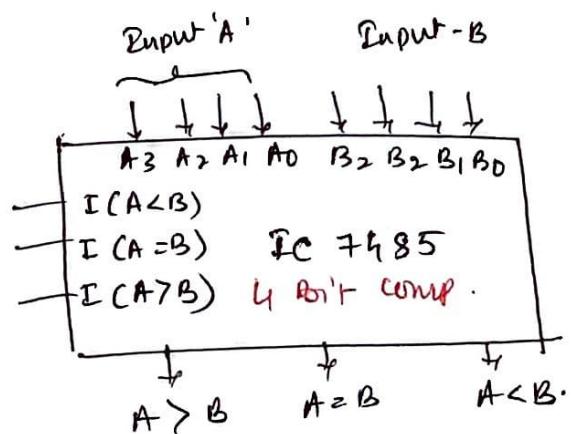


1 mark

$$EQ = \bar{A}B + AB \\ = (\overline{A \oplus B})$$

$$LT = \bar{A}B$$

- 2 marks



1 mark.

4 bit Example NO :- ① mark

Total = 5 marks.

## 5.1 Sequence generator.

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0 \rightarrow 1 \rightarrow 2$$

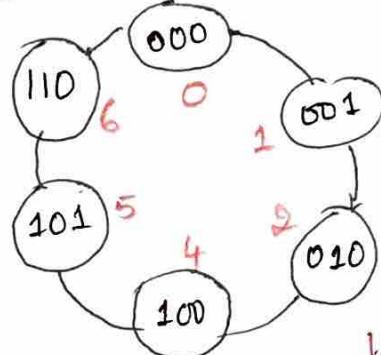
6 → is the highest count ∴ we need 3 flip flop to construct.  
 In above sequence.

JK Excitation table

$Q$	$Q^+$	$J$	$K$
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

→ 1/2 mark

state diagram



1/2 mark

State table

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	1	0	0	1	x	x	x	1	x
0	1	1	x	x	x	x	x	x	x	x	1
1	0	0	1	0	1	x	0	0	x	0	x
1	0	1	1	1	0	x	0	1	x	x	x
1	1	0	0	0	0	x	1	x	1	x	x
1	1	1	x	x	x	x	x	x	x	x	x

1 1/2 marks

$J_2$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	0	0	x	1
1	x	x	x	x

$K_2$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	x	x	x	x
1	0	0	x	1

$J_1$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	0	1	x	x
1	1	0	x	x

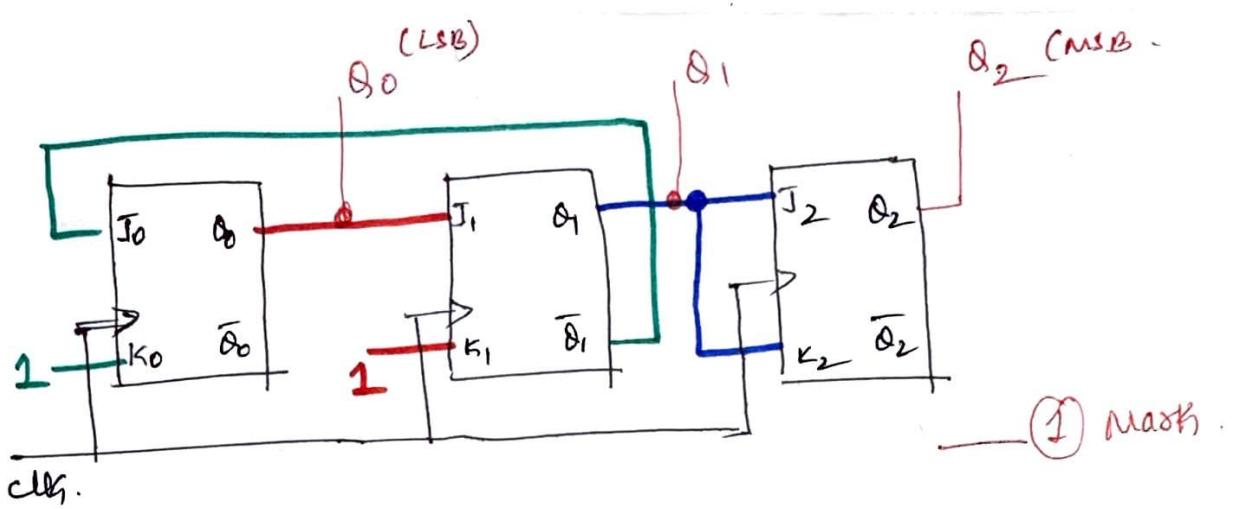
$J_1$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	0	1	x	x
1	1	0	x	x

$K_1$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	x	x	*	*
1	x	x	x	1

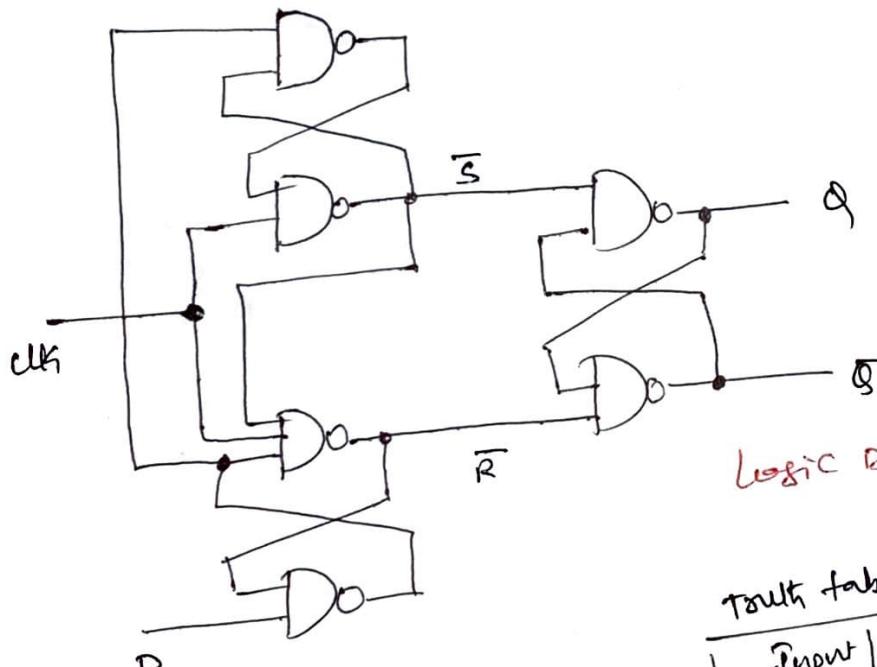
$J_0$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	1	x	x	0
1	1	x	x	0

$K_0$	$Q_0^0$	$Q_0^1$	$Q_0^2$	$Q_0^3$
0	x	1	x	x
1	x	1	x	x

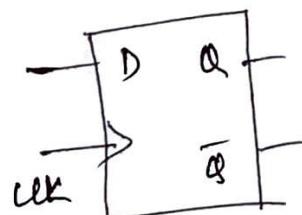
1 1/2 marks



5.b



Logic diagram — (1) Marks



Truth table.

Input		Output	
clk	D	Q	$\bar{Q}$
1	0	0	1
1	1	1	0
0	x	Q	$\bar{Q}$

— (1) Mark.



ADLD & CO Improvement Test & Quiz : Question Paper

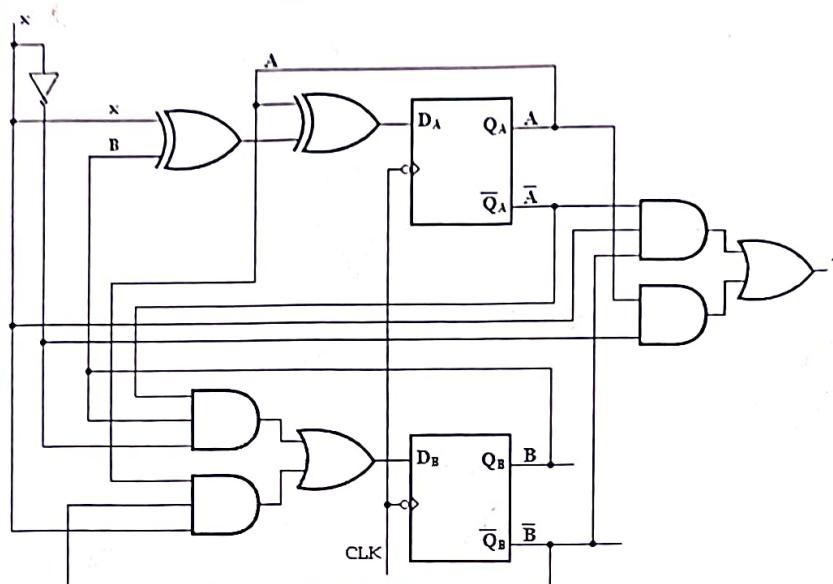
<b>Course Code</b> <b>CS234AI</b>	<b>Course Name: Applied Digital Logic Design and Computer Organisation</b>	Semester : <b>3<sup>rd</sup> BE</b>
Date : 7 <sup>th</sup> Jan 2025	Duration : 120 minutes	Staff : DD/PSB/KS/KBR/KB
Name :	USN :	Section : A/B/C/D/E/CD/CY

**Part-A**

Sl. no	Answer all the questions	M	L1- L6	CO
1	Program contains 500 instructions, out of that 40% of instructions requires 1 clock cycles, in the remaining 60%, 50 instructions require 2 clock cycles and the remaining requires 3 clock cycles for execution. Find the total time required to execute the program running on a 1 MHz machine.	2	L2	CO4
2	Draw the block diagram of the Processor/CPU design with Three Bus Datapath.	2	L3	CO4
3	Given program perform the addition of two numbers and indicate the value of the flags CY, Z, OF and N (Note: H means, number is represented in Hex) MOVE #99H, R0 MOVE #8AH, R1 ADD R0, R1, R2 ; R0 + R1 -> R2	2	L3	CO4
4	The shift register shown in the figure is initially loaded with the bit pattern 1010. subsequently the shift register is clocked, and with each clock pulse the pattern gets shifted by one bit position to the right. With each shift, the bit at the serial input is pushed to the left most position (MSB). The number of clock pulses required to have the content of the shift register 1010 again is _____	2	L4	CO5
5	Register R1 and R2 of a computer contain the decimal values 3500 and 2800. Find the effective address of the memory operand in the following instructions ADD +(R1), R5 ADD (R1), R5	2	L3	CO4

r 3500  
 2800

### Part B

<p>1 Analyze the following circuit and write the excitation equations, derive transition equations, compute excitation table, transition table and draw state diagram. Is this a <u>mealy</u> or a <u>moore</u> model and why?</p> 	<p>10 L4 CO5</p>
<p>2 Design and implement a <u>serial adder</u> using the Moore-type finite state machine (FSM) model, incorporating state transitions, output logic, and input-output synchronization to achieve sequential bit-wise addition of binary numbers.</p>	<p>10 L4 CO5</p>
<p>3 a. Discuss all the addressing Modes supported by the Processor instruction set, with an example instruction. b. Compare processor instructions and assembler directives. Indicate meaning of EQU, DATAWORD, ORG.</p>	<p>6 4 L3 CO4</p>
<p>4 a. Write Assembly Language Code snippets using <u>Three</u> and <u>One</u> address instructions to evaluate the following expression with suitable comments. <math>Y = X^3 + 24X - 45</math> b. Demonstrate with Machine Language / ALP code snippet, using suitable memory layouts (stack frame), the passing and returning of arguments to the subroutine using the stack.</p>	<p>6 4 L4 CO4</p>
<p>5 Write ALP code snippets, for the following scenarios, with clear memory layout (Assume input numbers and the results are stored in the memory, Processor supports direct/indirect operands and two address formats, all the arithmetic and data transfer instructions are supported) a. To find Sum and Average of 'N' 32 bit numbers. b. To find Smallest of two 32 bit numbers</p>	<p>10 L3 CO4</p>

**Course Outcomes:**

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test/Quiz	Marks				38	22		2	36	22		



ADLD & CO Improvement Test & Quiz : Question Paper

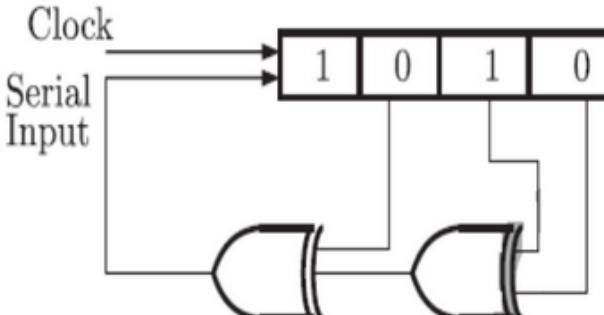
<b>Course Code</b> CS234AI	<b>Course Name:</b> Applied Digital Logic Design and Computer Organisation	Semester : 3 <sup>rd</sup> BE
-------------------------------	--	-------------------------------

Date : 7<sup>th</sup> Jan 2025 Duration : 120 minutes Staff : DD/PSB/KS/KBR/KB

Name : USN : Section : A/B/C/D/E/CD/CY

**Part-A**

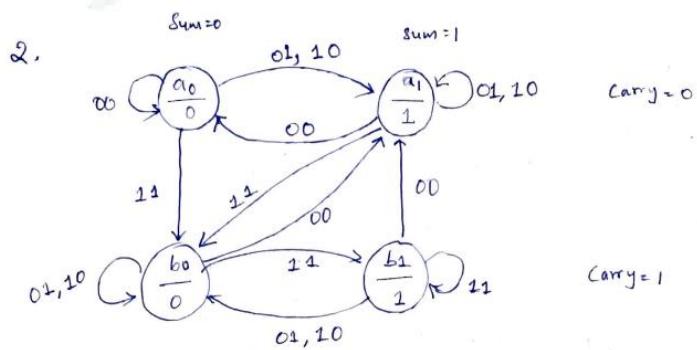
Sl. no	Answer all the questions	M	L1- L6	CO
1	<p>Program contains 500 instructions, out of that 40% of instructions requires 1 clock cycles, in the remaining 60%, 50 instructions require 2 clock cycles and the remaining requires 3 clock cycles for execution. Find the total time required to execute the program running on a 1 MHz machine.</p> <p>(40% of 500= 200 x 1 clk cycle = 200 clks Among 60% of 500, i.e 300, = 50 x 2 + 250 x 3 = 850 clks Total clocks = 1050 clks, Time for 1clk = 1/1MHz = 1MicroSecond,)</p> <p><b>Answer:</b> Hence Total Time = 1050 x 1Micro second = 1050 Micro seconds Note: Give Marks, for the answer represented in any units like seconds,nanoseconds,microseconds.</p>	2	L2	CO4
2	<p>Draw the block diagram of the Processor/CPU design with Three Bus Datapath.</p> <p style="text-align: center;"><b>MULTIPLE BUS CPU ORGANIZATION</b></p> <p>Note: Give Marks for Any equivalent diagram / as given in the reference book</p>	2	L3	CO4

3	<p>Given program perform the addition of two numbers and indicate the value of the flags CY, Z, OF and N (Note: H means, number is represented in Hex)</p> <pre>MOVE #99H, R0 MOVE #8AH, R1 ADD R0, R1,R2 ; R0 + R1 -&gt; R2</pre> <p><b>Assuming 8 bit processor</b></p> <p><b>99H - 1001 1001</b></p> <p><b>8AH - 1000 1010</b></p> <hr/> <p><b>0010 0011</b></p> <p><b>CY=1, Z=0,N=0,OF= Exor(1,0)=1</b></p>	2	L3	CO4																																																
4	<p>The shift register shown in the figure is initially loaded with the bit pattern 1010. subsequently the shift register is clocked, and with each clock pulse the pattern gets shifted by one bit position to the right. With each shift, the bit at the serial input is pushed to the left most position (MSB). The number of clock pulses required to have the content of the shift register 1010 again is_____</p>  <p>7 Clock Cycles:</p> <p>Serial Input = Q1.EXOR.(Q2.EXOR.Q3)</p> <table border="1" data-bbox="492 1280 857 1617"> <thead> <tr> <th>Clk</th> <th>Si/p</th> <th>Q0</th> <th>Q1</th> <th>Q2</th> <th>Q3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>3</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>4</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>5</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>6</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>7</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Clk	Si/p	Q0	Q1	Q2	Q3	1	0	1	0	1	0	2	0	0	1	1	0	3	0	0	0	1	1	4	0	0	0	0	1	5	1	1	0	0	0	6	0	0	1	0	0	7	1	1	0	1	0	2	L4	CO5
Clk	Si/p	Q0	Q1	Q2	Q3																																															
1	0	1	0	1	0																																															
2	0	0	1	1	0																																															
3	0	0	0	1	1																																															
4	0	0	0	0	1																																															
5	1	1	0	0	0																																															
6	0	0	1	0	0																																															
7	1	1	0	1	0																																															
5	<p>Register R1 and R2 of a computer contain the decimal values 3500 and 2800. Find the effective address of the memory operand in the following instructions</p> <pre>ADD +(R1), R5 ADD (R1)-, R5</pre> <p><b>ANSWER TYPE 1: (IF STUDENTS TREAT BOTH INSTRUCTIONS ARE INDEPENDENT AND NOT RELATED) AND IT IS OF 32 BIT CPU/MEM WORD EA for First Instruction: 3504</b></p> <p>(Reason- this addressing mode is of type, pre autoincrement addressing mode, here R1 is a pointer,which increments by 4 and then used for fetching the data.</p>	2	L3	CO4																																																

	<p>Hence, adds used for fetching the data is 3504 )</p> <p><b>EA for Second Instruction: 3500.</b></p> <p>(Reason-this addressing mode is of type,post autodecrement addressing mode, here first data is fetched, and then it will be incremented. So, address used for fetching is 3500.)</p> <p>(IF IT IS 8 BIT CPU/MEMORY WORD – ANS IS 3501/3500. ALWAYS TAKE DEFUALT AS 32 BIT)</p> <p><b>ANSWER TYPE 2: (IF STUDENTS TREAT BOTH INSTRUCTIONS ARE RELATED) AND IT IS OF 32 BIT CPU/MEM WORD</b></p> <p>EA FOR first instruction: 3504</p> <p>EA For second instruction:3504</p> <p>Note: Give Marks for correct answerer, reason is optional.</p>		
--	--	--	--

### Part B

1	<p>Analyze the following circuit and write the excitation equations, derive transition equations, compute excitation table, transition table and draw state diagram. Is this a mealy or a moore model and why?</p>	10	L4	CO5
2	<p>Design and implement a serial adder using the Moore-type finite state machine (FSM) model, incorporating state transitions, output logic, and input-output synchronization to achieve sequential bit-wise addition of binary numbers.</p>	10	L4	CO5



state table

Present state	New state				output
	00	01	10	11	
a0	a0	a1	a1	b0	0
a1	a0	a1	a1	b0	1
b0	a1	b0	b0	b1	0
b1	a1	b0	b0	b1	1

AB	00	01	11	10
00	0	0	1	v
01	v	0	1	0
11	0	1	1	1
10	0	1	1	1

$$D_A = AX + AY + XY$$

cont

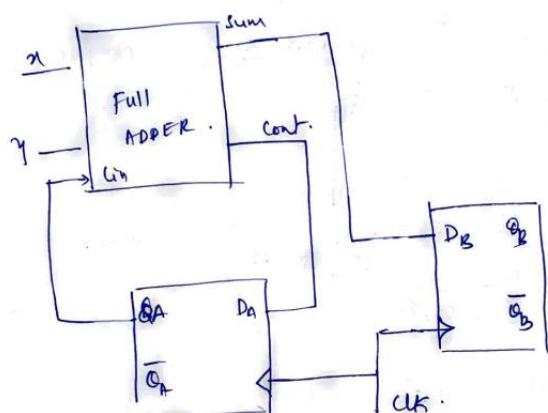
$\bar{A}$	$\bar{B}$	$Z$
0	0	1
0	1	0

AB	00	01	11	10
00	0	1	0	1
01	0	1	0	0
11	1	0	1	0
10	1	0	0	0

$$D_B = A \oplus X \oplus Y$$

sum.

$$Z = B.$$



logic diagram.

- 3 a. Discuss all the addressing Modes supported by the Processor instruction set, with an example instruction.

6 L3 CO4

Name	Example	Assembler syntax	Addressing function
Immediate	<b>EXAMPLES</b> <b>MOV #20,R0</b> move the value 20, to R0 <b>MOV R1,R0</b> move the contents of R1 to R0	#Value	Operand=Value
Register		Ri	EA=Ri
Absolute (Direct)	<b>MOV SUM,R2</b> add the number stored in memory location "SUM" to R2 and store the answer in R2	LOC	EA=LOC
Indirect	<b>ADD (R2),R0</b> add the number stored in memory location, whose address is stored in R2, with R0 and store ans in R0. R2 acts as pointer	(Ri) (LOC)	EA=[Ri] EA=[LOC]
Index	<b>MOV 4(R2),R0</b> move the contents of memory location, whose address is computed by adding 4 and contents of R2, to the register R0	X(Ri)	EA=[Ri]+X
Base with index	<b>ADD (R0,R1),R2</b> add the contents of memory location, whose address is computed by adding the contents of R0 and R1, with R2 and store the answer in R2	(Ri, Rj)	EA=[Ri]+[Rj]
Base with index and offset		X(Ri, Rj) <b>ADD 10(R0,R1),R2</b>	EA=[Ri]+[Rj]+X
Relative	<b>BNE LOOP</b> <small>[BRANCH NOT EQUAL]</small> program jumps/branches to the location, LOC, which is computed by adding the contents of PC with the offset value i.e distance of LOOP relative to PC	X(PC)	EA=[PC]+X
Autoincrement	<b>ADD (R2)+,R0</b> same as Indirect, but here pointer is incremented, after the operation	(Ri)+	EA=[Ri]; Increment Ri
Autodecrement	<b>ADD -(R2),R0</b> same as Indirect, but here pointer is decremented, before the operation	-(Ri)	Decrement Ri; EA=[Ri]

- b. Compare processor instructions and assembler directives. Indicate meaning of EQU, DATAWORD, ORG.

Answer: Refer Notes

In addition to providing a mechanism for representing instructions in a program, assembly language allows the programmer to specify other information needed to translate the source program into the object program. Such statements, called assembler directives (or commands), are used by the assembler while it translates a source program into an object program. Some popular directives are provided below.

EQU : used to name the constant values, like #define statements in C

N EQU 10

Wherever N is referred in the program, it will be replaced by 10, before assembly program is converted to object program

ORIGIN: which tells the assembler program, where in the memory to place the instructions that follow.

ORG 100

;It specifies that the instructions of the object program are to be loaded in the memory starting at address 100

DATAWORD: used to declare a variable, which can hold a word, with the initial value

N: DATAWORD 150

Value 150 is stored in the variable N, of size one word, and name of the variable is N

- 4 a. Write Assembly Language Code snippets using Three and One address instructions to evaluate the following expression with suitable comments.  $Y = X^3 + 24X - 45$

1 address instruction

```
MOVE X
MULT X
MULT X
STORE Z
MOVE X
```

6 L4 CO4

4

	<p>MULT 24 ADD Z SUB 45 STORE Y</p> <p>3 adds instruction</p> <pre>MULT X,X,Y MULT X,Y,Y MULT 24,X,Z ADD Z,Y,Y SUB 45,Y,Y</pre> <p>b. Demonstrate with Machine Language / ALP code snippet, using suitable memory layouts (stack frame), the passing and returning of arguments to the subroutine using the stack.</p> <p>Answer: Part a – any program is ok, only thing is instructions used should satisfy 3adds and 1 adds, and achieve the answer. Not required to be efficient.</p> <p>3adds: 3M 1adds: 3M</p> <p>Part B: Code snipped, to push one/multiple arguments to stack using Mov/Push, calling a subroutine (pushing return adds on stack),Optional-[save registers,access operand/argument,pop registers,save the result] Return to the main program (pop from stack).</p> <p>Code-2M,Diagram-2M</p>		
5	<p>Write ALP code snippets, for the following scenarios, with clear memory layout (Assume input numbers and the results are stored in the memory, Processor supports direct/indirect operands and two address formats, all the arithmetic and data transfer instructions are supported)</p> <ol style="list-style-type: none"> <li>To find Sum and Average of ‘N’ 32 bit numbers.</li> <li>To find Smallest of two 32 bit numbers</li> </ol> <p>Answer:</p> <p>A. Refer Notes/Ref book for adding N numbers, then use DIV instruction / successive subtraction for finding the average (let the student assume the place/reg where quotient is stored). Marks- 6Marks</p> <p>B. Use Subtraction/Comparison instruction to compare the two numbers, and use Branch. One such program is given. 4Marks</p> <pre>MOVE NUM1,R0 MOVE NUM2,R1 COMPARE R0,R1 ; OR CMP R0,R1 ; R0-R2 is done Branch &gt; 0 GREAT: MOVE R0,RESULT BRANCH SKIP</pre> <p>GREAT: MOVE R1,RESULT</p> <p>SKIP:</p> <pre>END</pre> <p>Note: Give marks for any flag, either Carry or Sign</p>	10	L3 CO4

**Course Outcomes:**

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test/Quiz	Marks				38	22		2	36	22		



**R V College of Engineering**  
 Department of Computer Science and Engineering (data science and cyber security)  
**Applied Digital Logic Design & Computer Organization**  
**Retest CIE -I : Question Paper**

Course: CS234AI	Applied digital logic design and computer organisation	Semester: 3 <sup>rd</sup> BE
Date : Dec 2024	Duration : 120 minutes	Staff : DD/PSB/KS/KBR/KB
Name :	USN :	Section: A/B/C/D/E/CD/CY

PART A

Sl. No.	Questions	M	BT	CO																		
1.1	<p>Write down the simplified Boolean expression for the following k-map</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td></td> <td style="text-align: right;">CD</td> </tr> <tr> <td></td> <td style="text-align: right;">AB</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1 0</td> </tr> <tr> <td>1</td> <td>1 1</td> <td>1 X</td> </tr> <tr> <td>0</td> <td>1 1</td> <td>1 1</td> </tr> <tr> <td>0</td> <td>1 0</td> <td>0 0</td> </tr> </table>			CD		AB		0	0	1 0	1	1 1	1 X	0	1 1	1 1	0	1 0	0 0	2	3	2
		CD																				
	AB																					
0	0	1 0																				
1	1 1	1 X																				
0	1 1	1 1																				
0	1 0	0 0																				
1.2	Given $F = \sum m(0, 1, 5, 7, 15, 14, 10)$ , find the number of implicants, Prime implicants and Essential Prime implicants.	2	3	2																		
1.3	Construct the Booth recording of a multiplier for the following sequence. 0 0 1 1 0 1 0 1 1 0 0 1 1 0 1 0 1 0 1 1 0	2	3	3																		
1.4	Construct a full adder using 2 half adders.	2	2	4																		
1.5	What is the range of numbers in 'n' bit 2's complement representation and write the 2's complement representation -28 in 16 bit representation.	2	2	2																		

PTO

## Part B

Sl. No.	Questions	M	BT	CO
1 a)	Perform multiplication of 10101(multiplicand) and 00101 (multiplier)using Booth's algorithm.	5	3	1
b)	Give the flow chart for restoring division and perform the division operation on following number. Dividend: 8 Divisor: 5	5	3	1
2 a)	Design a 4 bit parallel adder using 1bit Full adder and Perform Binary addition on the following numbers. (Assume the numbers are represented in 2's complement form.) i. 23 and -8	5	4	4
b)	Perform the multiplication of $13 \times (-8)$ using Bit pair recording algorithm	5	3	1
3 a)	Consider the decimal number: +105.625. Represent it in equivalent binary representation in IEEE single precision format.	5	3	1
b)	Minimize the following boolean function using Kmap and implement the reduced function using only Nand gates. $F(A, B, C, D) = \Sigma m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$	5	2	2
4	Bring out the design logic behind carry look ahead adder, design a four bit carry look ahead adder and justify the same through algebraic equations.	10	3	1
5	Simplify the expression $F(A,B,C,D) = \Sigma m(0,1,2,4,6,8,9,11,13,15)$ . Using Quine Mc-cluskey method and write the prime implicants and essential prime implicants.	10	3	4

**Course Outcomes:** After completing the course, the students will be able to:-

CO 1 Apply design requirements for digital systems and Computer organization

CO 2 Analyze the models used for designing various Combinational and Sequential circuits

CO 3 Develop applications of synchronous sequential networks using flip flops, registers and counters

CO 4 Design optimized modern processors and memories for given specifications

CO 5 Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

### BT-Blooms Taxonomy, CO-Course Outcomes, M-Marks

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	L1	L2	L3	L4	L5	L6
	Test	Max Marks	30	11	2	17		9	46	5	-	-

**RV COLLEGE OF ENGINEERING®**

(An Autonomous Institution Affiliated to VTU)

**III Semester B. E. Regular / Supplementary Examinations Feb/Mar -2025****Computer Science Engineering****Common to CD / CY / CS****APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION****Time: 03 Hours****Maximum Marks: 100****Instructions to candidates:**

1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2 is compulsory. Answer any one full question from 3 and 4, 5 and 6, 7 and 8, 9 and 10.

**PART-A****M BT CO**

1	1.1	Perform the operation $(+25) + (-18)$ and represent the result in an 8-bit signed binary format using two's complement.	02	2	1
	1.2	Multiply $-9$ and $-7$ using Booth's Algorithm. Represent the numbers in 5-bit binary format.	01	3	2
	1.3	Represent $-15.375$ in IEEE 754 Single Precision format.	01	2	1
	1.4	In a 2-variable K-Map, how many minterms can be represented?	02	2	1
	1.5	What is the significance of wrapping around the edges in K-Maps?	01	1	2
	1.6	How does the Quine-McCluskey method handle don't-care conditions?	02	1	1
	1.7	Show the steps to add two 4-bit binary numbers using a binary parallel adder. Numbers: $A = 1010A = 1010$ and $B = 0111B = 0111$	02	2	3
	1.8	What is a bistable element, and how is it used in digital circuits?	01	1	1
	1.9	Differentiate between a SR latch and a D latch.	01	2	4
	1.10	How does an edge-triggered flip-flop differ from a level-triggered flip-flop?	01	1	2
	1.11	Name the logic component often used self-correcting counters to detect errors?	02	1	2
	1.12	What is the purpose of an assembler directive in assembly language programming?	02	2	4
	1.13	What is the role of the instruction register (IR) in a CPU, and how does it differ from the program counter (PC)?	02	2	4

**PART-B**

2	a	Explain the Quine-McCluskey method for simplifying Boolean expressions. Use the function: $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10, 14)$ .	08	3	3
	b	Explain the IEEE standards used for single & double precision floating point number representation with examples.	08	3	4
3	a	Describe the operation of a D flip-flop and a JK flip-flop implemented as edge-triggered flip-flops. Explain their working with truth tables and timing diagrams.	08	3	4

b	What is priority encoder? Design a 4:2 priority encoder with necessary diagram.	OR	08	3	3
4 a	With a neat diagram explain 4-bit parallel adder with accumulator. Provide the truth table for the same.		08	3	5
b	What is a multiplexer? Write a logic diagram for 8:1 multiplexer using 4 input AND and OR gates.		08	3	4
5 a	Define counter. Design a mod5 counter using JK flip flop.		08	3	4
b	Explain the working of a 3-bit shift register with the help of a neat diagram.		08	2	4
		OR			
6 a	Design and explain 3-bit binary Up-counter using T flip-flops.		12	4	4
b	Mention any four applications of shift registers.		04	2	4
7 a	With a neat diagram of connections between the processor and memory, list and explain the basic steps needed to execute the machine instruction <i>Move LOCA, R0</i> in terms of transfers between the components of processor, memory and some control commands.		06	3	2
b	What are addressing modes? Explain any five different types of addressing modes with examples.		10	2	3
		OR			
8 a	What is a subroutine? Analyze the use of call and return instructions in a subroutine with assembly language program code.		06	3	3
b	Describe with an example, explain the usage of stack frame in nested subroutine. Also demonstrate how the main program calls subroutine1 and subroutine1 calls subroutine2.		10	4	4
9 a	List the actions needed to execute the instruction ADD (R3), R1. Write and explain the sequence of control steps for executing the same.		06	2	2
b	State the importance of cache memory. Describe different types of cache mapping techniques with a neat diagram.		10	2	2
		OR			
10 a	Analyze with an example how the complete execution of the instruction is carried out in the computer system?		06	3	3
b	With the help of a neat diagram explain the organization of (1K X 1) memory chip in detail.		10	3	4