Reg. No.:

Name :

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Continuous Assessment Test II – March 2024

Programme	:	**B.Tech. (CSE – All Programme)**	Semester	:	**WS 2023-24**
Course	:	**Digital System Design**	Code	:	**BECE102L**
			Slot	:	**A1 + TA1**
Faculty	:	**RANJEET KUMAR** **G LAKSHMI PRIYA** **DHEEREN KU MAHAPATRA** **KIRUTHIKA** **GIRIJA SHANKAR** **SUKRITI** **JEAN JENIFER NESAM J**	Class Nbr	:	**CH2023240502023** **CH2023240502025** **CH2023240502027** **CH2023240502029** **CH2023240502033** **CH2023240502045** **CH2023240502054**
Time	:	**90 Minutes**	Max. Marks	:	**50**

Answer <u>ALL</u> the questions

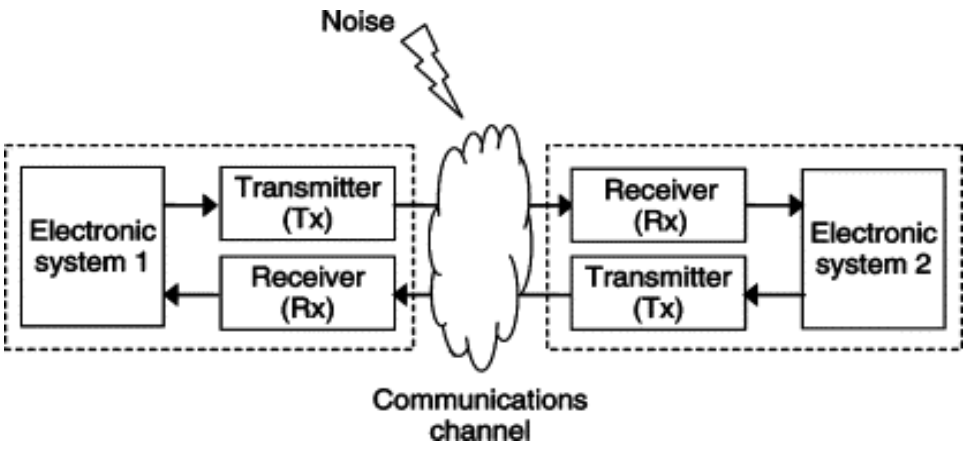
Q.No.	Sub. Sec.	Questions	Marks
1.			10

Figure.1

Errors may arise during the data transmission process depicted in Figure 1 when noise is introduced into the signal. These errors occur particularly when the level of noise surpasses a threshold significant enough to disrupt the transmitted data. To ensure data transmissions between communication nodes are accurate, design an even parity checker with data inputs A, B, and C. Also implement the circuit using suitable multiplexer and write the Verilog HDL code for the same.

Answer:

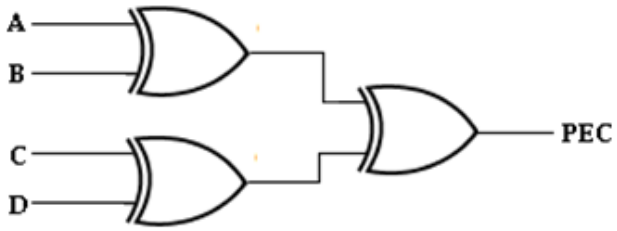
(i) Truth Table;Boolean Expression; Logic Diagram-3 Marks

4-Bit Received				**PEC**
A	**B**	**C**	**P**	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1

0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
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	0

$$PEC = (A \oplus B) \oplus (C \oplus D)$$

Logic Diagram:

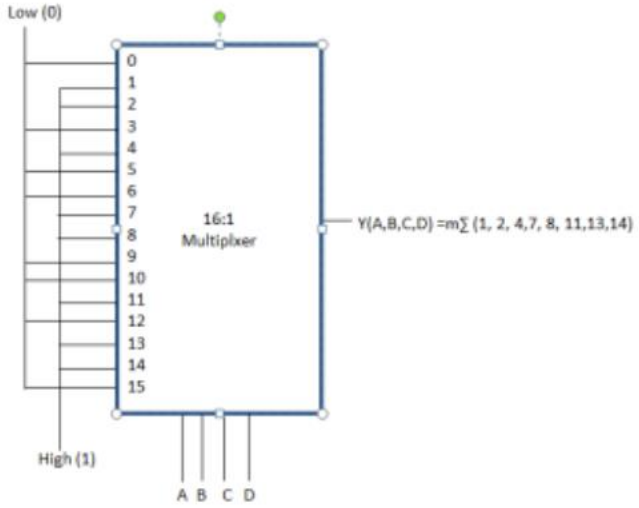


4-bit even parity checker

(ii) MUX Implementation – 4 Marks

Boolean expression in terms of minterms

$Y(A,B,C,D) = m\sum(1, 2, 4, 7, 8, 11, 13, 14)$



(iii) Verilog HDL code (Not Attached) – 3Marks

2.		Identify and implement the algorithm that involves 2's complement, addition, subtraction and shifting operations, to perform multiplication of two numbers 15 and -12. Elaborate on the step-by-step procedure involved using a flowchart.	10

Answer:

- **Initialization of registers [1 marks]**
  Set 15 (in binary 01111) as multiplicand (M) and -12 (in 2's complement binary 10100) as a multiplier (Q), set Accumulator (AC) as 0, Q-1 as 0 and count as 5.
- **Booth multiplication table [6 marks]**

{Q0,Q_{-1}}	Operation	Accumulator (AC)	Q (Q4Q3Q2Q1Q0)	Q_{-1}	Count
--	Initial	0 0 0 0 0	1 0 1 0 0	0	5
00	ASHR	0 0 0 0 0	0 1 0 1 0	0	4
00	ASHR	0 0 0 0 0	0 0 1 0 1	0	3
10	A=A-M (or) A + M' + 1	1 0 0 0 1	0 0 1 0 1	0	2
	ASHR	1 1 0 0 0	1 0 0 1 0	1	
01	A=A+M	0 0 1 1 1	1 0 0 1 0	1	1
	ASHR	0 0 0 1 1	1 1 0 0 1	0	
10	A=A-M (or) A + M' + 1	1 0 1 0 0	1 1 0 0 1	0	0
	ASHR	1 1 0 1 0	0 1 1 0 0	1	

- **Final validation with manual calculation [1 marks]**

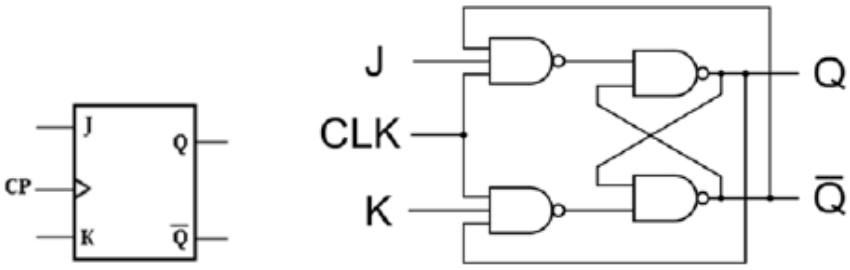
Convert final result into decimal, as 2's complement of 1101001100 => (0010110100)2 = (180)10. Q-1=1 means the output is negative hence 15*(-12) =-180.

- **Computation Analysis [2 marks]**

No. of Additions: 1
No. of Subtractions: 2
No. of Shift operations: 5

3. (a)Can JK flip-flop be considered as a universal flip-flop? If yes, then justify by comparing and contrasting with the help of its truth table and other flip-flops. (6 marks) — 10

Answer:



(a) Symbol (b) Logic diagram

CLK	Inputs		Present state	Next state	State
	J	K	Q_n	Q_{n+1}	
↑	0	0	0	0	No Change (NC)
	0	0	1	1	
↑	0	1	0	0	Reset (R)
	0	1	1	0	
↑	1	0	0	1	Set (S)
	1	0	1	1	
↑	1	1	0	1	Toggle (T)
	1	1	1	0	
↓	x	x	0	0	No Change (NC)
	x	x	1	1	

(c) Truth table

JK Flop can be used to realize all other flip flops.

(i) SR Flip flop using JK Flipflop: By eliminating the condition of J=K=1 in JK flipflop, SR Flipflop logic can be realized

(ii)D Flip flop using JK Flip flop: The Set and Reset condition alone considered for D flip flop realization using JK Flipflop. For this J & K input provided with complementary values to each other.

(iii) T Flip flop using JK Flipflop: For realizing T-Flipflop, the No Change and Toggle state will be consider. This can be achieved by assigning the same value (either 0 or 1) to JK input.

(b) Complete the timing diagram shown in Figure 2 for two types of D flip-flop devices: (i) an active HIGH enabled latch and (ii) a positive edge-triggered D flip-flop. (4 marks)

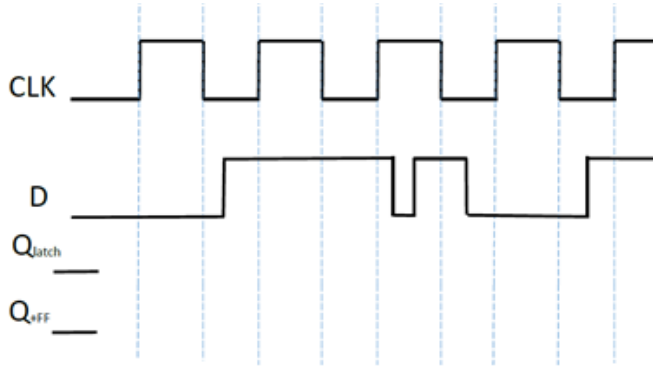
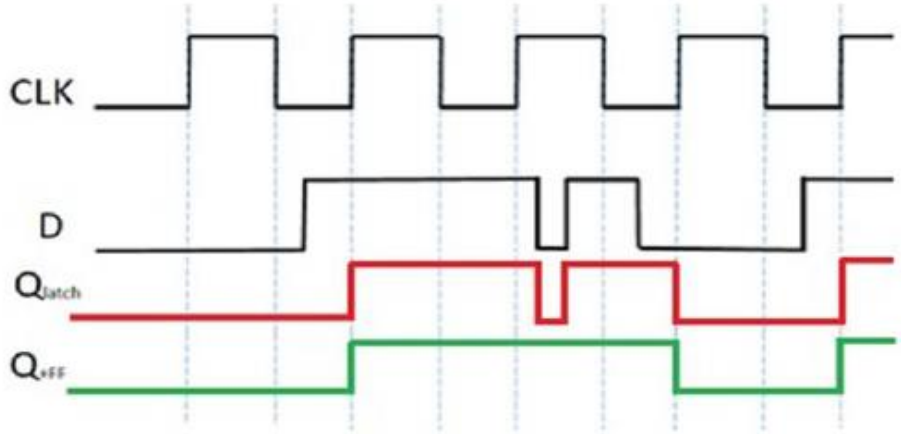


Figure.2

Answer:



4. Consider a control unit for a wheat packing machine to pack wheat of 6kg each. Present weight of the pack is measured and then it is compared with the reference value (6kg). If the wheat pack is less/greater than 6kg control signal generated to add/remove excess wheat into/from the wheat pack. If the wheat pack weight is exactly 6kg then control signal is generated to make a pack. Design an appropriate control circuit for the above mentioned comparison operation using basic logic gates. Also write a Verilog HDL code in dataflow modelling for the same. 10

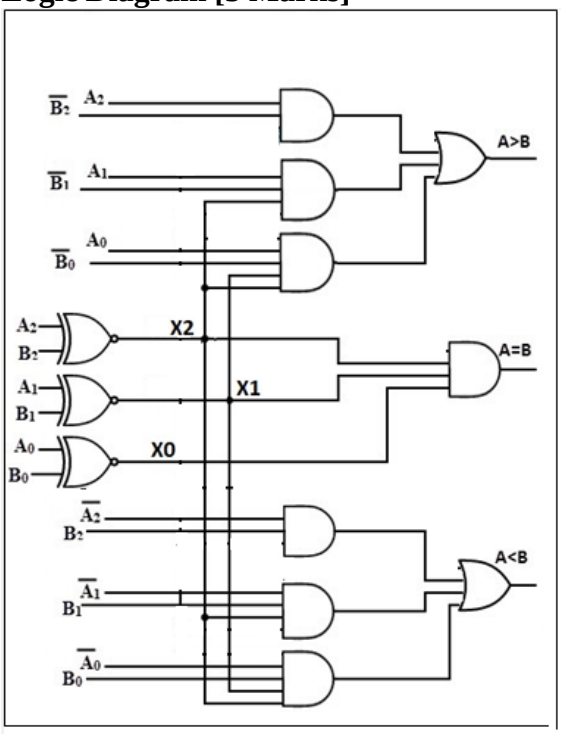
Answer:

- **Truth Table [2 Marks]**

Comparing inputs			**Outputs**		
A2,B2	**A1,B1**	**A0,B0**	**A>B**	**A=B**	**A<B**
A2>B2	X	X	1	0	0
A2<B2	X	X	0	0	1
A2=B2	A1>B1	X	1	0	0
A2=B2	A1<B1	X	0	0	1
A2=B2	A1=B1	A0>B0	1	0	0
A2=B2	A1=B1	A0<B0	0	0	1
A2=B2	A1=B1	A0=B0	0	1	0

- **Equation [2 marks]**
  (A = B) = X2X1X0 (where X = A xnor B)
  (A>B) = A2B2′ +X2A1B1′ +X2X1A0B0′
  (A<B) = A2′B2 +X2A1′B1 +X2X1A0′B0

- **Logic Diagram [3 Marks]**



- **Verilog code in dataflow modelling [3 Marks]**

```
module MC(AgreaterB, AlessB, AequalB, A, B);
input [2:0]A, B;
output AgreaterB, AlessB, AequalB;
wire [2:0]Y;
assign Y[0]= ~ (A[0]^B[0]);
assign Y[1]= ~ (A[1]^B[1]);
assign Y[2]= ~ (A[2]^B[2]);
assign AequalB = Y[0] & Y[1] & Y[2];
assign AgreaterB= (A[2] & (~B[2])) +(Y[2] & A[1] & (~B[1]) ) + (Y[2] & Y[1] & A[0] & (~B[0]));
assign AlessB= ((~A[2]) & B[2]) +(Y[2] & (~A[1]) & B[1] ) + (Y[2] & Y[1] & (~A[0])& B[0]);
endmodule
```

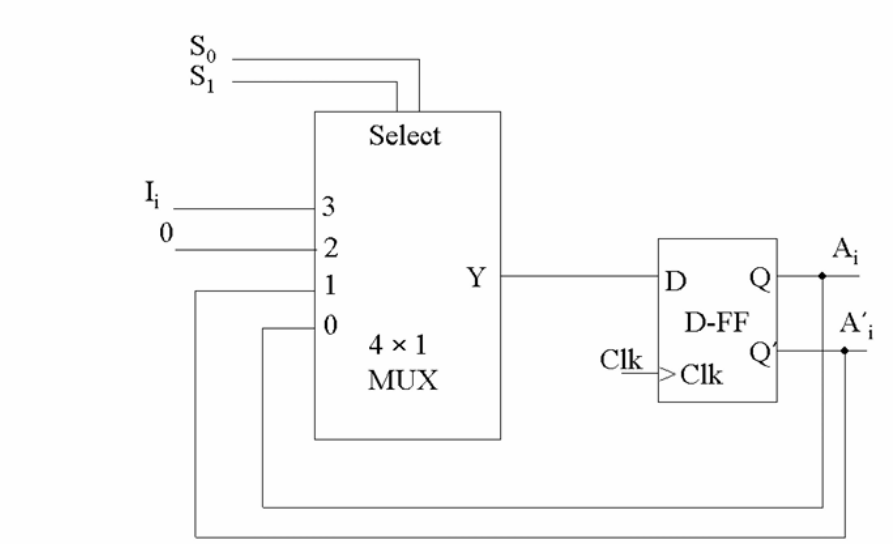
5. (a) Draw the logic diagram of a 4-bit register incorporating D flip-flop and a 4 × 1 multiplexer with mode selection inputs S_1 and S_0. The register functions according to the following Table 1: (5 Marks) — 10

S_1	S_0	**Register Operation**
0	0	No change
0	1	Complement the output
1	0	Clear register to 0
1	1	Load Data

Table. 1

Answer:

One stage of the register:



(b) For the circuit shown in figure 3, two 4-bit parallel-in serial-out shift registers loaded with the data shown are used to feed the data to a Full Subtractor (FS). Initially, all the flip-flops are in clear state. After applying two clock pulses, identify the outputs of the Full Subtractor (FS). (5 Marks)

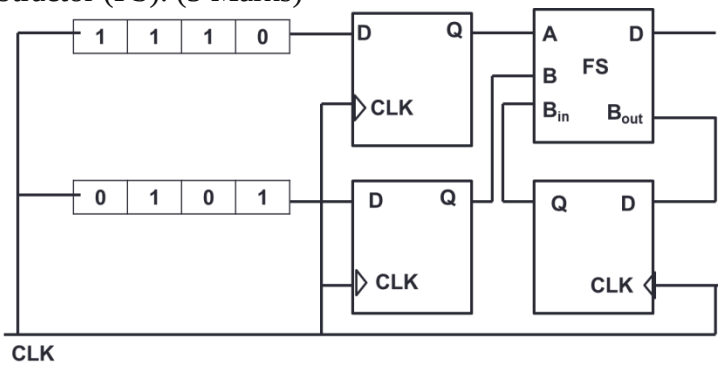


Figure.3

Answer: