USN | | | | | | | | | | |

Internal Assessment Test 3 – Feb. 2023

Sub:	Digital System Design using Verilog					Sub Code:	21EC32	Branch:	ECE
Date:	08-02-2023	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	3/A,B,C,D		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1 (a)	Write the characteristics equation for SR flip-flop & T flip –flop, and draw the excitation table for the JK flip-flop.	[04]	CO3	L1
1 (b)	Design a Mod-6 synchronous counter using T Flip-flop.	[06]	CO3	L2
2	Design a 4-bit universal shift register using positive edge triggered D-flip-flop and multiplexers to operate as indicated below:	[10]	CO3	L3
3	Design a 4-bit ripple counter using JK flip-flop along with the waveforms. Compare the counters based shift register.	[10]	CO3	L2
4	Illustrate with examples the data types used to define nets, registers, vectors and arrays.	[10]	CO4	L2
5	Write the Verilog code for the Half Adder using the data-flow description, assign a delay time to the signal assignment statement with the timing waveform.	[10]	CO4	L3
6	Illustrate the IF statement, IF as ELSE-IF, signal and variable assignment with an example.	[10]	CO4	L3
7	Explain the behavioral description of a Positive Edge-Triggered JK flip-flop using the CASE statement with the timing waveform.	[10]	CO4	L2

Mode select	Operation
00	Hold
01	Right Shift
10	Left Shift
11	Parallel Load

COURSE INSTRUCTOR CCI HOD

21EC32 - Digital System Design using Verilog

Internal Assessment Test 3

Answer Key

1. a) Characteristic Equation

SR Flip-Flop: $Q^+ = S + R'Q$ (1 Mark)

T Flip-Flop: $Q^+ = TQ' + T'Q$ (1 Mark)

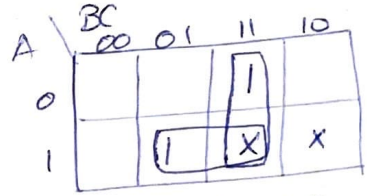
Excitation Table: (2 Marks)

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

1. b) Mod-6 synchronous counter using T-Flip-Flop

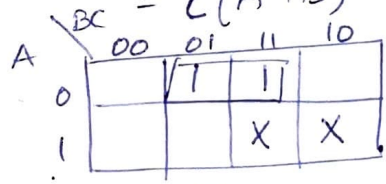
Q_A	Q_B	Q_C	Q_A^+	Q_B^+	Q_C^+	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0	1

(2 Marks)



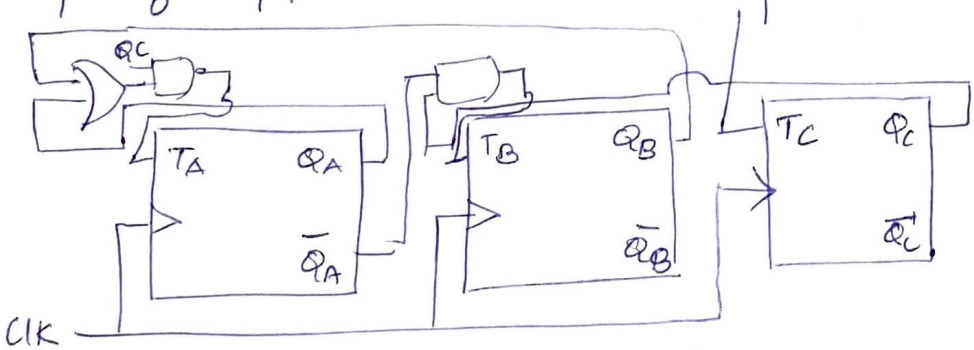
$T_A = BC + AC$

$= C(A + B)$



$T_B = \bar{A}C$

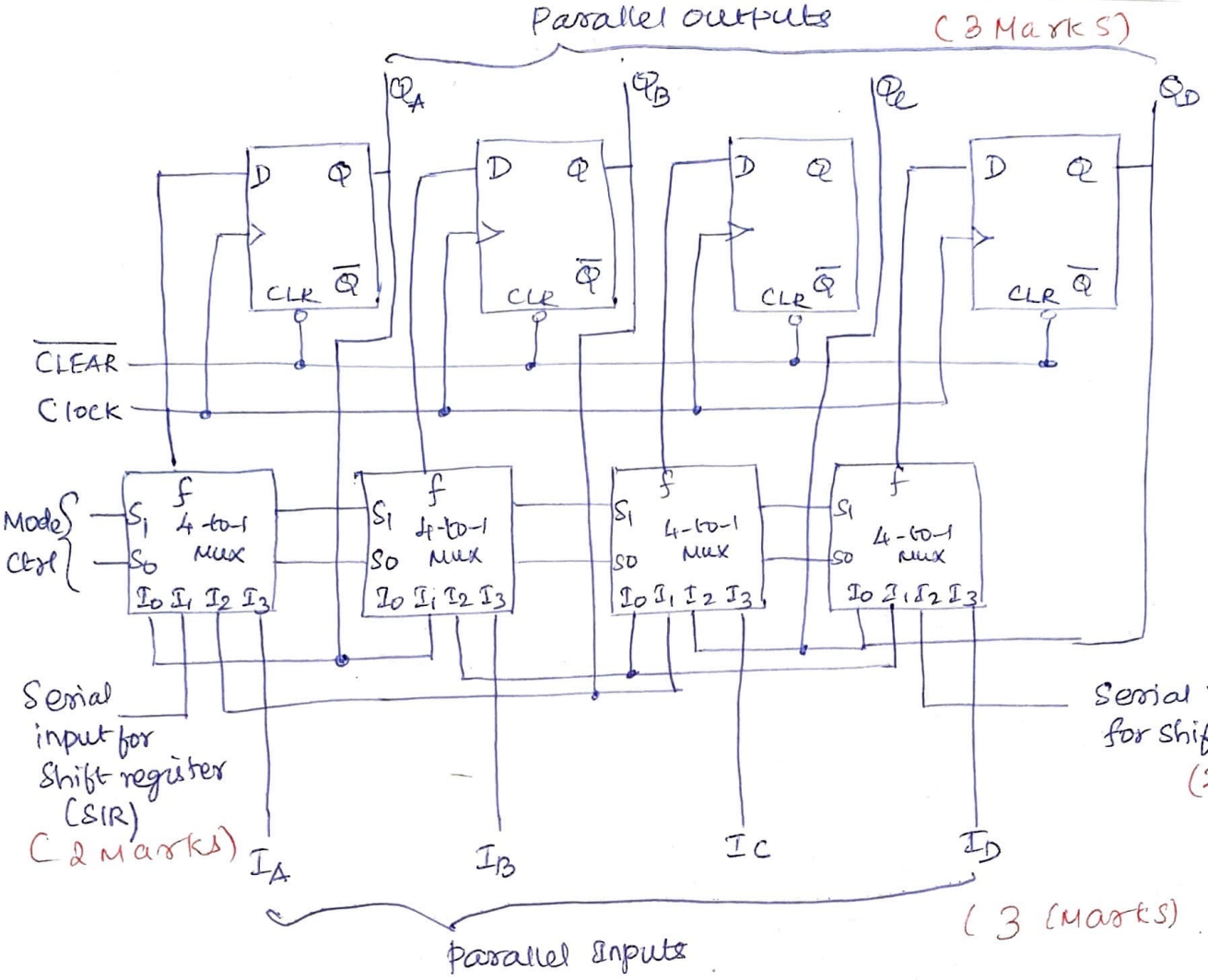
$T_C = 1$ (2 Marks)



(2 Marks)

2. 4-bit Universal Shift Register

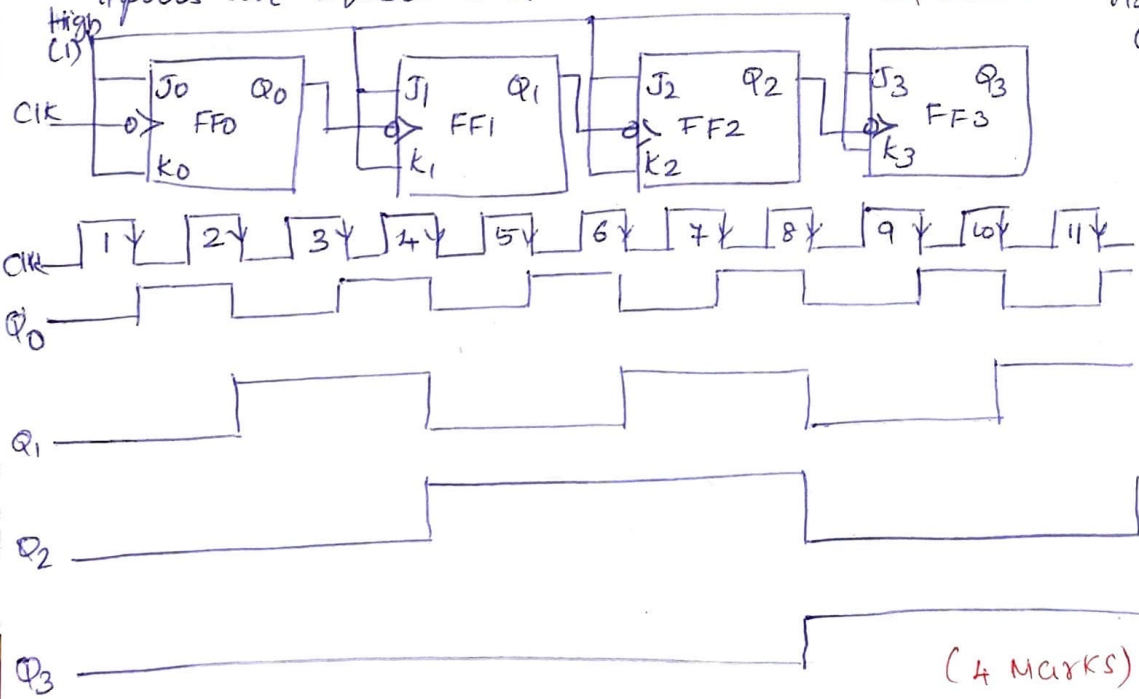
Mode select	Operation
00	Hold
01	Right shift
10	Left shift
11	parallel Load



3. 4-bit Ripple counter using JK Flip-Flops. (2 Marks)

A binary ripple counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the clock-pulse input of the next-higher order flip flop. All J and K inputs are equal to 1.

(4 Marks)



(4 Marks)

MSB			LSB	Decimal Value
Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

4. Verilog Data Types

Verilog supports several data types, including: nets, registers, vectors, integer, real, parameters and arrays.

Nets: It is declared by the predefined coord wire. Nets have value that change continuously by the circuits that are driving them. (2 Marks)

Verilog Net Values

Value	Definiteon
0	logic 0 (false)
1	logic 1 (true)
X	unknown
Z	High impedance

Example:

```
wire sum;          (Declares a net by name sum)
wire S1 = 1'b0;    (initial value of S1 is declared).
```

Registers: It stores the value until they are updated. It is defined by the predefined word reg (4 Marks)

Verilog Register Values

Value	Definition
0	Logic 0 (false)
1	Logic 1 (true)
X	Unknown
Z	High impedance

Example:

```
reg Sum_total;
```

The statement declares a register by the name Sum_total.

Vectors: Vectors are multiple bits. A register or a net can be declared as a vector. Vectors are declared by brackets[]. (2 Marks)

Example:

```
wire [3:0] a = 4'b1010;
reg [7:0] total = 8'd12;
```

The statement declares net a. It has 4-bits and its initial value is 1010. The second statement describes a register total. Its size is 8 bits and decimal value is 12. (2 Marks)

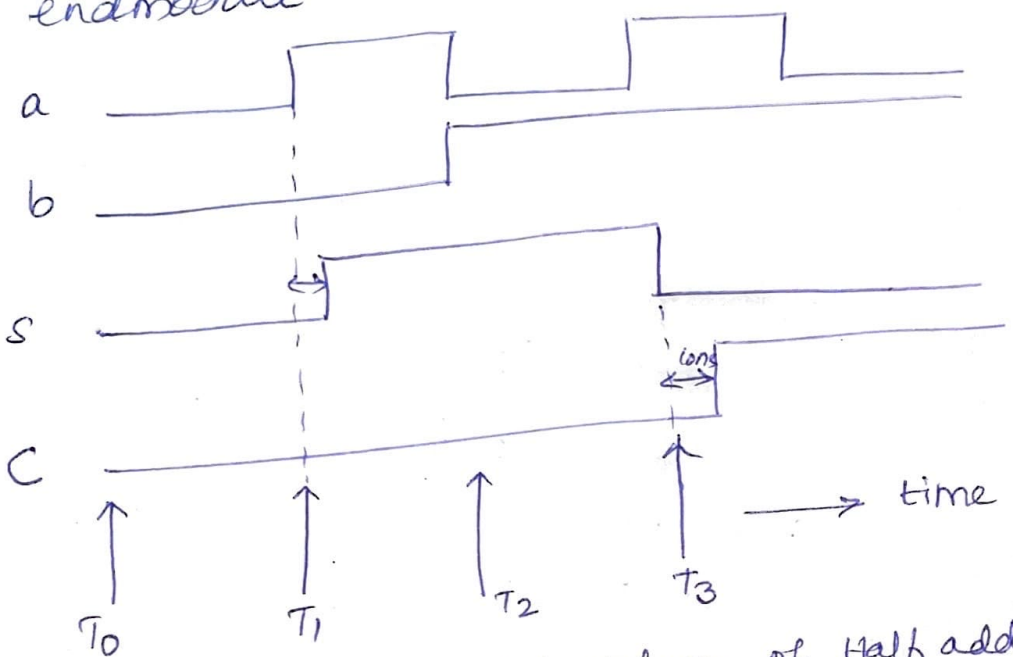
Arrays: It doesnot have a predefined word for array. Registers and integers can be written as arrays.

Example:

```
parameter N = 4;
parameter M = 3;
reg signed [M:0] carry[0:N];
reg [M:0] b[0:N];
integer Sum[0:N];
```

5. Verilog Half Adder Description - Dataflow (6 Marks)

```
module half_adder(a,b,s,c);
input a,b;
output s,c;
assign #10 s = a ^ b;
assign #10 c = a & b;
endmodule
```



(4 Marks)

Simulation Waveform of Half adder (6 Marks)

6. IF statement.

Verilog code

```
if (Boolean Expression)
begin
   Statement 1;
   Statement 2;
   ...
   end
else
begin
   statement a;
   statement b;
   ...
end.
```

Eg:
```
if(CLK==1)
   begin
      temp=S1;
   end
```

Execution of IF as else-IF.

Verilog code

```
if (Boolean Expression1) then
statement 1; statement 2; ....
elseif (Boolean Expression 2) then
statement i; statement ii; .....
else
statement a; statement b; ....
end if
```

Example

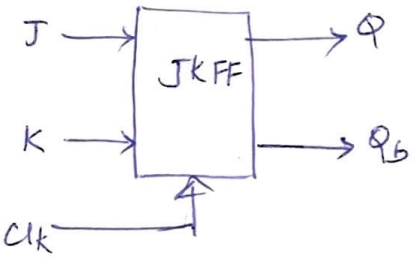
```
if (signal1 == 1)
temp = S1;
else if (signal2 == 1)
temp = S2;
else
temp = S3;
```

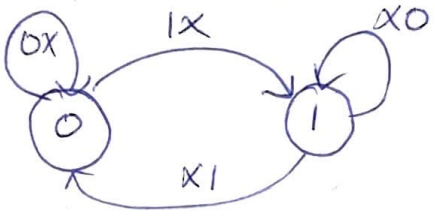
Signal and variable Assignment (4 Marks)

Example: module D_latch (d, E, Q, Qb);
input d, E;
Output Q, Qb;
reg Q, Qb;
always @ (d, E)
begin
if (E == 1)
begin Q = d;
Qb = ~Q;
end
end
endmodule

7. Positive Edge Triggered JK Flip-flop

positive edge Flip-Flops sample the input only at the positive edges of the clock; any change in the input that does not occur at the edges is not sampled by the output. (4 Marks)





State diagram

Verilog positive Edge Triggered JK Flip-flop using case (6 Marks)

module JK_FF (JK, Clk, q, qb);
input [1:0] JK;
input Clk;
output reg q, qb;
always @ (posedge clk)
begin
case (JK)
2'd0 : q = q;
2'd1 : q = 0;
2'd2 : q = 1;
2'd3 : q = ~q;
endcase
qb = ~q;
end
endmodule