Assignment-2 (Y22)

Design and Simulate the following modules on Iverilog

Sequential Circuits:

DFlipFlop
JKFlipFlop
SR Latch
Shift Register
T FlipFlop

Combinational Circuits:

4-bit FullAdder
4-to-1 Multiplexer
3-Bit Comparator
3-to-8 Decoder

D FLIP-FLOP

Roll No. 220280

```
module DFF (
input wire clk,
input wire reset,
input wire d,
output reg q
);
always @(posedge clk or posedge reset) begin
 if (reset)
 q <= 1'b0;
else
q <= d;
 end
endmodule
```

JK FLIP-FLOP

```
module JKFF (
 input wire clk,
input wire reset,
input wire j, k,
output reg q, qbar
);
always @(posedge clk or posedge reset) begin
 if (reset)
 q <= 1'b0;
else begin
case ({j, k})
 2'b00: ; // No change
2'b01: q <= 1'b0;
2'b10: q <= 1'b1;
2'b11: q <= ~q;
 endcase
 end
qbar <= ~q;
 end
endmodule
```

SR LATCH

```
module SRLat (
 input wire set, reset,
output reg q, qbar
);
always @(set, reset) begin
 if (set && ~reset)
q = 1'b1;
else if (~set && reset)
q = 1'b0;
else if(~set &&~reset)
 q=q;
else if (set&&reset)
q=1'bx;
 qbar <= ~q;
end
endmodule
```

SHIFT REGISTER

```
module SReg(
 input wire clk, reset,
input wire [7:0] din,
output reg [7:0] dout
);
always @(posedge clk or posedge reset) begin
 if (reset)
 dout <= 8'b0;
else begin
dout <= {dout[6:0], din};
end
 end
endmodule
```

T FLIP-FLOP

```
module TFF (
 input wire clk, reset,
input wire t,
output reg q
);
always @(posedge clk or posedge reset) begin
 if (reset)
 q <= 1'b0;
else if (t)
q <= ~q;
 end
endmodule
```

4 BIT FULL ADDER

```
module FA4bit (
 input [3:0] a, b,
input cin,
output [3:0] sum,
output cout
);
assign {cout, sum} = a + b + cin;
endmodule
```

4 TO 1 MUX

```
module Mux4to1 (
 input [3:0] data,
input [1:0] sel,
output wire out
);
assign out = (sel == 2'b00) ? data[0] :
 (sel == 2'b01) ? data[1] :
(sel == 2'b10) ? data[2] :
 data[3];
endmodule
```

3 BIT COMPARATOR

```
module Comp3bit (
 input [2:0] A, B,
output EQ, LT, GT
);
assign EQ = (A == B);
assign LT = (A < B);
assign GT = (A > B);
endmodule
```

3 TO 8 DECODER

```
module Dec3to8 (
 input [2:0] a,
output reg [7:0] y
);
always @* begin
 if (a == 3'b000)
 y = 8'b00000001;
else if (a == 3'b001)
y = 8'b00000010;
else if (a == 3'b010)
y = 8'b00000100;
else if (a == 3'b011)
y = 8'b00001000;
else if (a == 3'b100)
y = 8'b00010000;
else if (a == 3'b101)
y = 8'b00100000;
else if (a == 3'b110)
y = 8'b01000000;
else
y = 8'b10000000;
 end
endmodule
```