Ece241 part 2

Lab 5 part 2 (similar)

- Want a 4 bit counter

- Display on hex0

- Active high reset synchronous

- Up counter

- Counts at two speeds 1 Hz speed 0

- 0.5 Hz speeds 1

- 50 mHz clock on DEI-soC board

Module rateDivide( clock, reset, speed, enable);

input dock, reset, speed;

output A enable;

reg[26:0] downCount; //27 bit

always@(posedge clock)

begin

if ((reset == 1’b1) || (downCount == 27’d0))

begin

if (speed == 1’b0)

downCount <= 27’d50000000;

else

downCount <= 27’d10000000;

end

else

begin

downCount < = downCount - 1’b1;

end

end

assign enable = (downCount == 27’d0)

? 1’b1:1’b0;

end module

module T\_flip\_flop (

input wire clk, // Clock input

input wire T, // Toggle input

output reg Q, // Flip-flop output

output reg Qbar // Inverted output

);

reg Q\_next;

always @(posedge clk) begin

if(T) Q <= ~Q; // Toggle Q if T is 1

end

always @(Q) begin

Qbar = ~Q; // Invert Q for Qbar

end

endmodule

module T\_flip\_flop (

input wire clk, // Clock input

input wire T, // Toggle input

output reg Q, // Flip-flop output

output reg Qbar // Inverted output

);

reg Q\_next;

always @(posedge clk) begin

if(T) Q <= ~Q; // Toggle Q if T is 1

end

always @(Q) begin

Qbar = ~Q; // Invert Q for Qbar

end

endmodule

module T\_register\_8bit(

input wire clk, // Clock input

input wire [7:0] T, // 8-bit Toggle input

output reg [7:0] Q, // 8-bit Flip-flop output

output reg [7:0] Qbar // 8-bit Inverted output

);

// Instantiating 8 T flip-flops

T\_flip\_flop tff0(clk, T[0], Q[0], Qbar[0]);

T\_flip\_flop tff1(clk, T[1], Q[1], Qbar[1]);

T\_flip\_flop tff2(clk, T[2], Q[2], Qbar[2]);

T\_flip\_flop tff3(clk, T[3], Q[3], Qbar[3]);

T\_flip\_flop tff4(clk, T[4], Q[4], Qbar[4]);

T\_flip\_flop tff5(clk, T[5], Q[5], Qbar[5]);

T\_flip\_flop tff6(clk, T[6], Q[6], Qbar[6]);

T\_flip\_flop tff7(clk, T[7], Q[7], Qbar[7]);

endmodule