

RAJ KAUSHAL YADAV

REAL TIME CLOCK

RTL CODE

//DESIGN CODE

```
module rtc (  
    input wire clk,                // 1Hz clock input  
    input wire reset,              // synchronous reset  
    output reg [5:0] seconds,       // 0-59  
    output reg [5:0] minutes,       // 0-59  
    output reg [4:0] hours          // 0-23  
);  
  
always @(posedge clk) begin  
    if (reset) begin  
  
        // Reset all time values to 00:00:00  
  
        seconds <= 0;  
        minutes <= 0;  
        hours <= 0;  
    end  
    else begin  
  
        // Increment seconds  
        if (seconds == 6'd59) begin  
            seconds <= 0;  
            minutes <= minutes + 1;  
            if (minutes == 6'd59) begin  
                minutes <= 0;  
                hours <= hours + 1;  
                if (hours == 5'd23) begin  
                    hours <= 0;  
                end  
            end  
        end  
    end  
end
```

// Increment minutes

if (minutes == 6'd59) begin

minutes <= 0;

// Increment hours

if (hours == 5'd23)

hours <= 0;

else

hours <= hours + 1;

end

else

minutes <= minutes + 1;

end

else

seconds <= seconds + 1;

end

end

endmodule

//testbench

`timescale 1s / 1ms

// 1s simulation time unit

module rtc_tb;

// Inputs

reg clk;

reg reset;

// Outputs

wire [5:0] seconds;

wire [5:0] minutes;

wire [4:0] hours;

// Instantiate the RTC module

rtc uut (

.clk(clk),

.reset(reset),

.seconds(seconds),

.minutes(minutes),

.hours(hours)

);

// Clock generation: 1Hz clock (toggle every 0.5s)

initial begin

clk = 0;

forever #0.5 clk = ~clk;

end

// Stimulus

initial begin

 \$display("Time\t\tHours:Minutes:Seconds");

// Apply reset

 reset = 1;

 #2;

 reset = 0;

// Simulate for 100 seconds

 repeat (100) begin

 @(posedge clk);

 \$display("%t\t%0d:%0d:%0d", \$time, hours, minutes, seconds);

 end

 \$finish;

end

endmodule

WAVEFORM

