Real Time Clock

RTL

```
module rtc_24h_clock (
    input wire clk,
    input wire reset,
    output reg [4:0] hrs = 0,
    output reg [5:0] mins = 0,
    output reg [5:0] sec = 0
);
    always @(posedge clk or posedge reset) begin
        if (reset) begin
            hrs <= 0;
            mins <= 0;
            sec <= 0;
        end else begin
            if (sec == 59) begin
                sec <= 0;
                if (mins == 59) begin
                    mins <= 0;
                    if (hrs == 23)
                        hrs <= 0;
                    else
                        hrs <= hrs + 1;
                end else begin
                    mins <= mins + 1;
                end
            end else begin
                sec <= sec + 1;
            end
        end
    end
endmodule
```

Test Bench

```
`timescale 1ns / 1ps
module tb_rtc_24h_clock;
    reg clk = 0;
    reg reset = 1;
    wire [4:0] hrs;
    wire [5:0] mins;
    wire [5:0] sec;
    rtc_24h_clock uut (
        .clk(clk),
        .reset(reset),
        .hrs(hrs),
        .mins(mins),
        .sec(sec)
    );
    // Fast clock: 60ns period (30ns high + 30ns low)
    always #30 clk = ~clk;
    initial begin
        // Reset for first 100ns
        reset = 1;
        #100;
        reset = 0;
        // Run long enough to see multiple minutes/hours
        #1000000;
        $finish;
    end
endmodule
```

Output Waveforms