HDL PROJECT

DIGITAL CLOCK

WITH ALARM FEATURE

Group Members:

1.Kotike Siddartha – B230672EC

2.Ch. Nagendra Mahesh – B230268EC

PROJECT ABSTRACT

Project Objective:

Digital Clock with AM/PM Display, Reset and Alarm Features.

Description:

This project implements a fully functional 24-hour digital clock in Verilog. The clock features a cascading series of counters, beginning with a mod-60 counter for counting seconds, which is connected to a mod-60 counter for minutes, and finally to a mod-12 counter for hours. The system supports both AM/PM display and includes an adjustable alarm feature, allowing users to set specific times for alerts. A reset function is also integrated, enabling the clock to be reset to their initial states. Designed using modular components, this clock demonstrates efficient timekeeping and is well-suited for digital design applications. The behavioural level of abstraction approach simplifies the design, providing clear logic.

HDL Project Proposals:

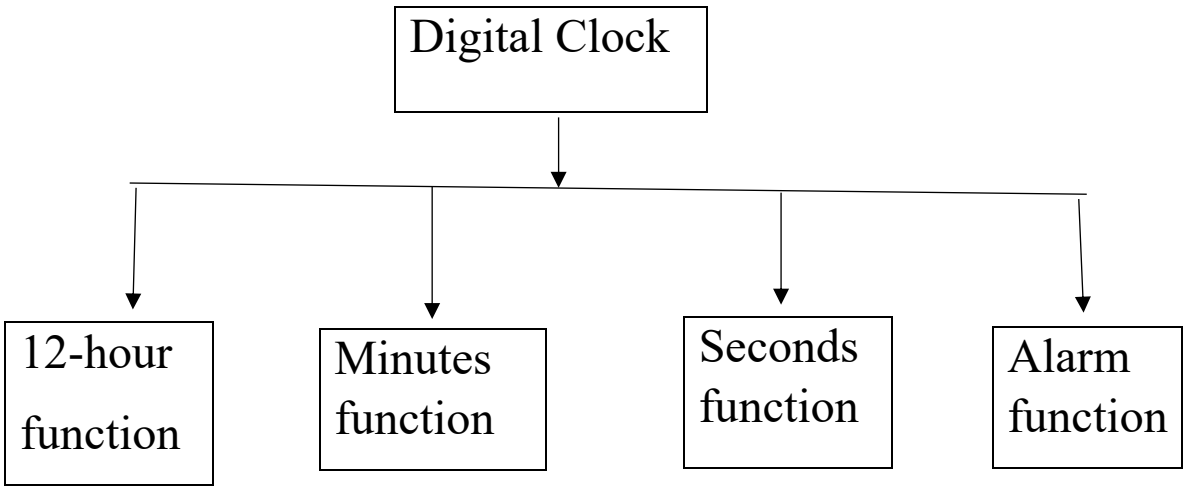
1) Objective:

1. The output of the project gives the functioning like a digital clock which displays ‘hrs: min: sec’ with am or pm in a 12-hour clock format.
2. The digital clock also contains an alarm feature.

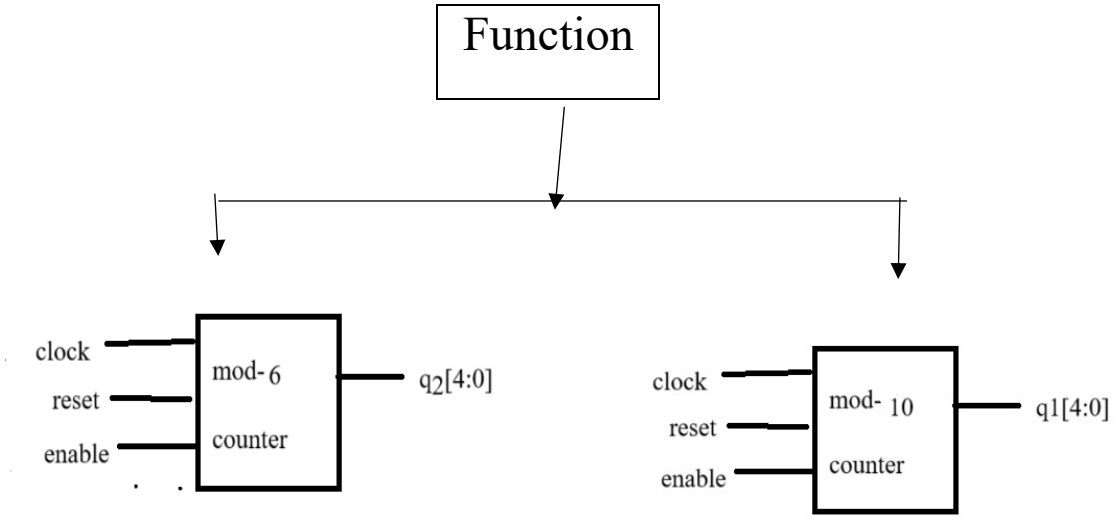
2) Ideas of Implementation:

1. We are creating a mod-6 and mod-10 counters to implement the seconds and minute function.
2. A separate module is written for mod-12 counter for implementing hour and detection of am or pm.
3. Alarm feature is directly implemented in the main module sequentially with respect to the clock.

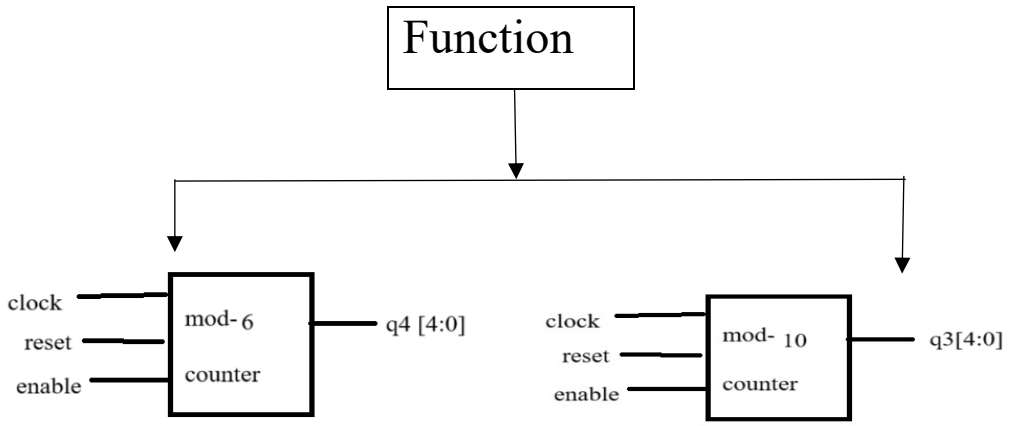
Block Diagram



Seconds function contains:



Minutes Function contains:



Project Code:

Mod-10 Counter:

```
module bcd_counter (input clock, reset, enable, output reg [3:0] q);
   always@(clock)begin
      if(reset)begin
         q<=4'd0;
      end
      else if (enable)begin
         if (q==4'd9) begin
            q<=4'd0;
         end
         else begin
            q<=q+1;
```

```
        end
      end
      else begin
            q<=q;
      end
   end
endmodule
```

Specifications:

Port Name	Terminal	Data type
Clock, reset, enable	Input	Net
[3:0] q1	Output	Reg

- A mod-10 bcd counter is designed using inputs clock, reset and enable and the output obtained is a 4-bit BCD number (0-9).

Mod-6 Counter:

```
module mod6 (input clock, reset, enable, output reg [3:0] q);
   always@(clock)begin
```

```
        if(reset)begin
            q<=4'd0;
        end
        else if (enable)begin
            if (q==4'd5) begin
                q<=4'd0;
            end
            else begin
                q<=q+1;
            end
        end
        else begin
             q<=q;
        end
    end
endmodule
```

Specifications:

Port Name	Terminals	Data types
Clock, reset, enable	Input	Net
[3:0] q2	Output	Reg

- A mod-6 counter is implemented using inputs clock, reset, enable and the output obtained is a 4-bit BCD number (0-5).

Hour Module:

```
module hour (
        input clock, reset, enable,
        output reg [3:0] d2, d1,
        output reg am);
   /*initial begin // (This initial block in meant for only simulation.
For synthesis, reset signal is used for initializing the values.)
       d2[3:0] =4'b0001;
       d1[3:0] =4'b0010;
       am=0;
   end*/
   always@ (posedge clock) begin
       if (reset)begin
          d2[3:0] =4'b0001;
          d1[3:0] =4'b0010;
          am<=0;
       end else if(enable)begin
          if ((d2[3:0] ==4'd1) &(d1[3:0] ==4'd1)) begin
             d1[3:0] <=d1[3:0] +1;
```

```
            am<=(~am);
        end else if ((d2[3:0] ==4'd1) &(d1[3:0] ==4'd2)) begin
            d2[3:0] <=4'b0000;
            d1[3:0] <=4'b0001;
        end else if (d1[3:0] ==4'd9) begin
            d1[3:0] <=4'd0;
            d2[3:0] <=d2[3:0] +1;
        end else begin
            d1[3:0] <=d1[3:0] +1;
        end
      end
       else begin
           d2<=d2;
           d1<=d1;
       end
   end
endmodule
```

Port name	Terminal	Data type
Clock, reset, enable	Input	Net
[3:0] d2, [3:0] d1	Output	Reg
am	Output	Reg

- An hour module is implemented using the inputs clock, reset, enable and q6, q5 as output terminals with am as an output.
- This module displays the output like a 12hr clock format with the am display.

Main Module:

```
module Clock (
    input clock, reset, enable, r_m,
    input [15:0] r_time,
    output [3:0] q1, q2, q3, q4, q5, q6,
    output am,
    output reg alarm);
    wire w, x, y, z;
    bcd_counter counter1(clock, reset, enable, q1);
    assign x = (q1 == 4'd9);
    mod6 counter2(clock, reset, x, q2);
    assign y = x & (q2 == 4'd5);
    bcd_counter counter3(clock, reset, y, q3);
    assign z = y & (q3 == 4'd9);
    mod6 counter4(clock, reset, z, q4);
    assign w = z & (q4 == 4'd5);
    hour counter5(clock, reset, w, q6, q5, am);
```

```
    always @(clock) begin

        if ((r_time == {q6, q5, q4, q3}) && (r_m == am)) begin
            alarm <= 1;
        end else begin
            alarm <= 0;
        end
    end
endmodule
```

Port Name	Terminal	Data Type
Clock, reset, enable, r_m	Input	Net
r_time	Input	Net
[3:0] q6, q5, q4, q3, q2, q1	Output	Net
am	Output	Net
alarm	Output	Reg

- In the main module, the modules instantiated are mod_6 and bcd_counter, hour.

- The bcd_counter ‘counter_1’ is given the enable as high for all the cases such that the unit place of the seconds clock runs.
- The mod-6 ‘counter_2’ is enabled only when the output of the counter_1 changes from 9→0.The output of this counter represents the ten’s digit of the second’s clock.
- The bcd_counter ‘counter_3’ enables only when q2 changes from 5→0 and q1 changes from 9→0.This counter represents the unit digit of the minutes part.
- The mod-6 ‘counter_4’ is then instantiated and enabled only when the enable of the counter_3 goes high and the output of the counter_3 changes from 9→0.This counter represents the tens digit of the minutes part.
- The hour module is then instantiated and it is enabled when the counter_1 is changing from 9→0 and the outputs of counter_2 =5, counter_3=9, counter_4=5. (59:59)
- The hour module also gives the output terminal ‘am’ high when the time is ‘am’ and goes low when the time is ‘pm’.
- When the clock reaches the input required time i.e., matches the r_time and r_m then the output terminal alarm goes high until the mismatch occurs.

TESTING AND VERIFICATION:

TESTBENCH 1:

```
module seconds_tb;
   reg clock, reset, enable, r_m;
   reg [15:0] r_time;
   wire [3:0] q1, q2, q3, q4, q5, q6;
   wire am, alarm;

   Clock dut(clock, reset, enable, r_m, r_time, q1, q2, q3, q4, q5, q6, am, alarm);
   initial begin
      reset = 1;
      clock = 0;
      enable=1;
      r_time = 16'b0001_0010_0000_0001;
      r_m = 0;
      #5 reset = 0;
      #200 reset =1;
      #10 $finish;
   end
   always #1 clock = ~clock;

   initial
      $monitor("$time=%g,clock=%b,reset=%b,enable=%b,r_m=%b,r_time=%b,%d%d
hrs:%d%d min:%d%d sec AM=%b,Alarm=%b", $time, clock, reset, enable, r_m, r_time,
q6, q5, q4, q3, q2, q1, am, alarm);
   endmodule
```

NOTE:

Initially, reset is set 1, hence the time is 12 hrs.

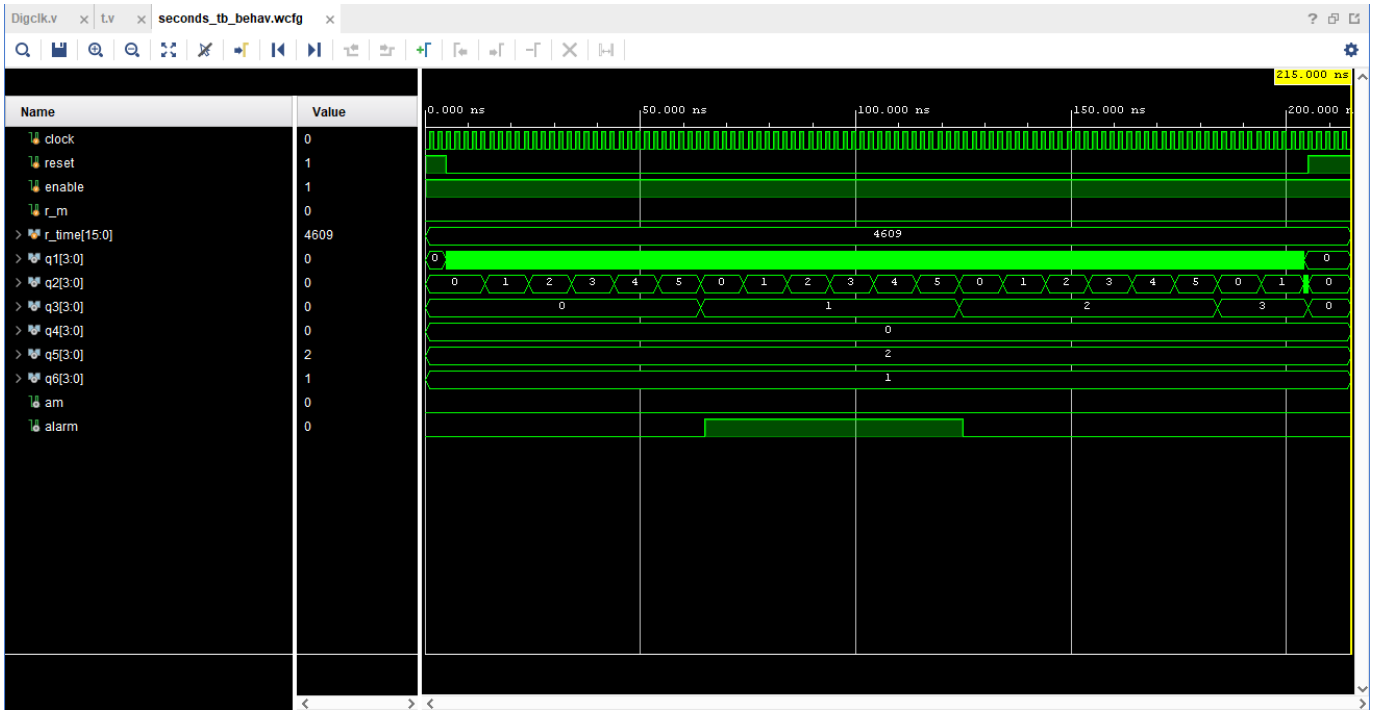
At t=5ns, reset is released. Therefore, time starts from 12:00 pm.

Alarm is set to 12:01 pm. Hence from 12:01:00 to 12:01:59 alarm is triggered.

Later, alarm signal is released and time starts counting.

Similar by increasing the test time we can show the change in hours and “am” and “pm”

Or we can directly test the hour module to witness the transitions, which is discussed soon.

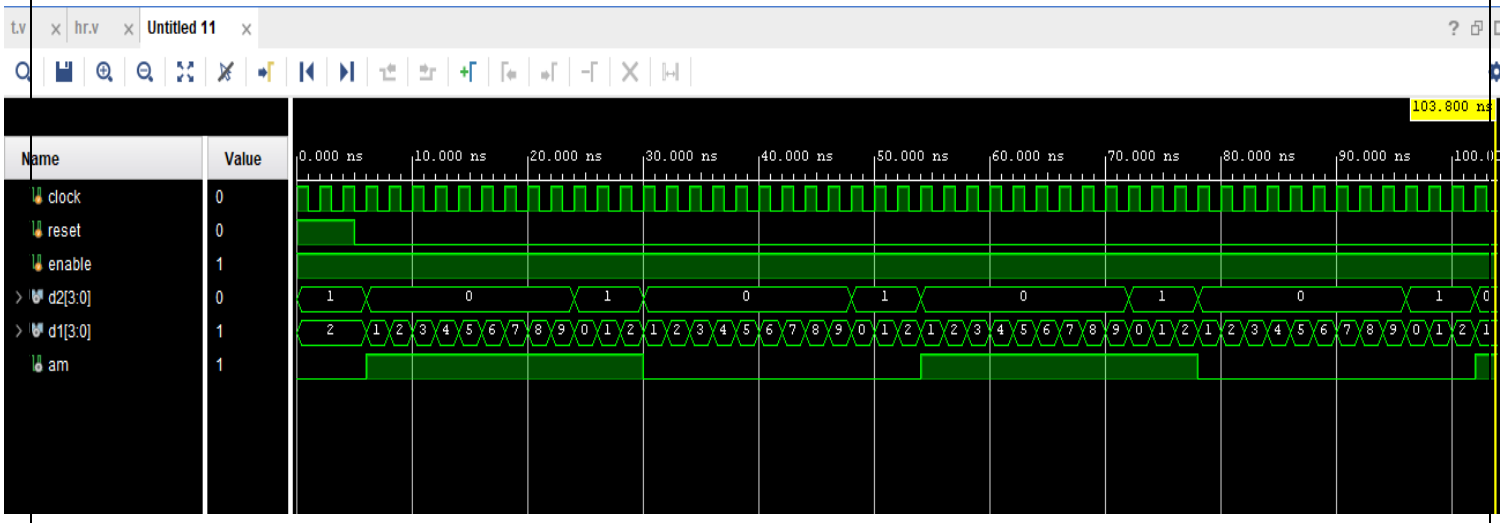


```
# run 1000ns
$time=0,clock=0,reset=1,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 0 sec AM=0,Alarm=0
$time=1,clock=1,reset=1,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 0 sec AM=0,Alarm=0
$time=2,clock=0,reset=1,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 0 sec AM=0,Alarm=0
$time=3,clock=1,reset=1,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 0 sec AM=0,Alarm=0
$time=4,clock=0,reset=1,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 0 sec AM=0,Alarm=0
$time=5,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 1 sec AM=0,Alarm=0
$time=6,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 2 sec AM=0,Alarm=0
$time=7,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 3 sec AM=0,Alarm=0
$time=8,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 4 sec AM=0,Alarm=0
$time=9,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 5 sec AM=0,Alarm=0
$time=10,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 6 sec AM=0,Alarm=0
$time=11,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 7 sec AM=0,Alarm=0
$time=12,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 8 sec AM=0,Alarm=0
$time=13,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 0 9 sec AM=0,Alarm=0
$time=14,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 0 sec AM=0,Alarm=0
$time=15,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 1 sec AM=0,Alarm=0
$time=16,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 2 sec AM=0,Alarm=0
$time=17,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 3 sec AM=0,Alarm=0
$time=18,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 4 sec AM=0,Alarm=0
$time=19,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 5 sec AM=0,Alarm=0
$time=20,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 6 sec AM=0,Alarm=0
$time=21,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 7 sec AM=0,Alarm=0
$time=22,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 8 sec AM=0,Alarm=0
$time=23,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 1 9 sec AM=0,Alarm=0
$time=24,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 0 sec AM=0,Alarm=0
$time=25,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 1 sec AM=0,Alarm=0
$time=26,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 2 sec AM=0,Alarm=0
$time=27,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 3 sec AM=0,Alarm=0
$time=28,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 4 sec AM=0,Alarm=0
$time=29,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 5 sec AM=0,Alarm=0
$time=30,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 6 sec AM=0,Alarm=0
$time=31,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 7 sec AM=0,Alarm=0
$time=32,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 2 8 sec AM=0,Alarm=0
$time=33,clock=1,reset=0,enable=1,r m=0,r time=0001001000000001, 1 2 hrs: 0 0 min: 2 9 sec AM=0,Alarm=0
$time=34,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 0 sec AM=0,Alarm=0
$time=35,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 1 sec AM=0,Alarm=0
$time=36,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 2 sec AM=0,Alarm=0
$time=37,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 3 sec AM=0,Alarm=0
$time=38,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 4 sec AM=0,Alarm=0
$time=39,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 5 sec AM=0,Alarm=0
$time=40,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 6 sec AM=0,Alarm=0
$time=41,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 7 sec AM=0,Alarm=0
$time=42,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 8 sec AM=0,Alarm=0
$time=43,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 3 9 sec AM=0,Alarm=0
$time=44,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 0 sec AM=0,Alarm=0
$time=45,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 1 sec AM=0,Alarm=0
$time=46,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 2 sec AM=0,Alarm=0
$time=47,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 3 sec AM=0,Alarm=0
$time=48,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 4 sec AM=0,Alarm=0
$time=49,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 5 sec AM=0,Alarm=0
$time=50,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 6 sec AM=0,Alarm=0
$time=51,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 7 sec AM=0,Alarm=0
$time=52,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 8 sec AM=0,Alarm=0
$time=53,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 4 9 sec AM=0,Alarm=0
$time=54,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 0 sec AM=0,Alarm=0
$time=55,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 1 sec AM=0,Alarm=0
$time=56,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 2 sec AM=0,Alarm=0
$time=57,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 3 sec AM=0,Alarm=0
$time=58,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 4 sec AM=0,Alarm=0
$time=59,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 5 sec AM=0,Alarm=0
$time=60,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 6 sec AM=0,Alarm=0
$time=61,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 7 sec AM=0,Alarm=0
$time=62,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 8 sec AM=0,Alarm=0
$time=63,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 0 min: 5 9 sec AM=0,Alarm=0
$time=64,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 0 sec AM=0,Alarm=0
$time=65,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 1 sec AM=0,Alarm=1
$time=66,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 2 sec AM=0,Alarm=1
```

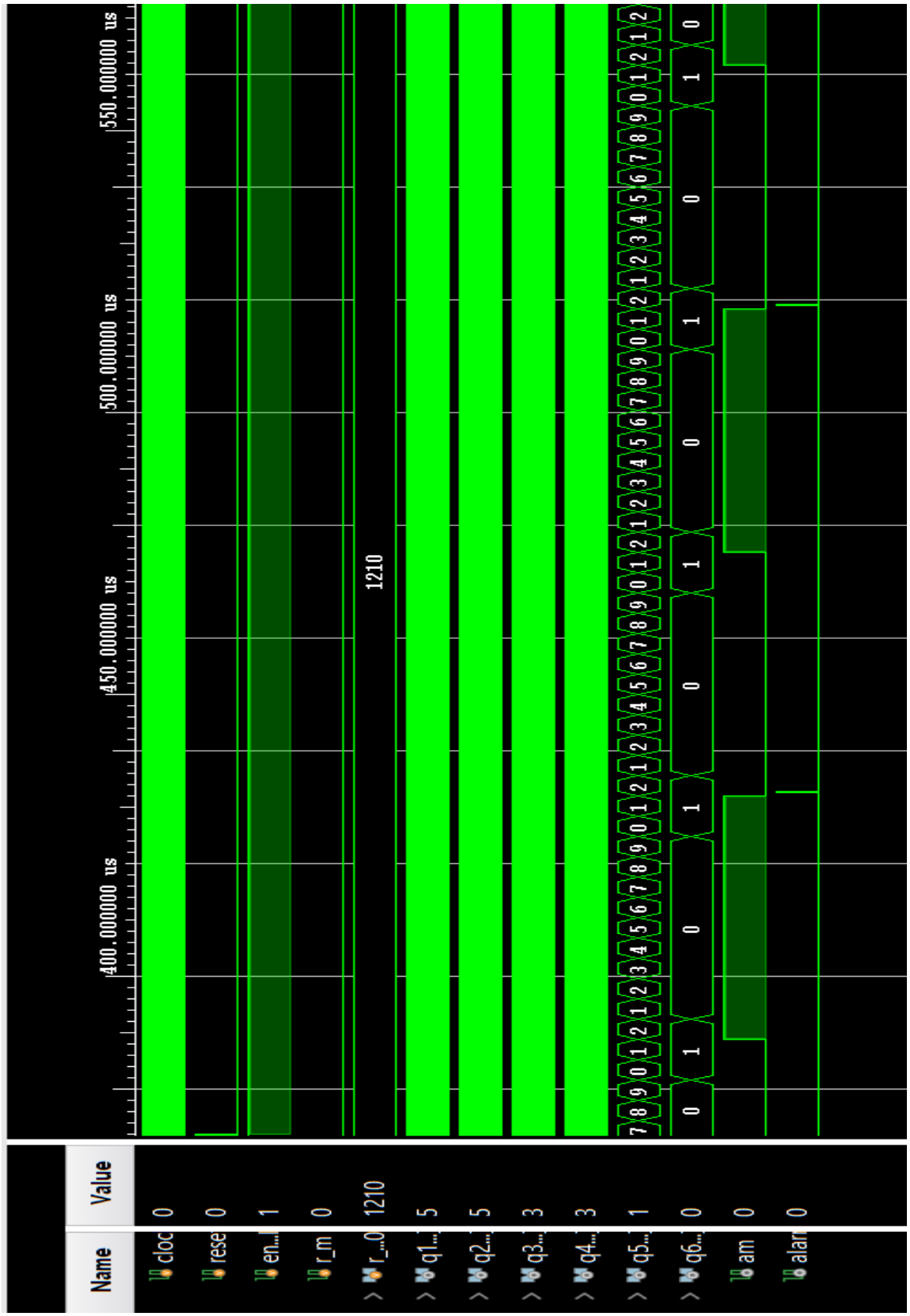
```
$time=67,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 3 sec AM=0,Alarm=1
$time=68,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 4 sec AM=0,Alarm=1
$time=69,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 5 sec AM=0,Alarm=1
$time=70,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 6 sec AM=0,Alarm=1
$time=71,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 7 sec AM=0,Alarm=1
$time=72,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 8 sec AM=0,Alarm=1
$time=73,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 0 9 sec AM=0,Alarm=1
$time=74,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 0 sec AM=0,Alarm=1
$time=75,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 1 sec AM=0,Alarm=1
$time=76,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 2 sec AM=0,Alarm=1
$time=77,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 3 sec AM=0,Alarm=1
$time=78,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 4 sec AM=0,Alarm=1
$time=79,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 5 sec AM=0,Alarm=1
$time=80,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 6 sec AM=0,Alarm=1
$time=81,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 7 sec AM=0,Alarm=1
$time=82,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 8 sec AM=0,Alarm=1
$time=83,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 1 9 sec AM=0,Alarm=1
$time=84,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 0 sec AM=0,Alarm=1
$time=85,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 1 sec AM=0,Alarm=1
$time=86,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 2 sec AM=0,Alarm=1
$time=87,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 3 sec AM=0,Alarm=1
$time=88,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 4 sec AM=0,Alarm=1
$time=89,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 5 sec AM=0,Alarm=1
$time=90,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 6 sec AM=0,Alarm=1
$time=91,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 7 sec AM=0,Alarm=1
$time=92,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 8 sec AM=0,Alarm=1
$time=93,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 2 9 sec AM=0,Alarm=1
$time=94,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 0 sec AM=0,Alarm=1
$time=95,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 1 sec AM=0,Alarm=1
$time=96,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 2 sec AM=0,Alarm=1
$time=97,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 3 sec AM=0,Alarm=1
$time=98,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 4 sec AM=0,Alarm=1
$time=99,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 5 sec AM=0,Alarm=1
$time=100,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 6 sec AM=0,Alarm=1
$time=101,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 7 sec AM=0,Alarm=1
$time=102,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 8 sec AM=0,Alarm=1
$time=103,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 3 9 sec AM=0,Alarm=1
$time=104,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 0 sec AM=0,Alarm=1
$time=105,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 1 sec AM=0,Alarm=1
$time=106,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 2 sec AM=0,Alarm=1
$time=107,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 3 sec AM=0,Alarm=1
$time=108,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 4 sec AM=0,Alarm=1
$time=109,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 5 sec AM=0,Alarm=1
$time=110,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 6 sec AM=0,Alarm=1
$time=111,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 7 sec AM=0,Alarm=1
$time=112,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 8 sec AM=0,Alarm=1
$time=113,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 4 9 sec AM=0,Alarm=1
$time=114,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 0 sec AM=0,Alarm=1
$time=115,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 1 sec AM=0,Alarm=1
$time=116,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 2 sec AM=0,Alarm=1
$time=117,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 3 sec AM=0,Alarm=1
$time=118,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 4 sec AM=0,Alarm=1
$time=119,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 5 sec AM=0,Alarm=1
$time=120,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 6 sec AM=0,Alarm=1
$time=121,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 7 sec AM=0,Alarm=1
$time=122,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 8 sec AM=0,Alarm=1
$time=123,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 1 min: 5 9 sec AM=0,Alarm=1
$time=124,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 0 sec AM=0,Alarm=1
$time=125,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 1 sec AM=0,Alarm=0
$time=126,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 2 sec AM=0,Alarm=0
$time=127,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 3 sec AM=0,Alarm=0
$time=128,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 4 sec AM=0,Alarm=0
$time=129,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 5 sec AM=0,Alarm=0
$time=130,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 6 sec AM=0,Alarm=0
$time=131,clock=1,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 7 sec AM=0,Alarm=0
$time=132,clock=0,reset=0,enable=1,r_m=0,r_time=0001001000000001, 1 2 hrs: 0 2 min: 0 8 sec AM=0,Alarm=0
```

TESTBENCH 2: (Hour and AM/PM Transitions)

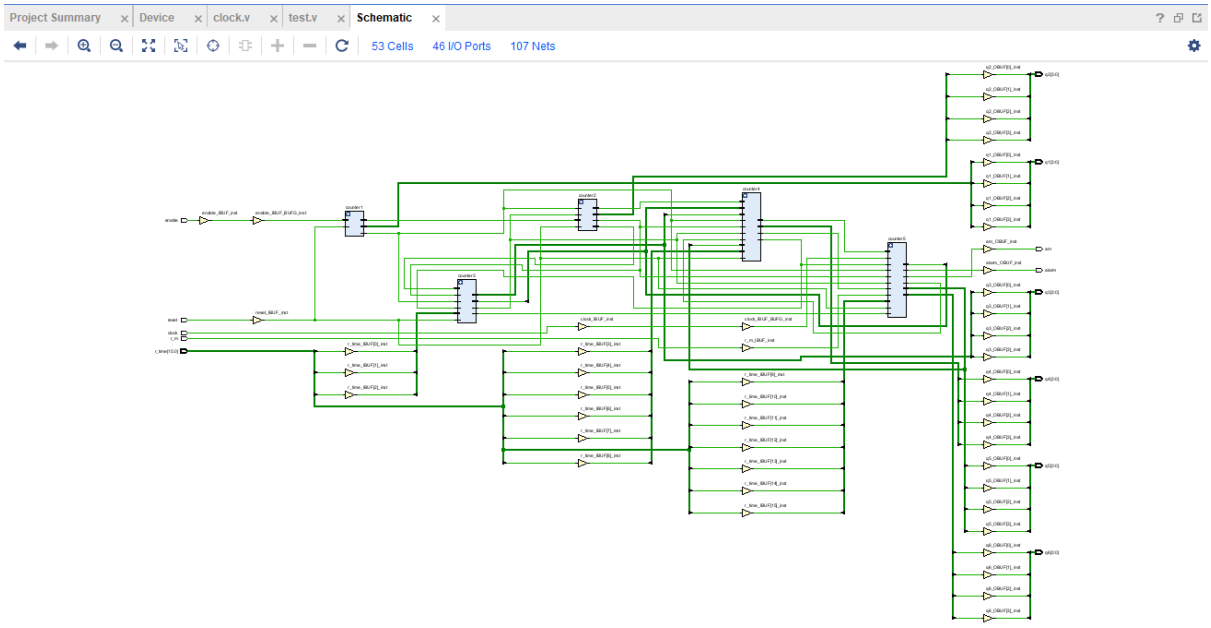
```
module test;
reg clock,reset,enable;
wire [3:0]d2,d1;
wire am;
hour dut(clock,reset,enable,d2,d1,am);
initial
begin
 enable = 1;
 clock=1;
 reset=1;
 #5 reset=0;
 #150 $finish;
end
always #1 clock = ~clock;
initial
$monitor("$time=%g,clock=%b,enable=%b,reset=%b,d2=%d,d1=%d,am=%b"
,$time,clock,enable,reset,d2,d1,am);
endmodule
```



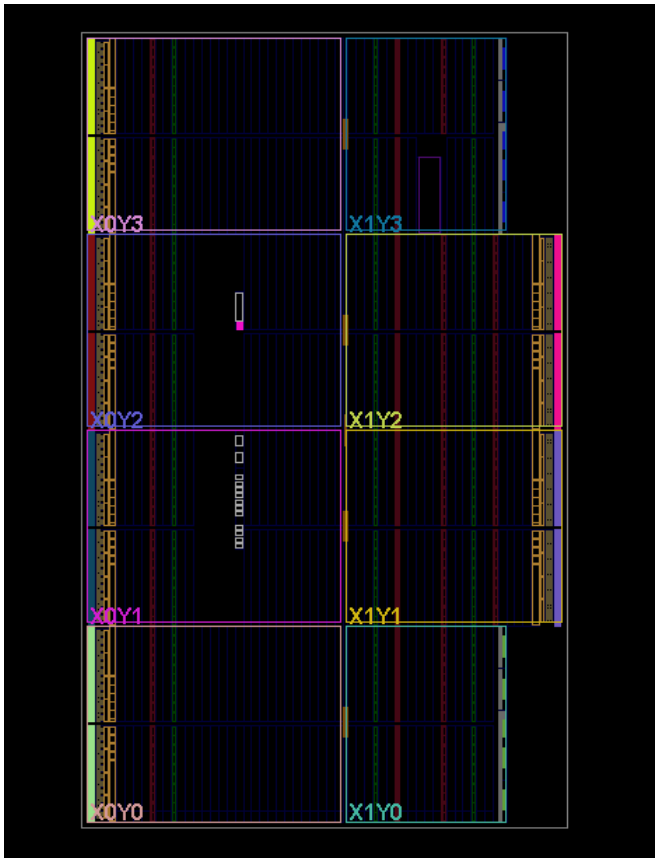
On increasing the simulation time up to 1000us, the waveform generated is attached below. From the same all the transitions of the specifications (hrs, min, sec, am/pm, alarm) are clearly evident.



SCHEMATIC:



IMPLEMENTATION:



FUTURE SCOPE (Other possible implementations)

1. In this project, we have used behavioral modelling for setting mod 60 counter for the functionality of seconds and minutes. Further entire clock module can be designed using gate level modelling i.e. using registers and basic logic gates. But it is a tedious task.
2. Adding a feature to manually set the clock time enhances functionality.
3. display the time on a 7-segment or LCD, creating separate display modules for the different parts (hours, minutes, seconds).
4. Implementing the alarm functionality with a finite state machine (FSM) could offer more flexibility, especially if we want to add features like multiple alarms, snooze, or different sounds in the future.

Conclusion:

The digital clock project successfully demonstrated the implementation of a timekeeping system in Verilog, showcasing key concepts in digital design and hardware description languages. Through careful planning, modular design, and rigorous testing, the project achieved its primary goals of accurately displaying time in a 12-hour format with AM/PM indicators and providing an alarm functionality.