

# Lab0

## 实验题目

希望完成一个时分秒时钟，该时钟在每次 clk 上升沿时秒位加 1（clk 信号不必以秒为周期），满 20 后清零，分位加 1；分位满 10 后清零，时位加 1；时位满 5 后三个位全部清零，如此循环。也就是说，我们设计的时钟一分钟只有 20 秒，一小时只有 10 分钟，一天只有五小时。请尝试使用模块化的设计方法完成该时钟，并给出所有模块输入输出的仿真波形。

## 实验代码

设计文件：

```
1  `timescale 1ns / 1ps
2  //PB21111723王涵
3  module Clock(
4      input clk,
5      input rstn,
6      output [2:0] hour,
7      output [3:0] min,
8      output [4:0] sec
9  );
10
11  wire en1, en2;
12
13  Sec sec1(
14      .clk(clk),
15      .sec_reset(rstn),
16      .dout(sec),
17      .co(en1)
18  );
19
20  Min min1(
21      .en(en1),
22      .clk(clk),
23      .min_reset(rstn),
24      .dout(min),
25      .co(en2)
26  );
27
28  Hour hour1 (
29      .clk(clk),
```

```

30         .hour_rstn(rstn),
31         .en1(en1),
32         .en2(en2),
33         .dout(hour)
34     );
35
36 endmodule
37
38 module Sec(
39     input clk,
40     input sec_reset,
41     output reg [4:0] dout,
42     output co
43 );
44
45 always@(posedge clk or posedge sec_reset)
46 begin
47     if(sec_reset)
48     begin
49         dout <= 5'b00000;
50     end
51     else if(dout == 5'b10011)
52     begin
53         dout <= 5'b00000;
54     end
55     else
56     begin
57         dout <= dout + 1'b1;
58     end
59 end
60 assign co = dout[0] & dout[1] & dout[4];
61 endmodule
62
63 module Min(
64     input en,
65     input clk,
66     input min_reset,
67     output reg [3:0] dout,
68     output co
69 );
70
71 always@(posedge clk or posedge min_reset)
72 begin
73     if(min_reset)
74     begin
75         dout <= 4'b0000;
76     end

```

```

77     else if(en)
78     begin
79         if(dout == 4'b1001)
80         begin
81             dout <= 4'b0000;
82         end
83         else
84         begin
85             dout <= dout + 1'b1;
86         end
87     end
88 end
89 assign co = dout[0] & dout[3];
90 endmodule
91
92 // 时模块
93 module Hour (
94     input clk,
95     input hour_rstn,
96     input en1,en2,
97     output reg [2:0] dout
98 );
99
100 always @(posedge clk or posedge hour_rstn)
101 begin
102     if (hour_rstn)
103     begin
104         dout <= 3'b000;
105     end
106     else if (en1 && en2 && (dout == 3'b100))
107     begin
108         dout <= 3'b000;
109     end
110     else if (en2 && en1)
111     begin
112         dout <= dout + 1'b1;
113     end
114 end
115 endmodule
116

```

仿真测试文件(testbench.v由助教提供)

```

1  `timescale 1ns / 1ps
2  //////////////////////////////////////
   ///

```

```
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 2023/03/20 09:51:18
7 // Design Name:
8 // Module Name: tb_Clock
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////
21 ///
22
23 module tb_Clock(
24
25 );
26
27 // Clock Parameters
28 parameter PERIOD = 10;
29
30
31 // Clock Inputs
32 reg clk = 0 ;
33 // Clock Reset
34 reg rstn = 1 ;
35
36 // Clock Outputs
37 wire [2:0] hour;
38 wire [3:0] min;
39 wire [4:0] sec;
40
41 initial begin
42     #10 rstn = 0;
43 end
44
45 initial
46 begin
47     forever #(PERIOD/2) clk=~clk;
48 end
```

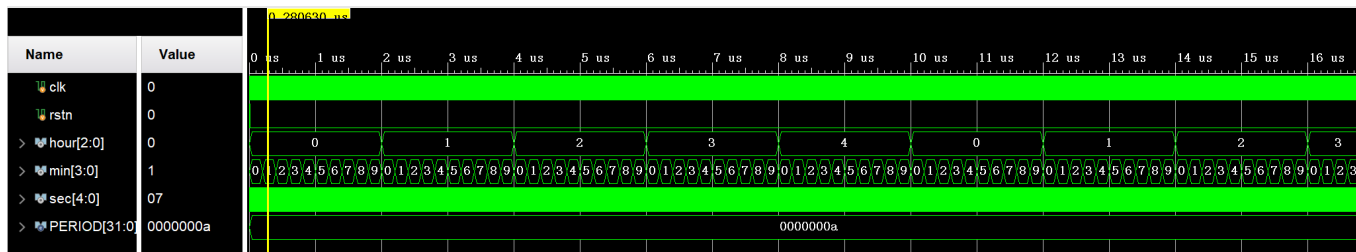
```

49
50 Clock u_Clock (
51     .clk           ( clk     ),
52     .rstn          ( rstn    ),
53     .hour          ( hour    ),
54     .min           ( min     ),
55     .sec           ( sec     )
56 );
57
58
59 // set simulation time
60 initial
61 begin
62     #(PERIOD * 2000);
63     $finish;
64 end
65
66 endmodule
67

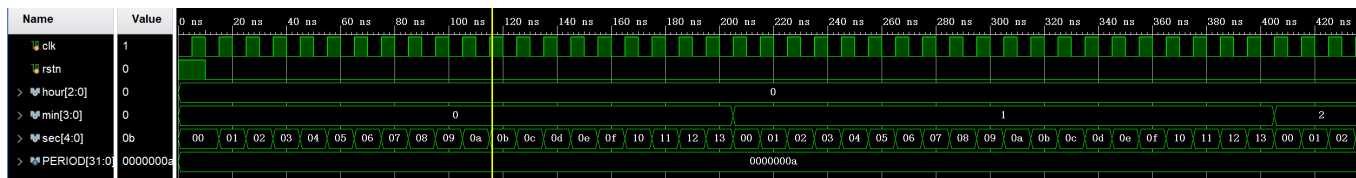
```

## 仿真波形

下图的仿真，1 day = 5hour，1 hour = 10 min的关系清晰可见。

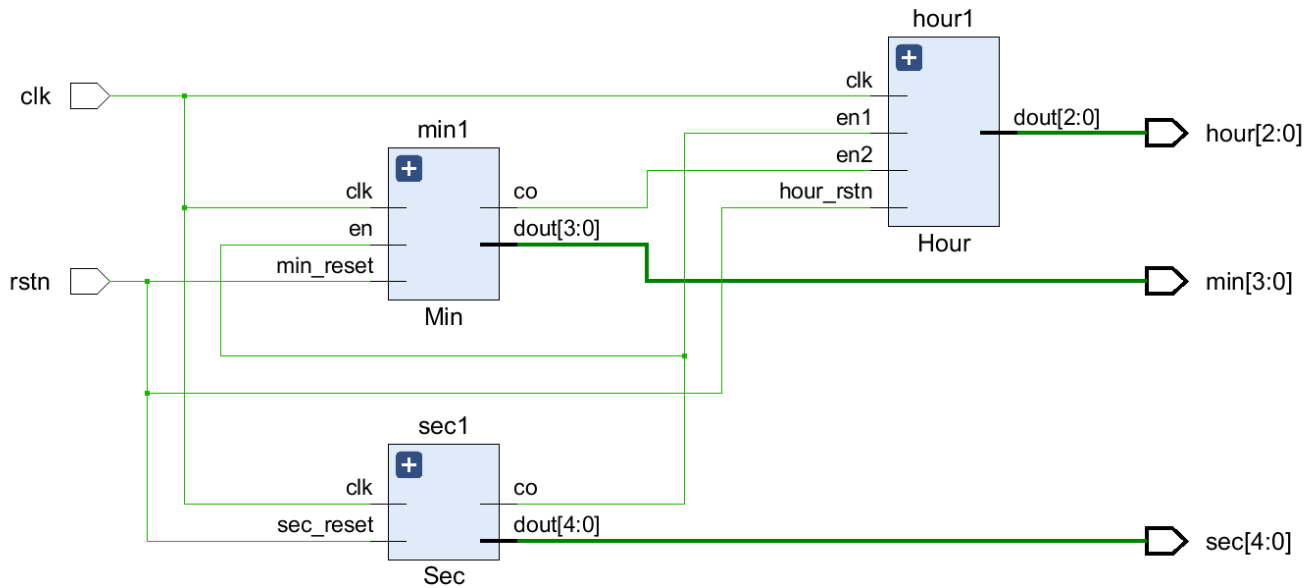


下图更加精细化，显示1 min = 20 sec。



## 电路图

运行 RTL Analysis 任务下的 Open Elaborated Design的到vivado生成的电路图，如下



## 实验反馈

难度：本次实验难度一般

收获：学会了将vivado与vscode关联，并温习了上学期数电实验的部分操作。

建议：希望助教给的实验文件再详细一点，比如配置过程并不完善，最后还要Google或CSDN。