LAB02_312651057

吳鴻明

實驗目的

透過RTL coding實現24小時制的電子鐘，熟悉記憶元件D-Flip-Flop，以及如何將50MHZ的Clock降為1HZ，實現1秒步進的時鐘

實驗程式碼

檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) ··· LAB02

time_bg.v timer.v

time_bg.v

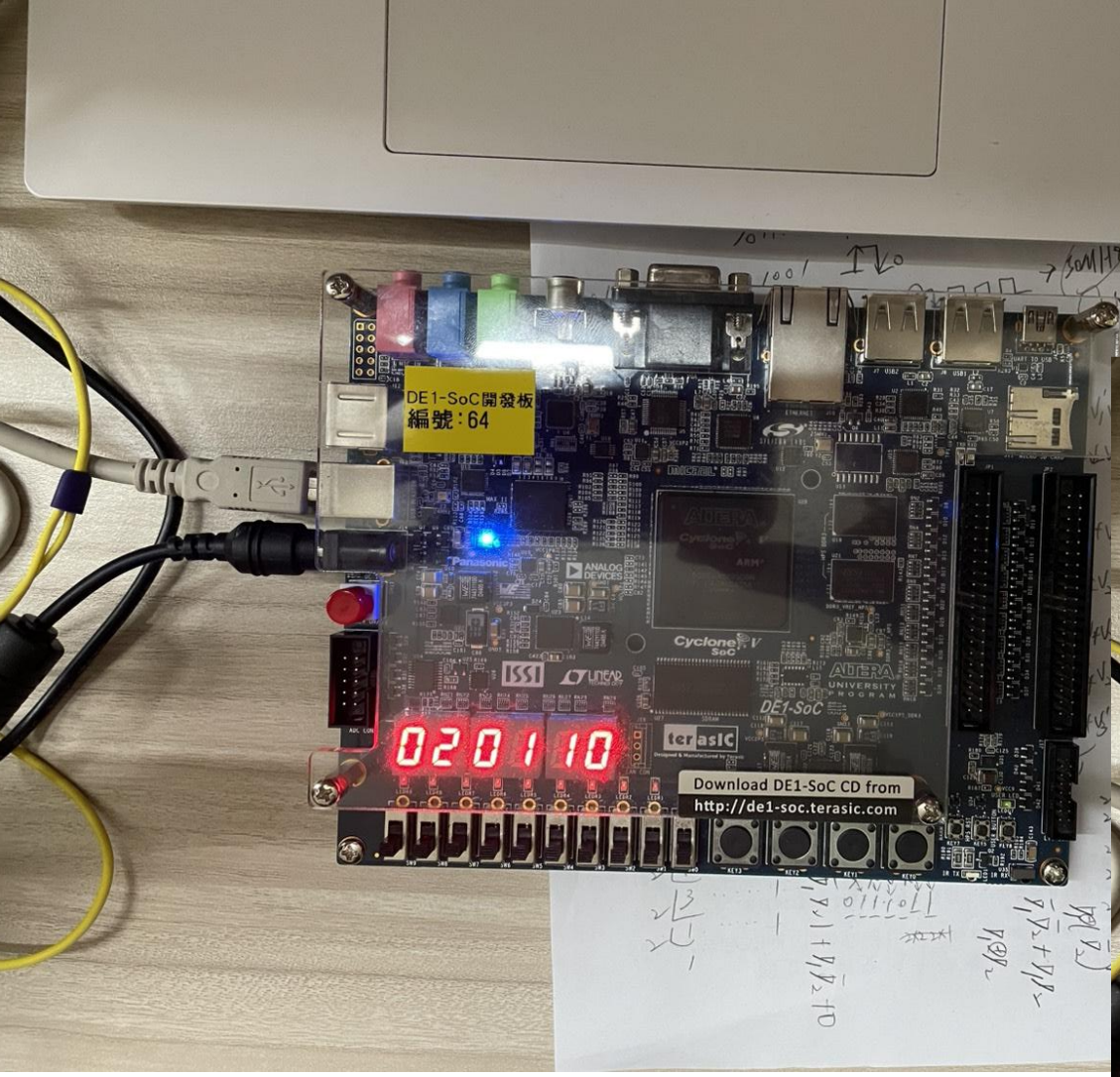
```
/*Original code by <HONG-MING-WU>*/
module clock(CLOCK_50,SW,HEX0,HEX1,HEX2,HEX3,HEX4,HEX5);
input CLOCK_50;
input [9:0]SW;
output reg [6:0]HEX0,HEX1,HEX2,HEX3,HEX4,HEX5;

//reg declare
reg[26:0]sys_cnt; //use to count 1 second
reg[5:0]sec_cnt; //use to count real second
reg[7:0]min_cnt;//use to count minute
reg[7:0]hr_cnt;//use to count hour
reg SW8_ff;
//wire declare
wire SW8_n;

//delay one cycle
always@(posedge CLOCK_50)begin
    SW8_ff <= SW[8];
end

//
always@(posedge CLOCK_50 )begin
    if ((SW[9]==1'b1)&&(SW8_ff==1'b0)&&(SW[8]==1'b1))begin
        hr_cnt <= SW[7:0];
    end
    else if ((SW[9]==1'b0)&&(SW8_ff==1'b0)&&(SW[8]==1'b1))begin
        min_cnt <= SW[7:0];
    end
    //when sys_cnt reaches 50000000, it means 1 second has passed
    else if(sys_cnt < 50000000)begin
        sys_cnt <= sys_cnt+26'b1;
    end

    else begin
        sys_cnt <= 0;
        if ((sec_cnt == 6'd59)&&(min_cnt == 6'd59)&&(hr_cnt == 6'd23))begin
            sec_cnt <= 0;
            min_cnt <= 0;
            hr_cnt <= 0;
        end
        else if ((sec_cnt == 6'd59)&&(min_cnt == 6'd59))begin
            sec_cnt <= 0;
            min_cnt <= 0;
            hr_cnt <= hr_cnt+1;
        end
        else if ((sec_cnt == 6'd59))begin
            sec_cnt <= 0;
            min_cnt <= min_cnt+1;
```

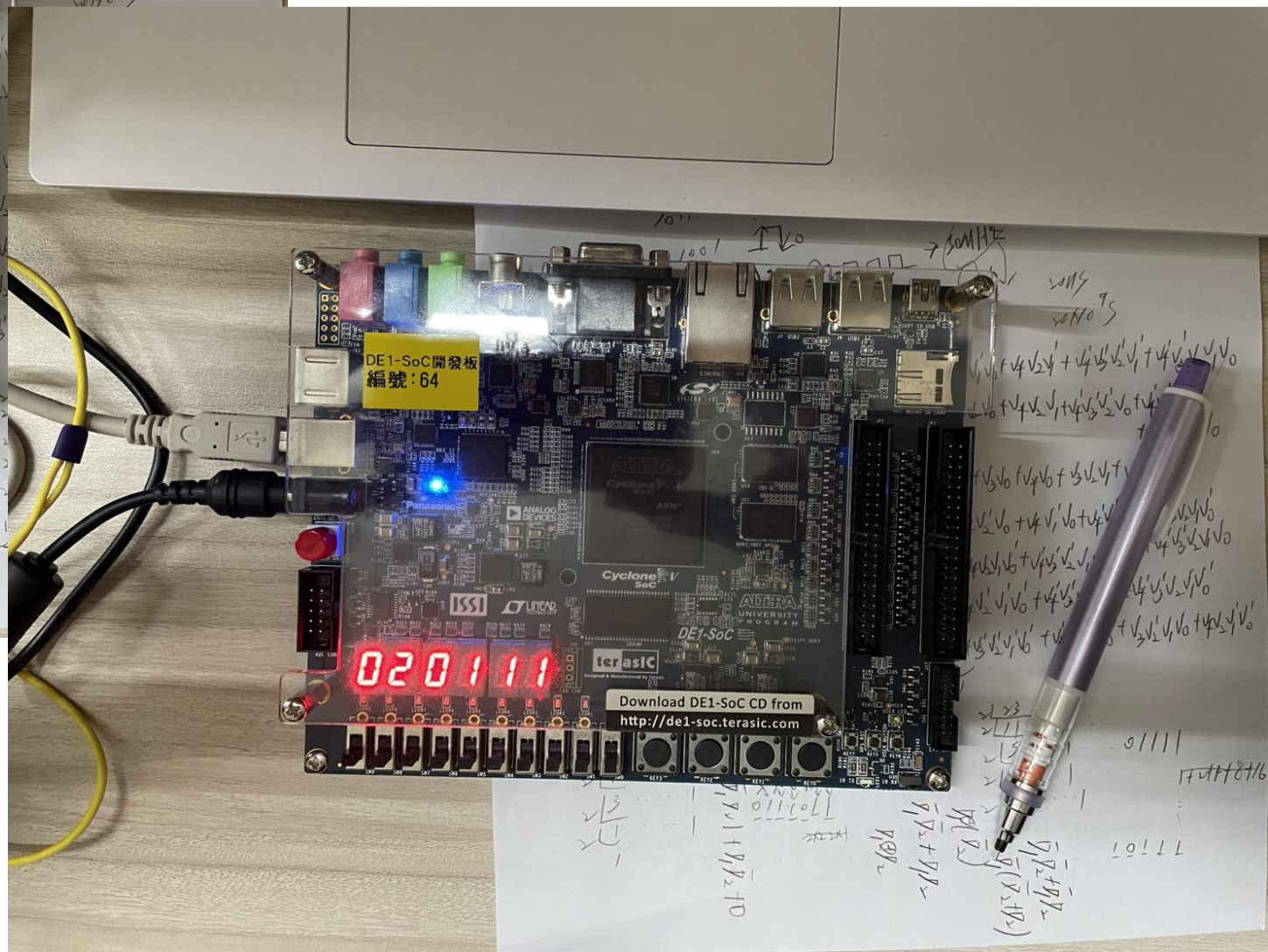
time_bg.v timer.v

time_bg.v

```
22   always@(posedge CLOCK_50 )begin
34       else begin
45           end
46           else if ((sec_cnt == 6'd59))begin
47               sec_cnt <= 0;
48               min_cnt <= min_cnt+1;
49           end
50           else begin
51               sec_cnt <= sec_cnt+1;
52           end
53       end
54   end
55
56
57   /*The reason why I didn't use this part
58   ,Because I didn't solve the problem of pull down set_valid signal.*/
59
60   /*always@(posedge SW[8])begin
61       if (SW[9]==1'b1)begin
62           set_hr <= SW[7:0];
63           set_valid <= 1'b1;
64       end
65       else if (SW[9]==1'b0)begin
66           set_min <= SW[7:0];
67           set_valid <= 1'b1;
68       end
69   end*/
70
71   //sec display
72   always@(*)begin···
359  end
360
361  //min display
362  always@(*)begin···
654  end
655
656  //hour diplay
657  always@(*)begin···
773  end
774  endmodule
```

實驗結果照片


DE1-SoC開發板
編號：64
020110
Download DE1-SoC CD from
http://de1-soc.terasic.com


DE1-SoC開發板
編號：64
020111
Download DE1-SoC CD from
http://de1-soc.terasic.com

問題與討論

討論 ： 針對本次LAB，比上次的難了一點，但還可以接受，主要原因是多了記憶元件，所以衍伸出了很多問題。

問題 ： 左下圖是我在coding時遇到的問題，我如果設定小時與分鐘後，都會把set_valid設為1，但這就是問題點，如果我設為1，我就無法在always block裡面把他拉下來變0，也有可能是我沒想到，但最後還是換了一個寫法，如實驗程式碼那邊所張貼的。

```
/*The reason why I didn't use this part
,Because I didn't solve the problem of pull down set_valid signal.*/

/*always@(posedge SW[8])begin
    if (SW[9]==1'b1)begin
        set_hr <= SW[7:0];
        set_valid <= 1'b1;
    end
    else if (SW[9]==1'b0)begin
        set_min <= SW[7:0];
        set_valid <= 1'b1;
    end
end*/
```