實驗四

FSM (finite state machine)

學生: 謝旻錡

學號: 313512078

日期: 2025/03/23

1. 實驗目的

學習有限元素機 FSM 架構，並學習其應用。

2. 實驗程式碼

Part4 demo 題，使用 case 架構的 FSM 寫一個 counter-like circuit。其功能為 0～9 的數字循環，FSM 擁有四種狀態，1.保持原狀 2.數字加一 3.數字加二 4.數字減一。

Table 1. input /output

Input		Output	
SW_2	Input(w1)	HEX0	Count(0~9cycle)
SW_1	Input(w2)		
SW_1	Reset		
KEY_0	clk		

程式架構，分兩部，狀態機及 7 段顯示器。

- 狀態機 FSM

  I. 設定 FSM 的狀態推移。State 為當下狀態 next_state 為下一個狀態。

```
always @(negedge clk)begin
    if(reset)
        state <= A;
    else
        state <= next_state;
end
```

Figure 1

狀態機由 clk 負緣觸發 (negedge clk)。

當 reset 為 True，狀態更新為 A 狀態。當 reset 為 Faulse，FSM 狀態更新。

II. 定義 0~9 個狀態及其邏輯電路。

```
parameter A = 4'd0;
parameter B = 4'd1;
parameter C = 4'd2;
parameter D = 4'd3;
parameter E = 4'd4;
parameter F = 4'd5;
parameter G = 4'd6;
parameter H = 4'd7;
parameter I = 4'd8;
parameter J = 4'd9;
```

Figure 2. 宣告 0~9 狀態變數

下圖程式 Figure 3, Figure 4, Figure 5, Figure 6，為 0~9 (A~J) 狀態的邏輯電路。sw 為判斷條件，當 sw 為: 00 = 保持原狀態; 01 = 狀態加一; 10 = 狀態加二; 11: 狀態減一。

```
always @(*)begin
    sw = SW[2:1];
    case(state)
        A:begin
            if(sw == 2'b00)
                next_state = A;
            else if(sw == 2'b01)
                next_state = B;
            else if(sw == 2'b10)
                next_state = C;
            else
                next_state = J;
        end
        B:begin
            if(sw == 2'b00)
                next_state = B;
            else if(sw == 2'b01)
                next_state = C;
            else if(sw == 2'b10)
                next_state = D;
            else
                next_state = A;
        end
        C:begin
            if(sw == 2'b00)
                next_state = C;
            else if(sw == 2'b01)
                next_state = D;
            else if(sw == 2'b10)
                next_state = E;
            else
                next_state = B;
        end
```

Figure 3

```
        D:begin
            if(sw == 2'b00)
                next_state = D;
            else if(sw == 2'b01)
                next_state = E;
            else if(sw == 2'b10)
                next_state = F;
            else
                next_state = C;
        end
        E:begin
            if(sw == 2'b00)
                next_state = E;
            else if(sw == 2'b01)
                next_state = F;
            else if(sw == 2'b10)
                next_state = G;
            else
                next_state = D;
        end
        F:begin
            if(sw == 2'b00)
                next_state = F;
            else if(sw == 2'b01)
                next_state = G;
            else if(sw == 2'b10)
                next_state = H;
            else
                next_state = E;
        end
```

Figure 4

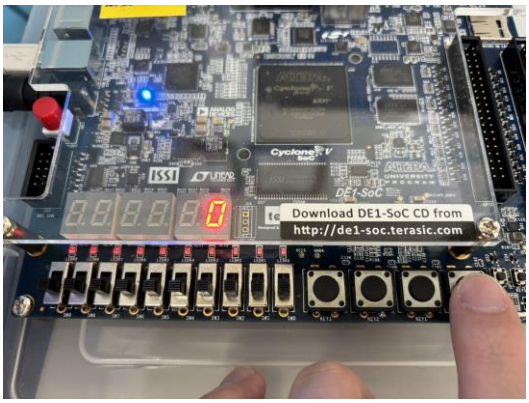
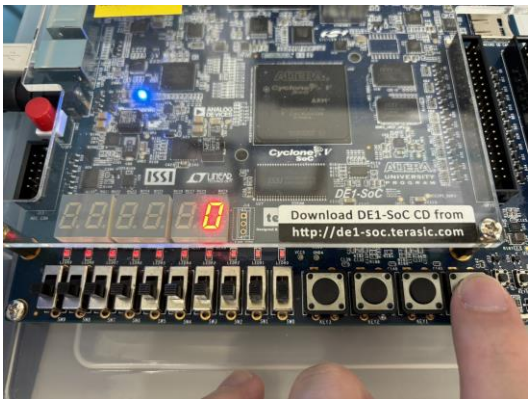
```
        G:begin
            if(sw == 2'b00)
                next_state = G;
            else if(sw == 2'b01)
                next_state = H;
            else if(sw == 2'b10)
                next_state = I;
            else
                next_state = F;
        end
        H:begin
            if(sw == 2'b00)
                next_state = H;
            else if(sw == 2'b01)
                next_state = I;
            else if(sw == 2'b10)
                next_state = J;
            else
                next_state = G;
        end
        I:begin
            if(sw == 2'b00)
                next_state = I;
            else if(sw == 2'b01)
                next_state = J;
            else if(sw == 2'b10)
                next_state = A;
            else
                next_state = H;
        end
```

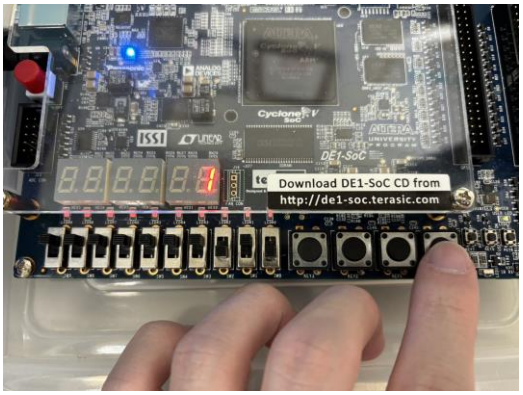
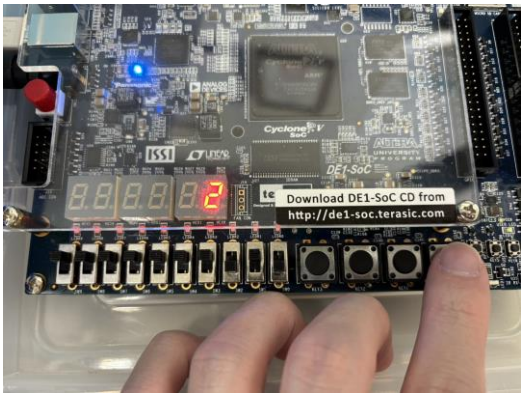
Figure 5

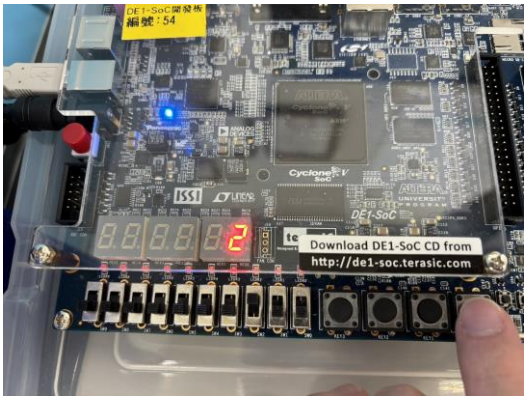
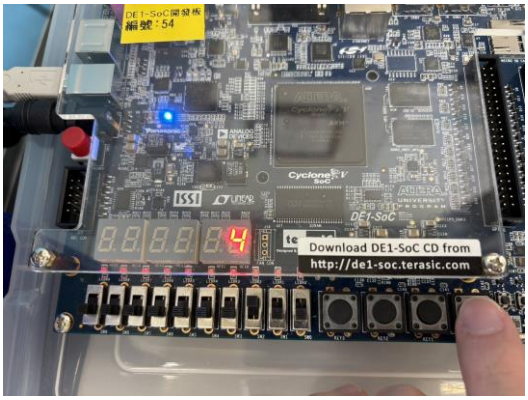
```
        J:begin
            if(sw == 2'b00)
                next_state = J;
            else if(sw == 2'b01)
                next_state = A;
            else if(sw == 2'b10)
                next_state = B;
            else
                next_state = I;
        end
    endcase
end
```

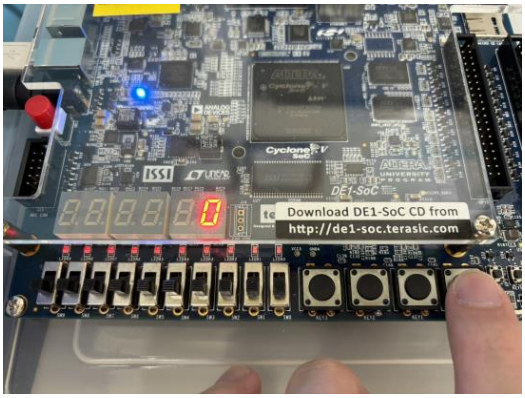
Figure 6

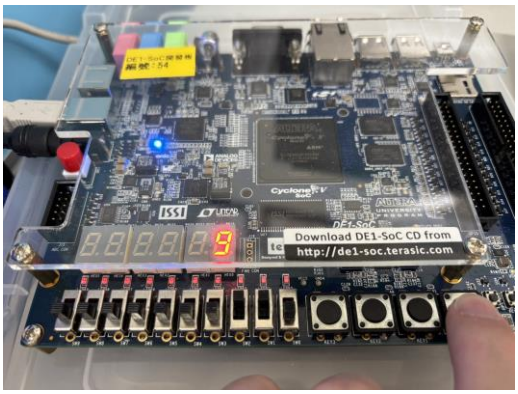
3. 實驗結果照片(optional)

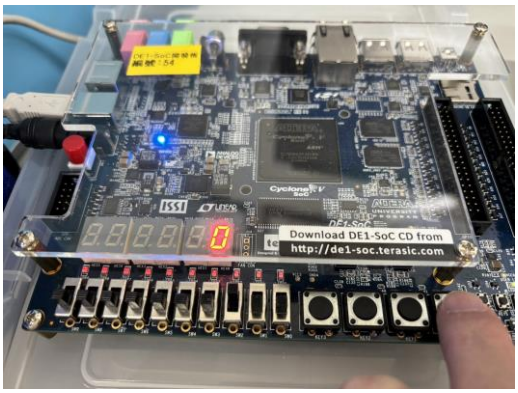
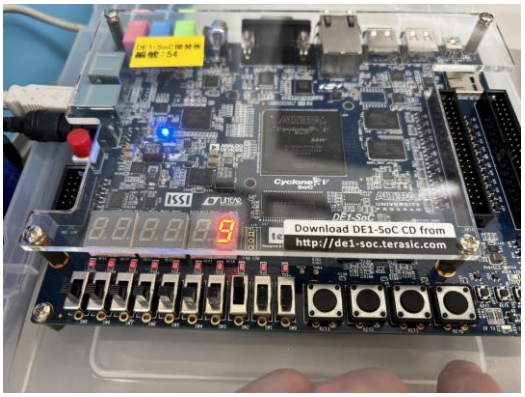
sw = 00 (狀態不變)	
	

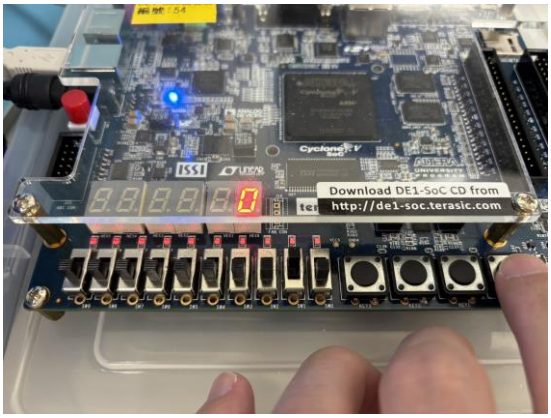
sw = 01 (狀態加一)	
	

sw = 10 (狀態加二)	
	

sw = 11 (狀態減一)	
	

0 – 1 => 9	
	

8 + 2 => 0, 9 + 2 => 1	
	
	

9 + 1 => 0	
	

4. RTL 布局(optional)

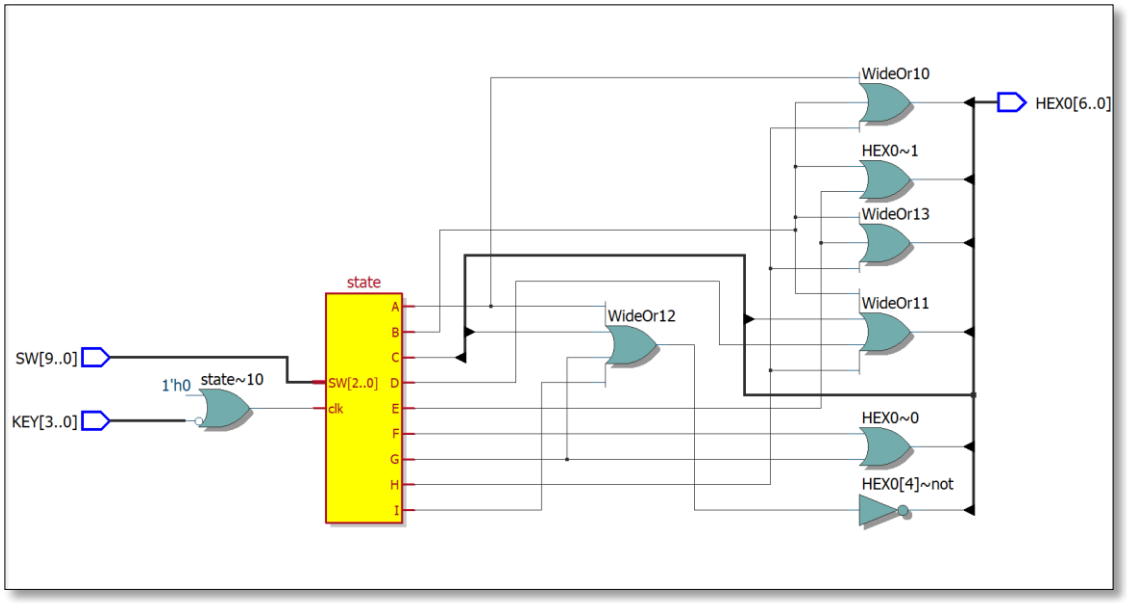


Figure 7. RTL circuit

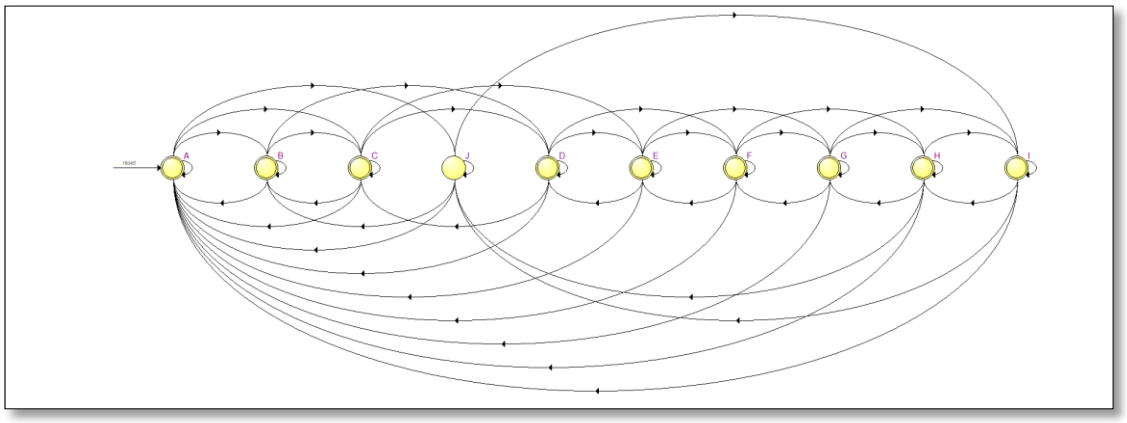


Figure 8. FSM state diagram

5. 問題與討論

這次沒時間，只做了 demo 題，而 demo 題沒遇到什麼問題。很快就做完了。