

Lab 8: Latch and Flip-flop

1. 實驗目的

了解基本的循序邏輯電路及 Latch 和 Flip-flop 等電路之操作原理，使用 Verilog 語言設計電路，並應用 CAD 軟體來輔助驗證所設計的電路。

2. 實驗器材

Quartus II (CAD tools) 、 DE2

3. 實驗內容

請使用 Verilog HDL 寫出：

(a) 1-bit D latch

(b) 負邊緣觸發之 D Flip-flop

(c) 正邊緣觸發之 T Flip-flop

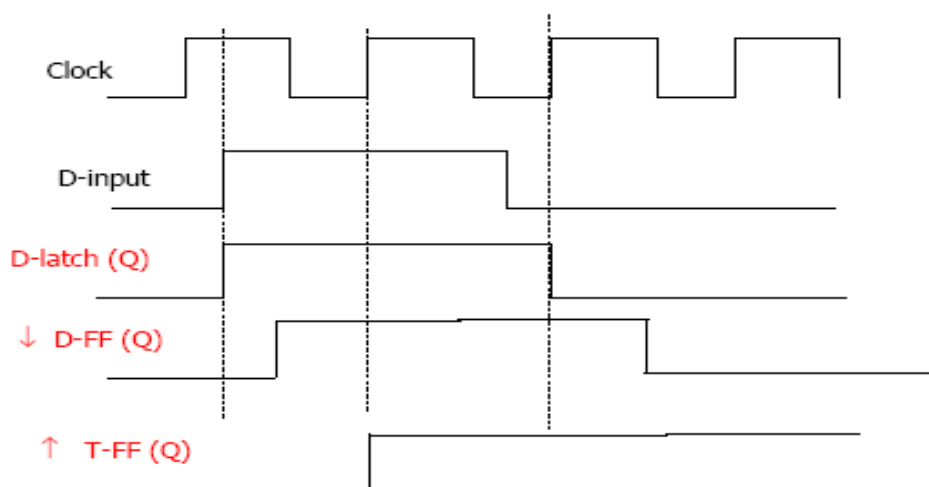


圖8.1

於 Quartus II 與 DE2 模擬訊號波型加以驗證結果。(利用 DE2 實驗板上的 FPGA 晶片 CycloneII 系列 EP2C35F672C6)

4. 實驗原理

(a) SR Latch with Control Input

Latch 與 Flip-flop 均是數位電路中可以提供位元狀態儲存的裝置，它可以將邏輯狀態「0」或「1」存放在裝置內直到位元值需要改變或電源被切除。由於有兩個穩定的輸出狀態，所以被稱為雙穩態電路。下圖8.2 (a) 為 SR Latch with

Control Input，圖8.2 (b) 為其 Function table。Latch 是由 control input 來選擇改變其狀態，稱為觸發 (trigger)，而此處是用 Level trigger 的方式，如圖8.3。

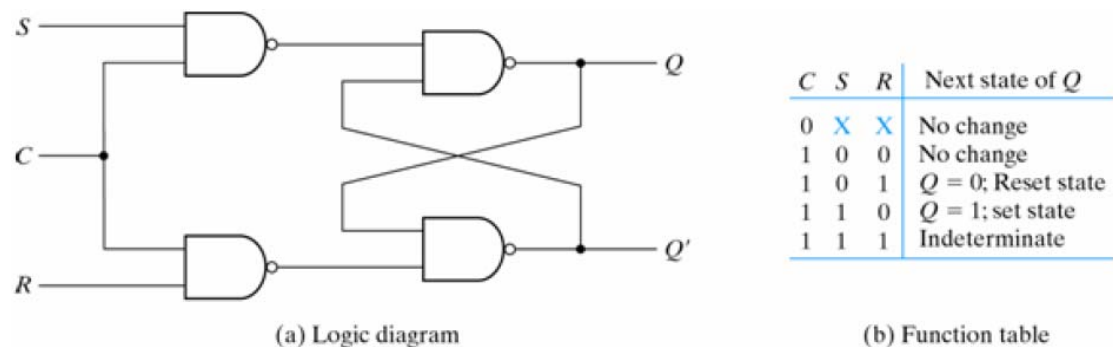


圖 8.2 SR Latch

Source: Mano, "Digital Design," Prentice Hall.



圖8.3 Response to positive level

Source: Mano, "Digital Design," Prentice Hall.

(b) Master-slave D Flip-flop

下圖8.4 為Master-slave D Flip-flop，圖8.5 為其中的D latch。當輸入之CLK 為High 時，master D latch 運作，而slave D latch 維持不變；但當CLK 為由High 轉為Low 時，slave D latch 運作，將值由Q 輸出，negative-edge-triggered flip-flop。

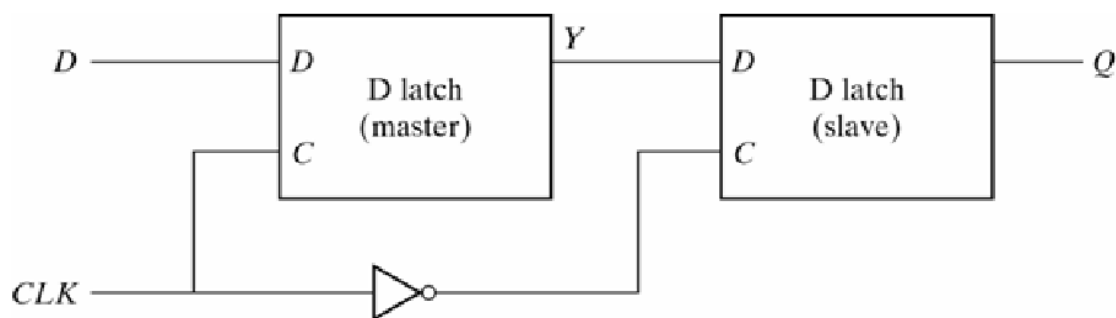


圖8.4 Master-slave D Flip-flop

Source: Mano, "Digital Design," Prentice Hall.

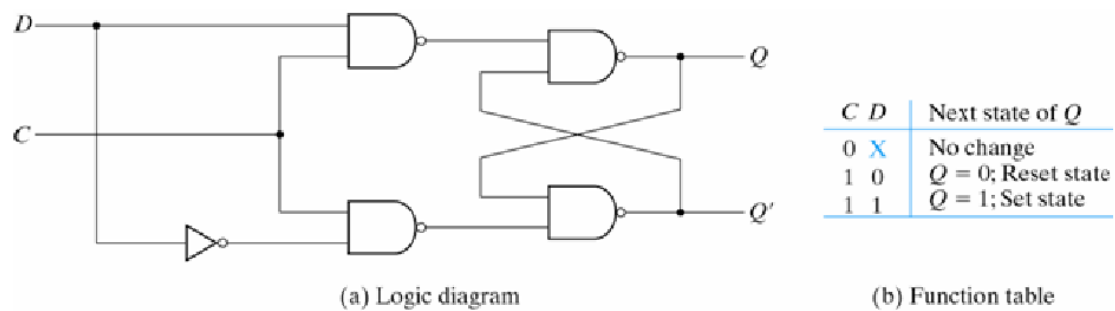


圖8.5 D latch

Source: Mano, "Digital Design," Prentice Hall.

如圖 8.6 所示，當 CLK 由 High 變 Low 時，D 值為 1 則作 set 的動作，輸出 Q 之值 set 為 1；若 D 為 0，當 CLK 由 High 變 Low 時，Q 之值 reset 為 0。圖 8.7 為 D Flip-flop 的 Characteristic Table。

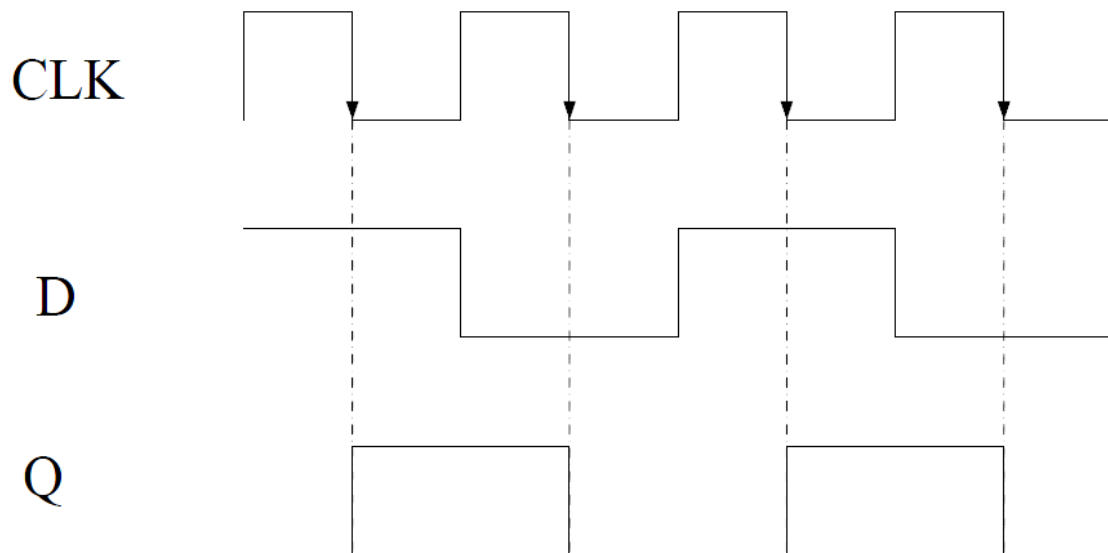


圖8.6 Negative-edge-triggered response

D	Q (t + 1)
0	0 (Reset)
1	1 (Set)

圖8.7 Characteristic Table of D Flip-flop

(c) T Flip-flop :

如下圖 8.8 所示，由 D Flip-flop， $D = T \oplus Q = TQ' + T'Q$ 。D Flip-flop 為正源觸發，當輸入 T 為 0 時， $Q(t+1) = Q(t)$ ，保持原狀不變；當 CLK 正源觸發，輸入之 T 為 1 時， $Q(t+1) = Q'(t)$ ，原本狀態變補。下圖 8.9 為 T Flip-flop 之時序圖，圖 8.10 為 T Flip-flop 的 Characteristic Table。

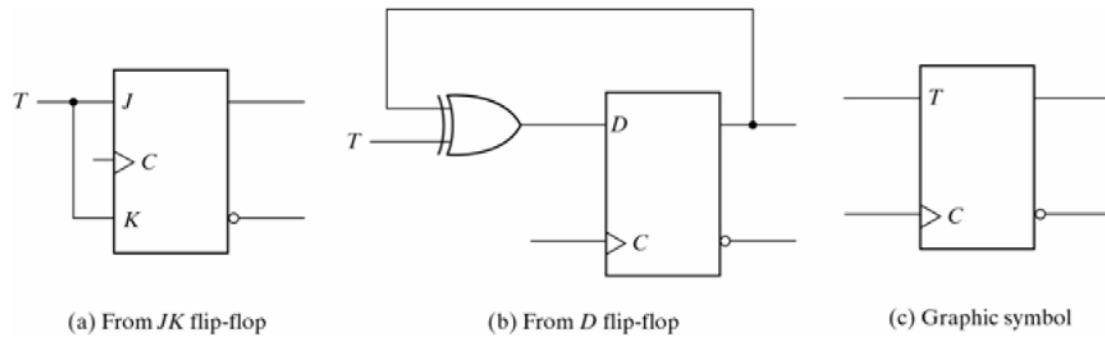


圖 8.8 T Flip-flop

Source: Mano, "Digital Design," Prentice Hall.

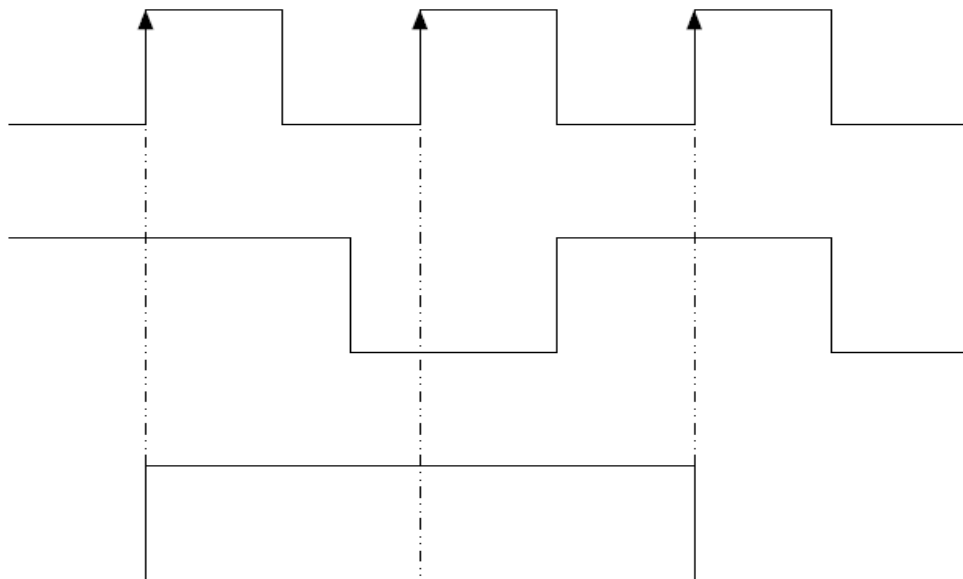


圖 8.9 T Flip-flop 之時序圖

T	$Q(t+1)$
0	$Q(t)$ (No change)
1	$Q'(t)$ (Complement)

圖 8.10 Characteristic Table of T Flip-flop

5. 實驗範例與結果：

(1) D Latch:

```
//to de2
module D_l(seg7_out,D,control);
input D;
input control;
output [6:0] seg7_out;
wire ddlatch_out;
ddlatch test1(ddlatch_out, D, control);
binary_to_7seg test2(ddlatch_out,seg7_out);
endmodule
```

```
//D latch
module ddlatch(Q,D,control);
input D;
input control;
output Q;
reg Q;
always @(control or D)
if (control) Q = D;
endmodule
```

```
//DE2_7seg
module binary_to_7seg(binary_in,seg7_out);
input [3:0] binary_in;
output [6:0] seg7_out;
reg [6:0] seg7_out;
always @ (binary_in)
begin
case(binary_in)
4'h0: seg7_out = 7'b1000000;
4'h1: seg7_out = 7'b1111001;
4'h2: seg7_out = 7'b0100100;
4'h3: seg7_out = 7'b0110000;
4'h4: seg7_out = 7'b0011001;
```

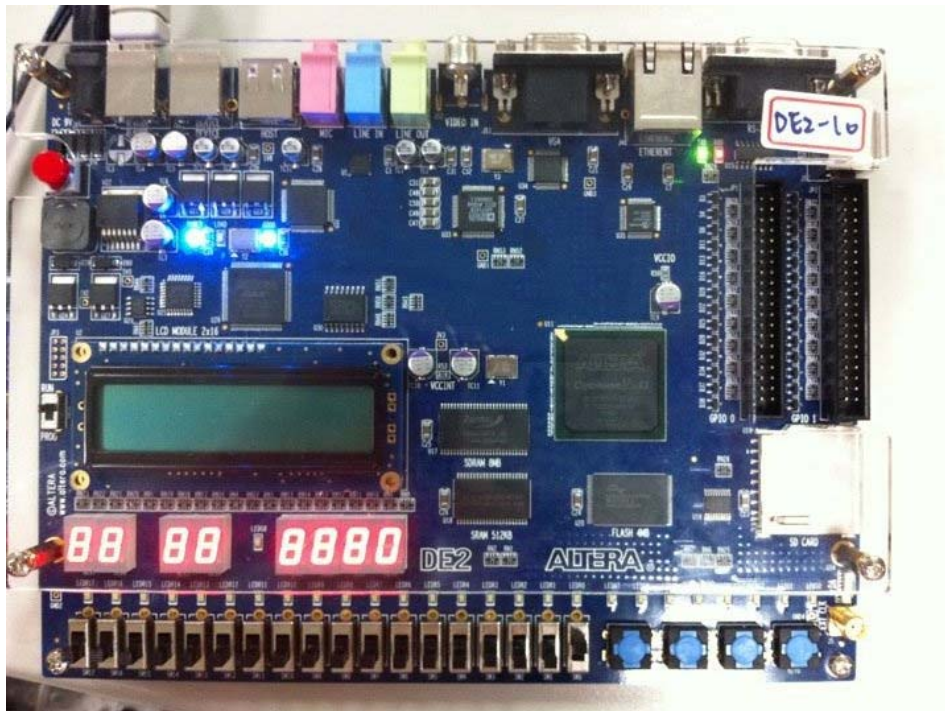
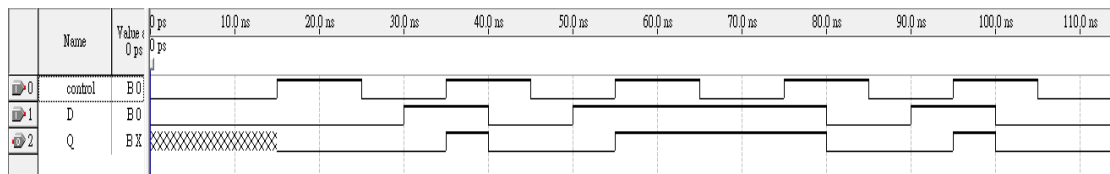
```

4'h5: seg7_out = 7'b0010010;
4'h6: seg7_out = 7'b0000010;
4'h7: seg7_out = 7'b1111000;
4'h8: seg7_out = 7'b0000000;
4'h9: seg7_out = 7'b0011000;
4'hA: seg7_out = 7'b0001000;
4'hB: seg7_out = 7'b0000011;
4'hC: seg7_out = 7'b1000110;
4'hD: seg7_out = 7'b0100001;
4'hE: seg7_out = 7'b0000110;
4'hF: seg7_out = 7'b0001110;
default: seg7_out = 7'b1111111;
    endcase
    end

```

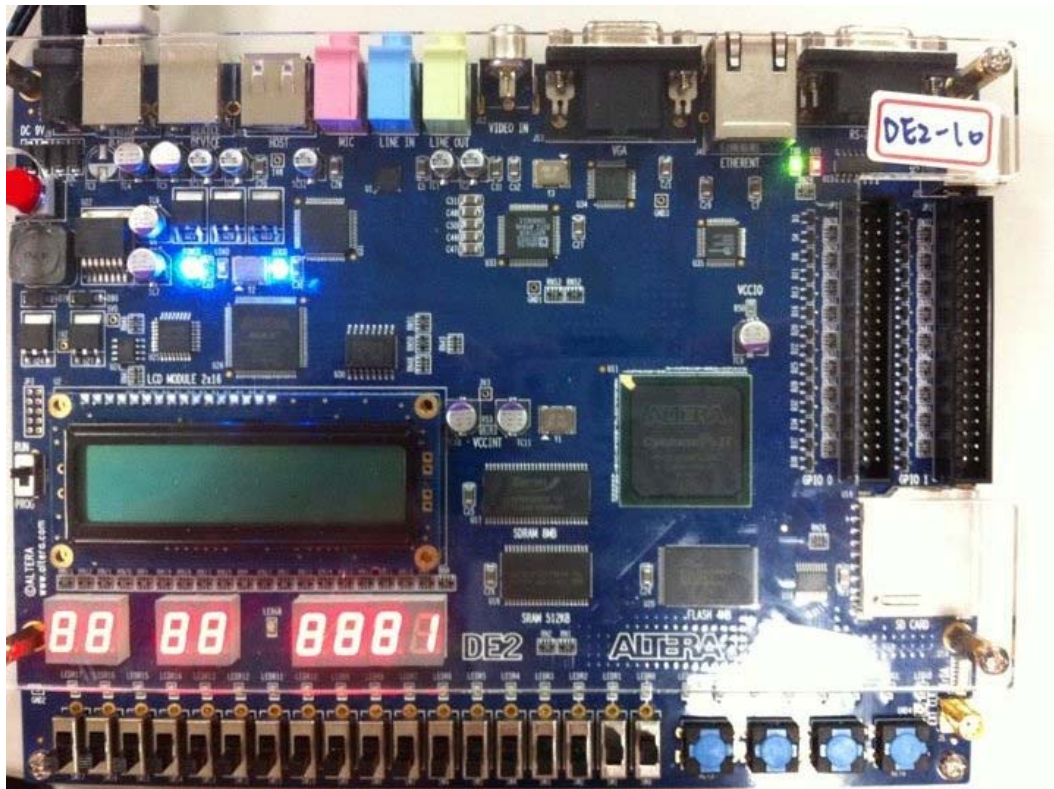
endmodule

模擬結果：



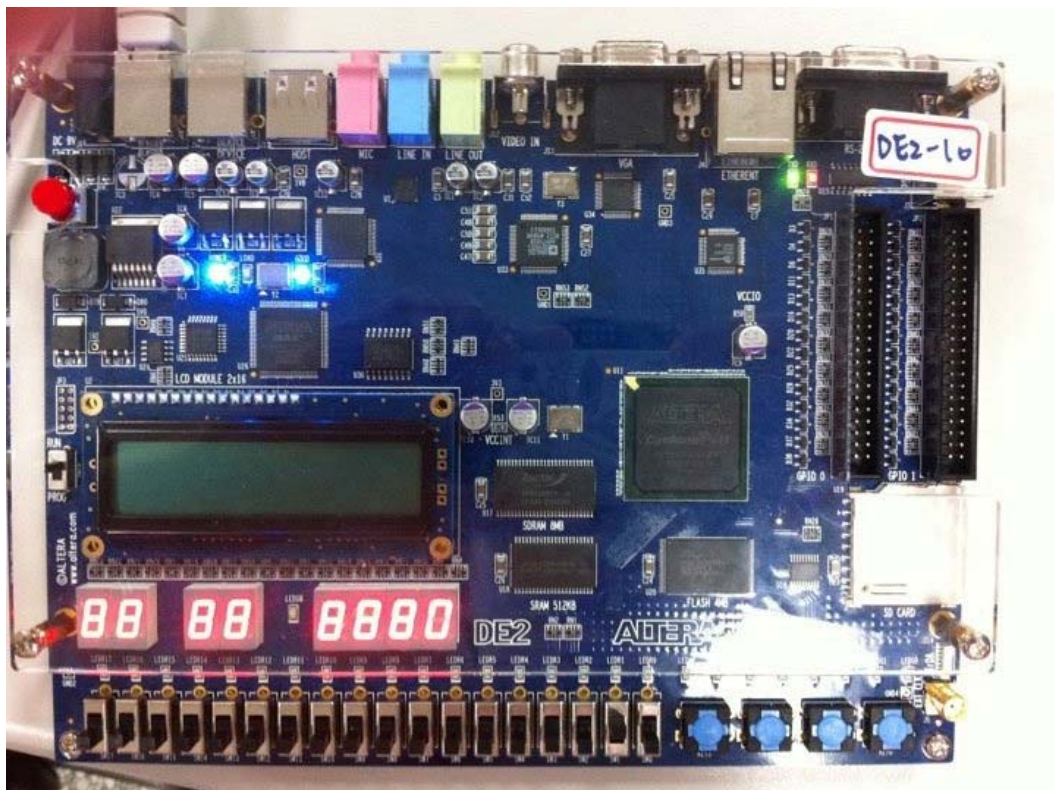
圖一

由圖一可看出，此時雖然D(SW0)=1，但因為control(SW1)=0,所以Q=0。



圖二

由圖二可看出，此時 $D(SW0)=1$ ，且因為 $control(SW1)=1$ ，所以 $Q=D=1$ 。



圖三

把 $D(SW0)$ 由圖二為1切換為圖三為0，且因為 $control(SW1)=1$ ，所以 $Q=D=0$ 。

(2) Master-slave D Flip-flop :

//to de2

```
module D_FF(seg7_out,D,CLK);
```

```
input D;
```

```
input CLK;
```

```
output [6:0] seg7_out;
```

```
wire eFF_out;
```

```
eFF test1(eFF_out,D,CLK);
```

```
binary_to_7seg test2(eFF_out,seg7_out);
```

```
endmodule
```

```
module eFF(Q,D,CLK);
```

```
input D;
```

```
input CLK;
```

```
output Q;
```

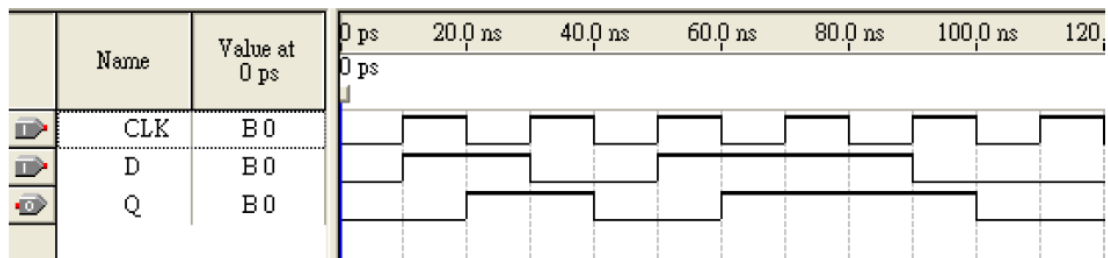
```
reg Q;
```

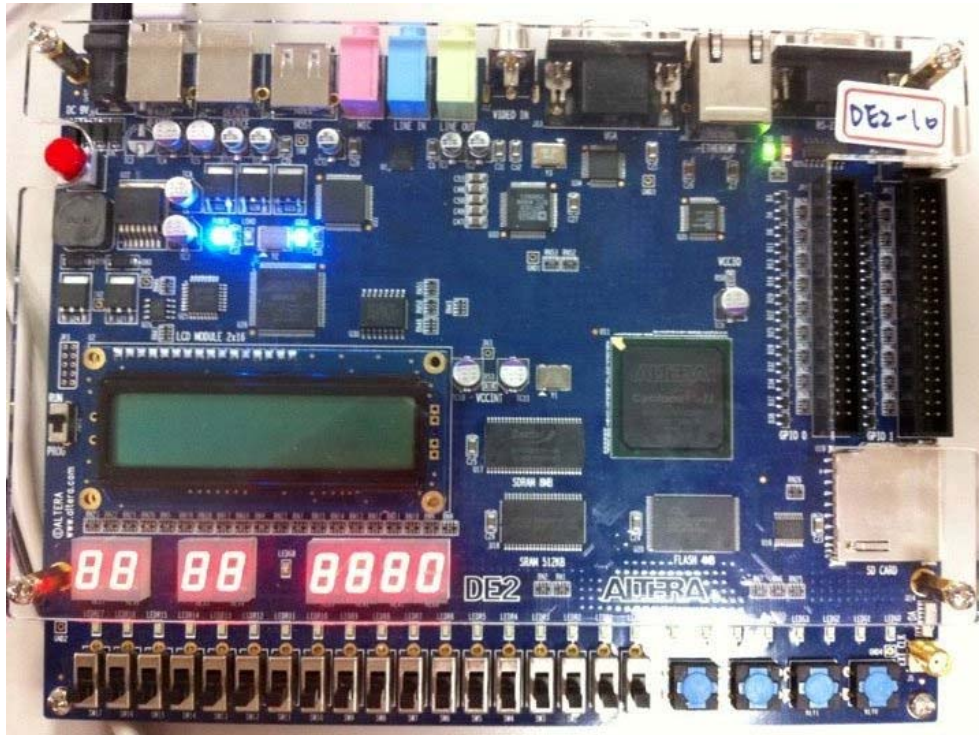
```
always @(negedge CLK)
```

```
Q = D;
```

```
endmodule
```

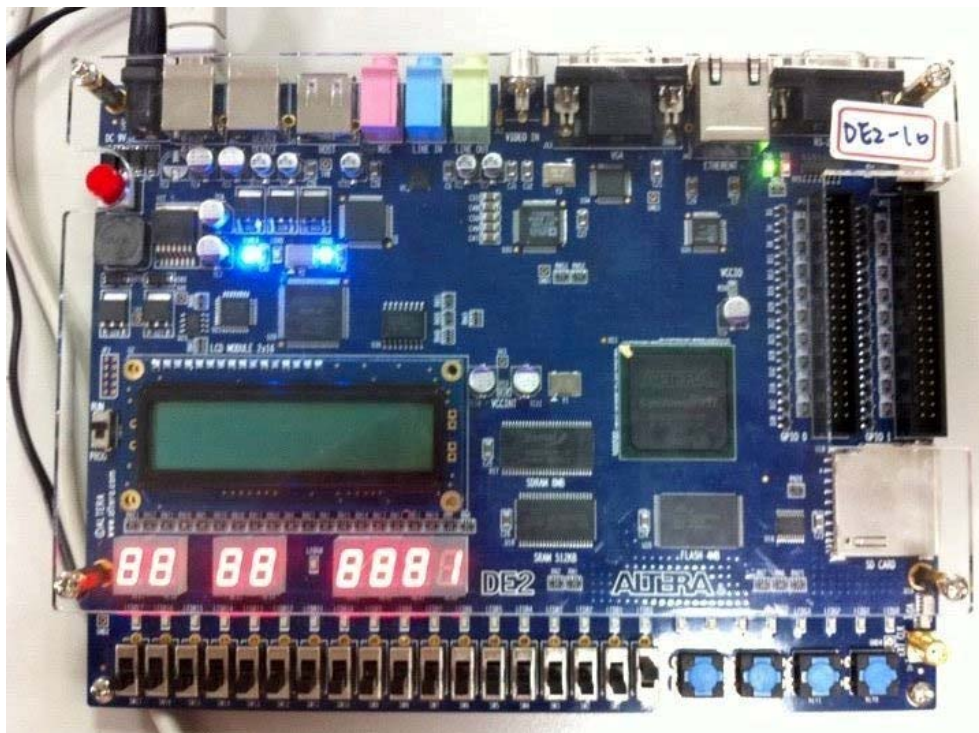
模擬結果：





圖四

由圖四可看出因為 $D(SW0)=0$ ，當 $CLK(SW1)=1 \rightarrow 0$ 時，七段顯示器最右邊(HEX0)的值=0。



圖五

由圖五可看出因為 $D(SW0)=1$ ，當 $CLK(SW1)=1 \rightarrow 0$ 時，七段顯示器最右邊(HEX0)的值=1。

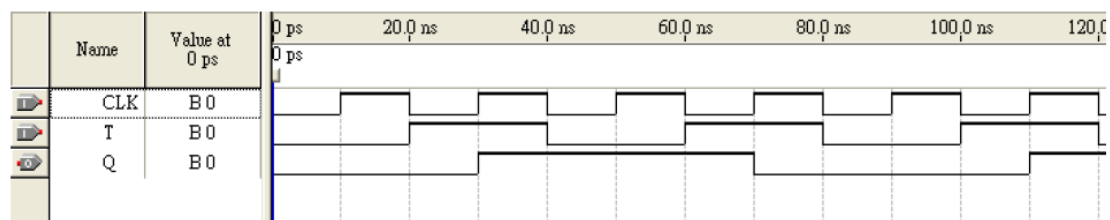
(3) T Flip-flop :

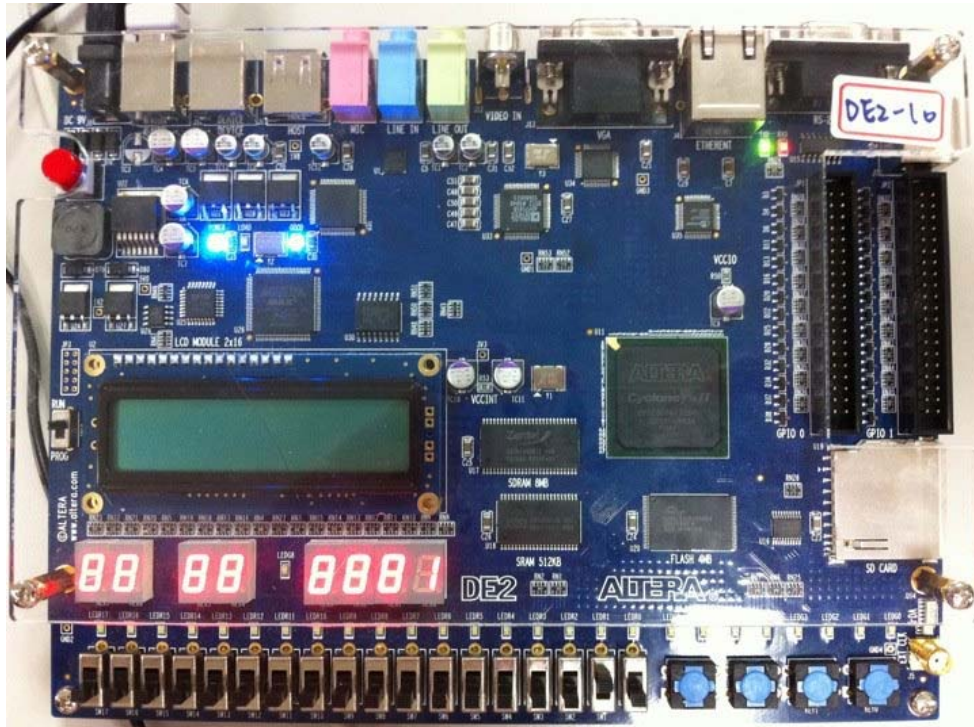
```
//to de2
module T_FF(seg7_out,T,CLK);
input  T;
input  CLK;
output [6:0] seg7_out;
wire    ttFF_out;
ttFF    test1(ttFF_out,T,CLK);
binary_to_7seg    test2(ttFF_out,seg7_out);
endmodule
```

```
module ttFF(Q,T,CLK);
input T, CLK;
output Q;
wire DT;
assign DT = Q ^ T;
D_FF D1(DT, CLK, Q);
endmodule

// D Flip-flop
module D_FF (D, CLK, Q);
input D, CLK;
output Q;
reg Q;
always @(posedge CLK)
Q = D;
endmodule
```

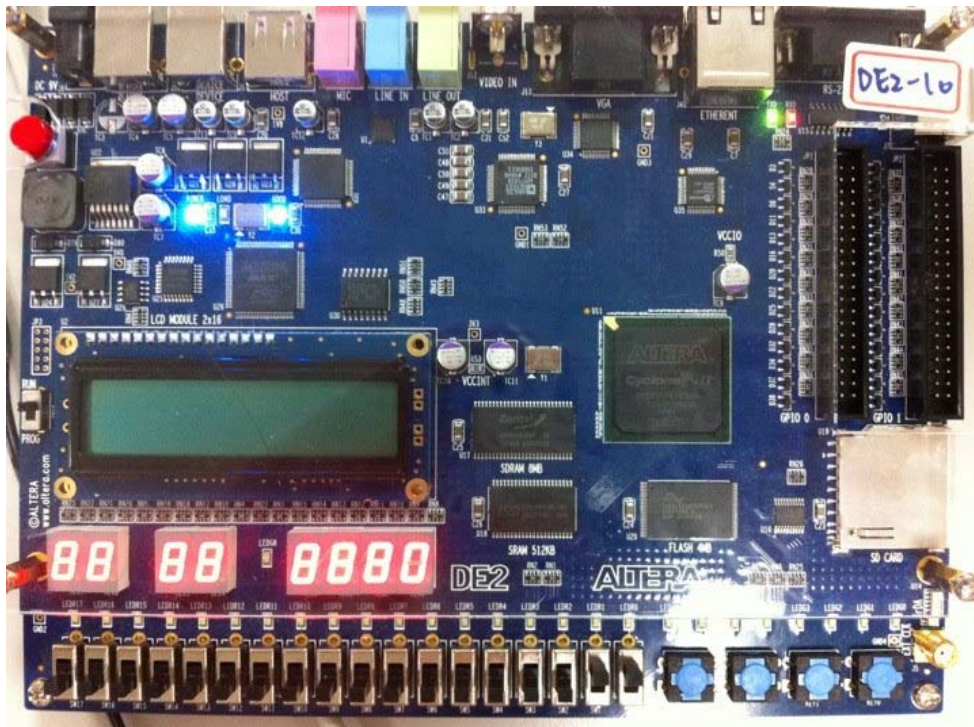
模擬結果：





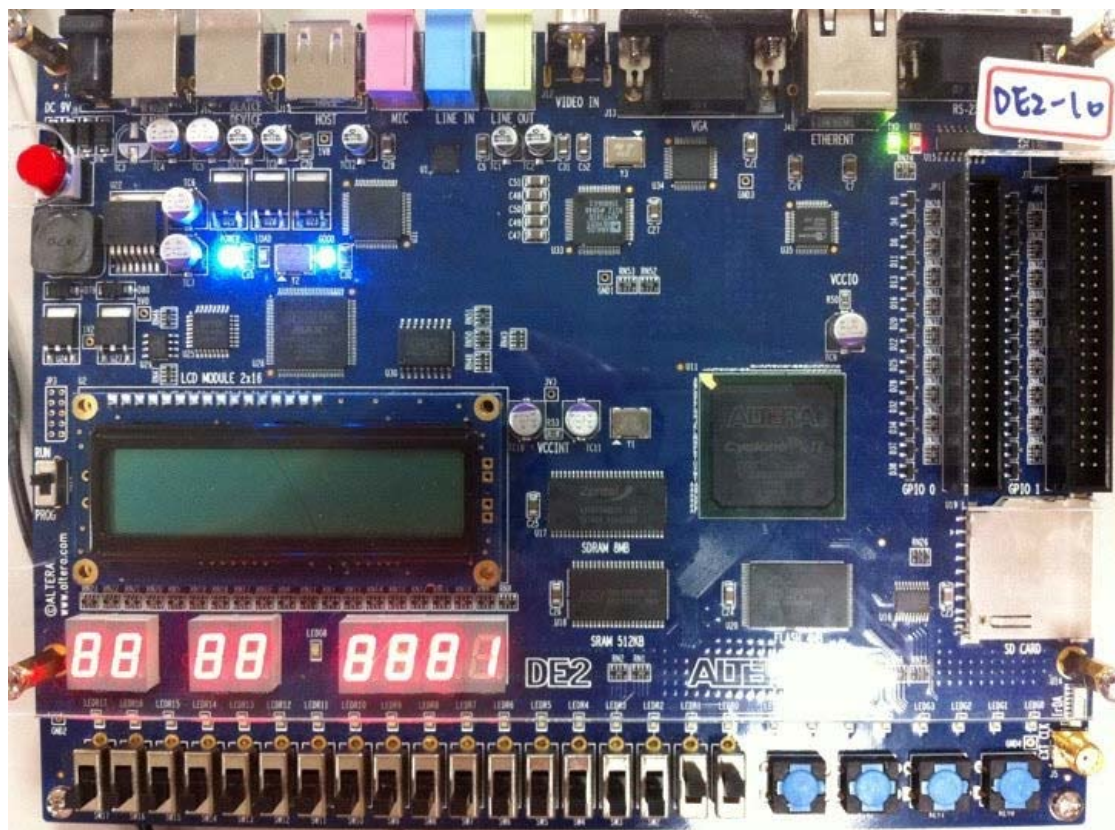
圖六

由圖六可看出，首先七段顯示器最右邊(HEX0)=1時，因為這個時候 $T(SW0)=0$ ，所以當 $CLK(SW1)=0 \rightarrow 1$ 時，HEX0依然=1。



圖七

由圖七可看出，當 $T(SW0)$ 轉變為1時， $CLK(SW1)=0 \rightarrow 1$ 時，七段顯示器最右邊(HEX0)值會從圖六的1變為圖七的0。



圖八

由圖八可看出，此時T(SW0)依然是1時，再把一次CLK(SW1)由0切換為1，七段顯示器最右邊(HEX0)值會從圖七的0變為圖八的1。

6. 問題與討論

- (1) 使用Verilog 撰寫下列電路，並給於適當的輸入驗證其輸出。
 - (a) 一個 4-bit之 D latch，其有共同之control線。
 - (b) 一個負邊緣觸發 4-bit之 D Flip-flop，且此電路有enable控制，當enable=1 且clock觸發時，正反器輸出(Q)會受輸入(D)影響，當enable=0 則不受影響。
 - (c) 一個正邊緣觸發之JK Flip-flop 。