

# Lab7 結報

---

E24066226 魏晉成

E24065018 章子嚴

E24066129 張 軒

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

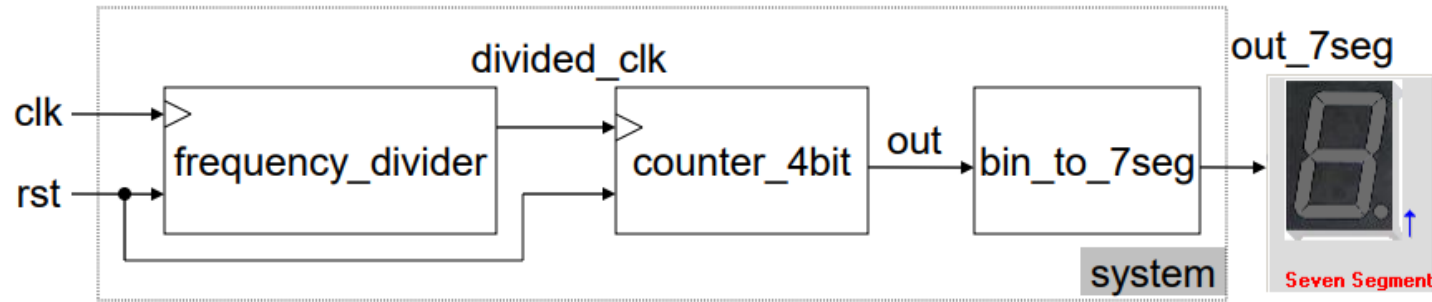
---

此實驗如下頁的(圖一)，先利用除頻器將頻率調至**1Hz**，再接至計數器模組，並利用之後的二進制轉七段顯示器模組顯示於七段顯示器上。

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

---



(圖一) 系統構想圖

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

```
1 module freqDevider(clk, rst, dclk);
2     input clk, rst;
3
4     output reg dclk;
5
6     reg[24:0] count;
7
8     always@(posedge clk, posedge rst)begin
9         if(rst)begin
10             dclk<=1'b0;
11             count<=25'd0;
12         end
13
14         else if(count==25'd24000000)begin
15             dclk<=~dclk;
16             count<=25'd0;
17         end
18
19         else count<=count+25'd1;
20     end
21 end
22
23 endmodule
```

(圖二) 除頻器模組

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

---

```
1  module counter(clk, rst, out);  
2      input clk, rst;  
3  
4      output reg [3:0]out;  
5  
6      always@(posedge clk, posedge rst)begin  
7          if(rst) out<=4'd0;  
8          else out<=out+4'd1;  
9      end  
10 endmodule
```

(圖三)計數器模組

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

```
1  module sevenSeg(in, out);
2      input [3:0]in;
3
4      output[6:0]out;
5      reg[6:0]out;
6
7      always@(*)begin
8          case(in)
9              4'b0000:out=7'b0111111;
10             4'b0001:out=7'b0000110;
11             4'b0010:out=7'b1011011;
12             4'b0011:out=7'b1001111;
13             4'b0100:out=7'b1100110;
14             4'b0101:out=7'b1101101;
15             4'b0110:out=7'b1111101;
16             4'b0111:out=7'b0000111;
17             4'b1000:out=7'b1111111;
18             4'b1001:out=7'b1101111;
19             4'b1010:out=7'b1110111;
20             4'b1011:out=7'b1111100;
21             4'b1100:out=7'b0111001;
22             4'b1101:out=7'b1011110;
23             4'b1110:out=7'b1111001;
24             4'b1111:out=7'b1110001;
25             default:out=7'b0000000;
26         endcase
27     end
28 endmodule
```

(圖四) 二進制轉  
七段顯示器模組

# 基礎題(一)

## 4bit計數器以七段顯示器顯示

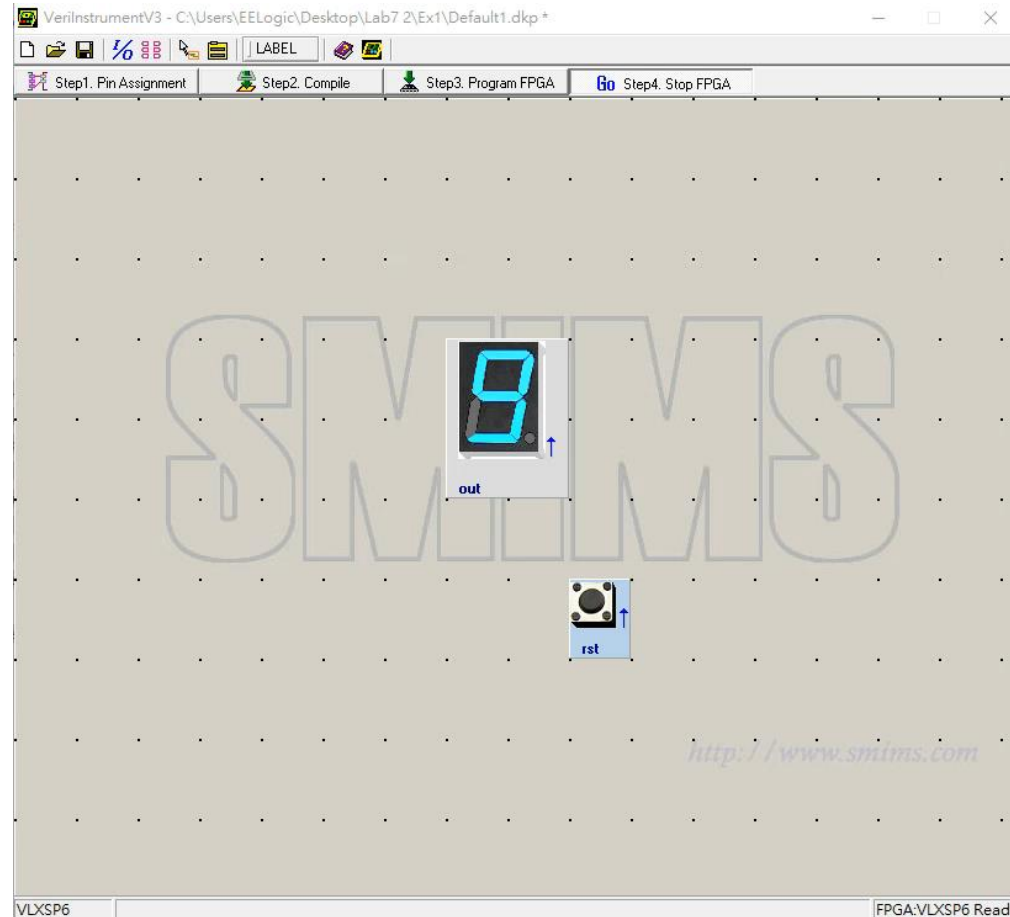
---

```
1 module system(clk, rst, out);  
2     input clk, rst;  
3  
4     output [6:0]out;  
5  
6     wire dclk;  
7     wire [3:0]cntOut;  
8  
9     freqDivider FD(.clk(clk), .rst(rst), .dclk(dclk));  
10    counter CNT(.clk(dclk), .rst(rst), .out(cntOut));  
11    sevenSeg SS(.in(cntOut), .out(out));  
12  
13 endmodule
```

(圖五) 系統

# 基礎題(一)

## 4bit計數器以七段顯示器顯示



(影片一)

VeriInstrument  
驗證



# 基礎題(一)

## 4bit計數器以七段顯示器顯示

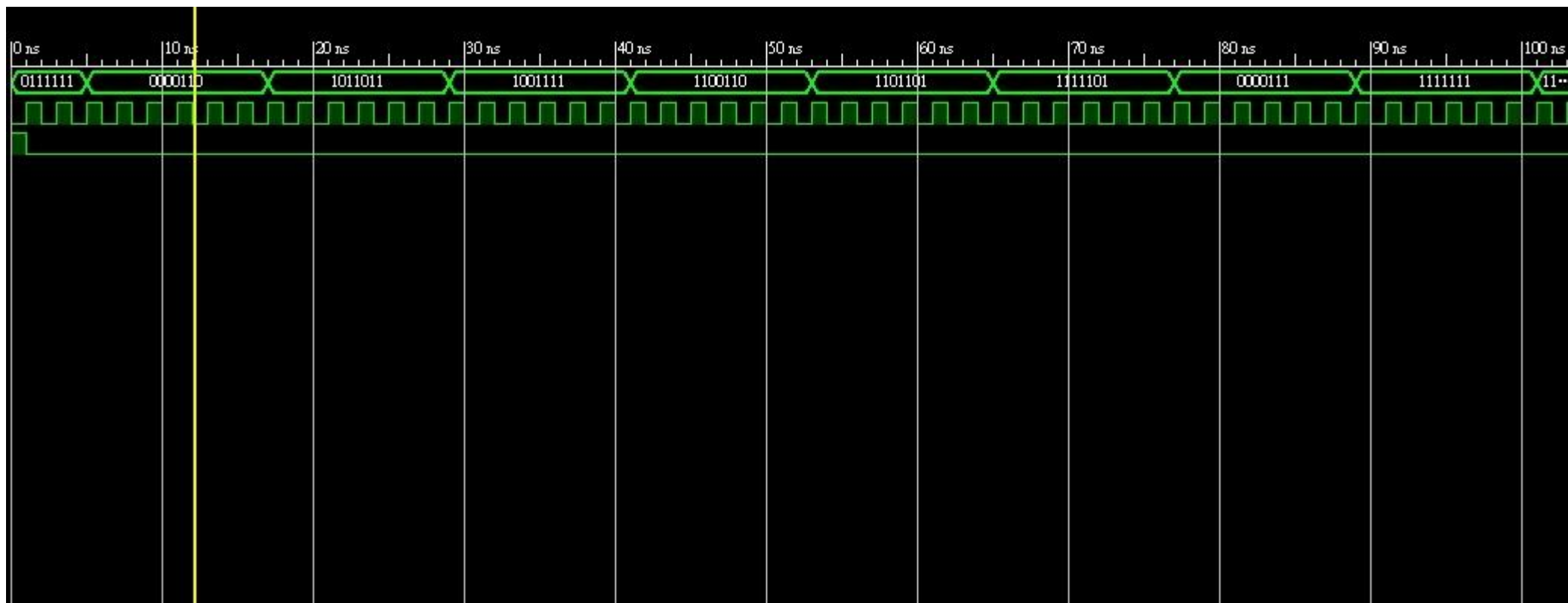
---

```
40
41     initial begin
42         // Initialize Inputs
43         clk = 0;
44         rst = 1;
45     end
46
47     initial #1 rst=0;
48
49     always #1 clk=~clk;
50
51 endmodule
52
53
```

(圖六) testbench

# 基礎題(一)

## 4bit計數器以七段顯示器顯示



(圖七) testbench 結果

# 基礎題(二)

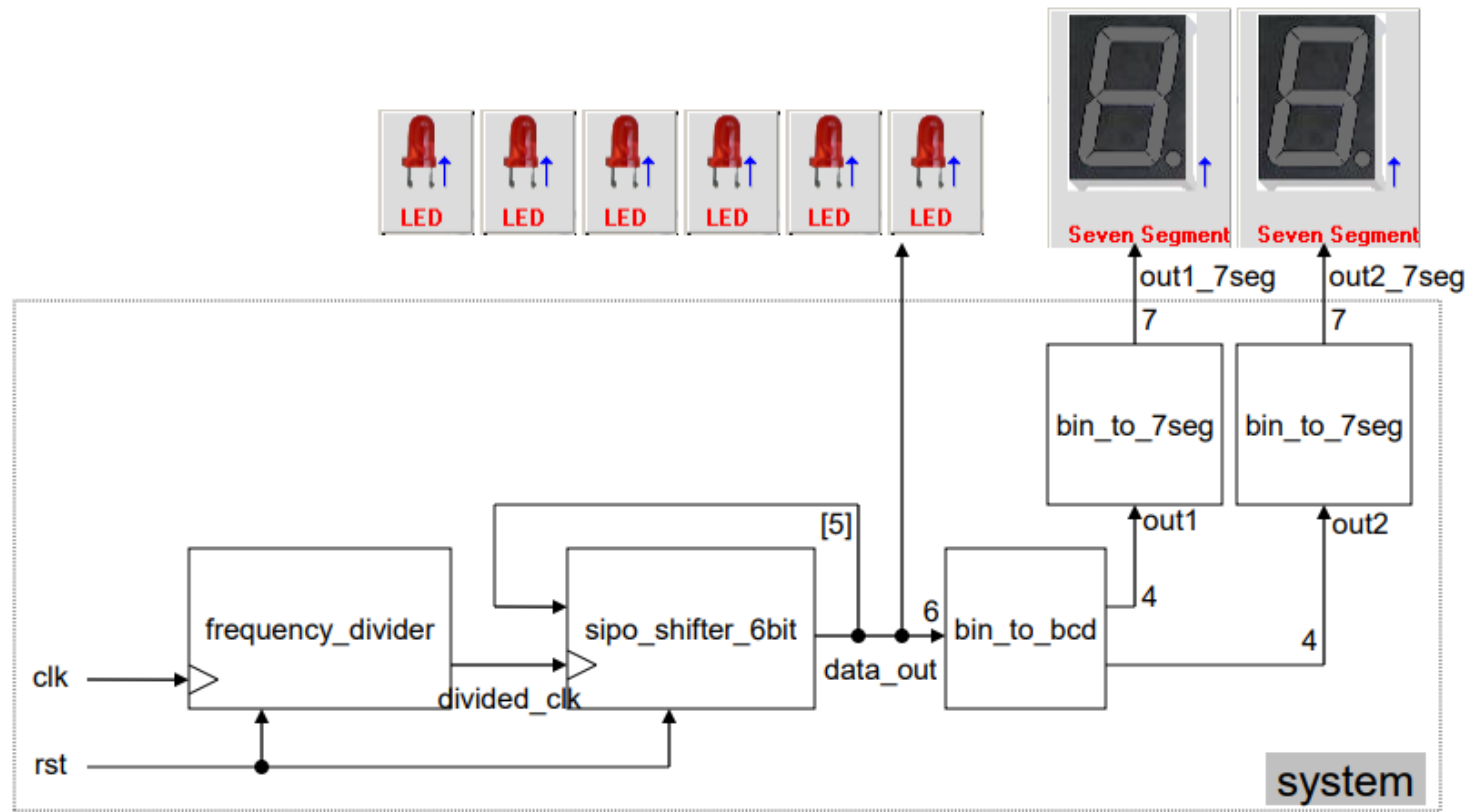
## 6bit串入並出移位器與七段顯示器

---

此實驗如下頁的(圖八)，先利用除頻器將頻率調至1Hz，再接串入並出暫存器，使發光的LED燈能隨時間由LSB往MSB移動，接著連上6位元binary轉8位元BCD的模組，使二進制碼能轉換成二位數的十進制碼，最後接到二進制轉七段顯示器模組，將輸入顯示在七段顯示器上，變大功告成。

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器



(圖八)  
系統構想圖

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器

```
1 module freqDevider(clk, rst, dclk);  
2     input clk, rst;  
3  
4     output reg dclk;  
5  
6     reg[24:0]count;  
7  
8     always@(posedge clk, posedge rst)begin  
9         if(rst)begin  
10             dclk<=1'b0;  
11             count<=25'd0;  
12         end  
13  
14         else if(count==25'd24000000)begin  
15             dclk<=~dclk;  
16             count<=25'd0;  
17         end  
18  
19         else count<=count+25'd1;  
20     end  
21 end  
22  
23 endmodule
```

(圖九)除頻器模組

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器

```
1  module sipoShifter(clk, rst, in, out);
2      input in, clk, rst;
3
4      output reg [5:0]out;
5
6      always@(posedge rst or posedge clk)begin
7          if(rst) out<=6'b000001;
8          else begin
9              out<=out<<1;
10             out[0]<=in;
11         end
12     end
13
14 endmodule
```

(圖十)  
SIP0 暫存器

## 基礎題(二)

### 6bit串入並出移位器與七段顯示器

```
1  module binToBCD(in, outOne, outTen);  
2      input [5:0]in;  
3      output reg [3:0]outTen, outOne;  
4  
5      always@(*)begin  
6          outTen<=in/4'd10;  
7          outOne<=in%4'd10;  
8      end  
9  
10 endmodule
```

(圖十一) 二進位二進制轉二進位十進制模組

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器

```
1  module sevenSeg(in, out);
2      input [3:0]in;
3
4      output[6:0]out;
5      reg[6:0]out;
6
7      always@(*)begin
8          case(in)
9              4'b0000:out=7'b0111111;
10             4'b0001:out=7'b0000110;
11             4'b0010:out=7'b1011011;
12             4'b0011:out=7'b1001111;
13             4'b0100:out=7'b1100110;
14             4'b0101:out=7'b1101101;
15             4'b0110:out=7'b1111101;
16             4'b0111:out=7'b0000111;
17             4'b1000:out=7'b1111111;
18             4'b1001:out=7'b1101111;
19             4'b1010:out=7'b1110111;
20             4'b1011:out=7'b1111100;
21             4'b1100:out=7'b0111001;
22             4'b1101:out=7'b1011110;
23             4'b1110:out=7'b1111001;
24             4'b1111:out=7'b1110001;
25             default:out=7'b0000000;
26         endcase
27     end
28 endmodule
```

(圖十二) 二進制轉  
七段顯示器模組



# 基礎題(二)

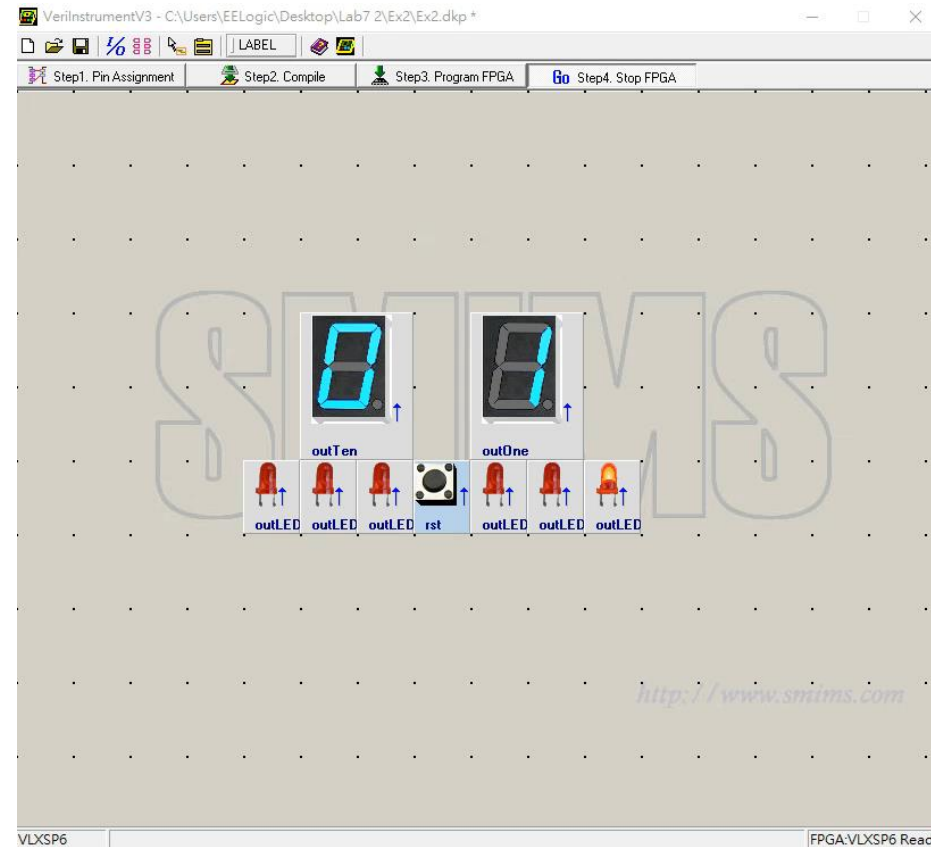
## 6bit串入並出移位器與七段顯示器

```
1  module system(clk, rst, outLED, outOne, outTen);
2      input clk, rst;
3
4      output [5:0]outLED;
5      output [6:0]outOne, outTen;
6
7      wire dclk;
8      wire [5:0]SIPOout;
9      wire [3:0]BCDoutTen, BCDoutOne;
10     wire [6:0]SSOne, SSTen;
11
12     freqDevider FD(.clk(clk), .rst(rst), .dclk(dclk));
13     sipoShifter SIPO(.clk(dclk), .rst(rst), .in(outLED[5]), .out(outLED));
14     binToBCD BTB(.in(SIPOout), .outOne(BCDoutOne), .outTen(BCDoutTen));
15     sevenSeg SS0(.in(BCDoutOne), .out(outOne));
16     sevenSeg SS1(.in(BCDoutTen), .out(outTen));
17
18     assign SIPOout=outLED;
19
20 endmodule
```

(圖十三)  
系統

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器



(影片二)

VeriInstrument  
驗證

## 基礎題(二)

# 6bit串入並出移位器與七段顯示器

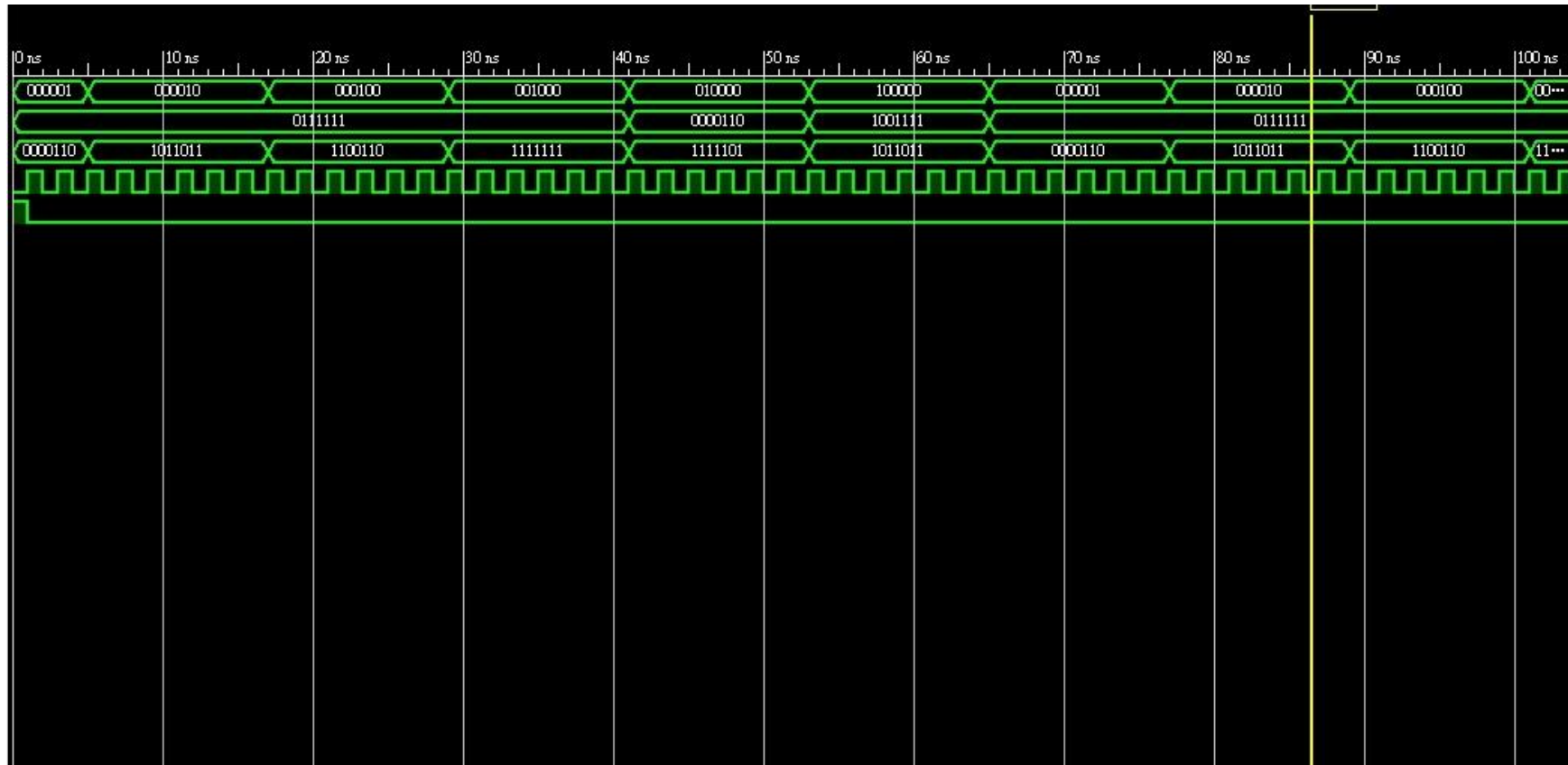
---

```
44
45     initial begin
46         // Initialize Inputs
47         clk = 0;
48         rst = 1;
49     end
50
51     initial #1 rst=0;
52
53     always #1 clk=~clk;
54
55 endmodule
56
```

(圖十四) testbench

# 基礎題(二)

## 6bit串入並出移位器與七段顯示器



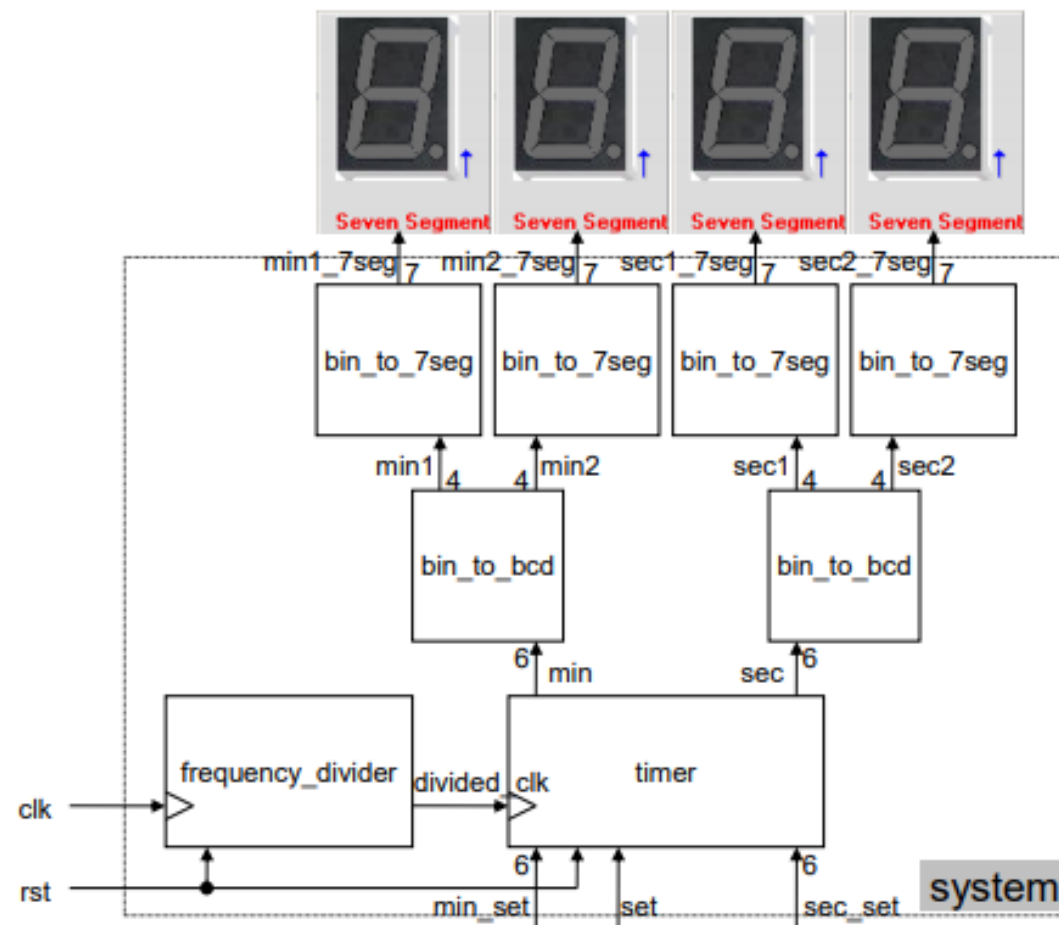
(圖十五) testbench

# 挑戰題 計時器

---

此實驗如下頁的(圖十六)，先利用除頻器將頻率調至1Hz，再接至計時器模組，並且分別將分與秒的兩組輸出接至6位元binary轉8位元BCD的模組，使二進制碼能轉換成二位數的十進制碼；最後將個別的個位數與十位數接到二進制轉七段顯示器模組，使輸入顯示在七段顯示器上，變大功告成。

# 挑戰題 計時器



(圖十六)  
系統構想圖

# 挑戰題 計時器

```
1 module freqDevider(clk, rst, dclk);
2     input clk, rst;
3
4     output reg dclk;
5
6     reg[24:0] count;
7
8     always@(posedge clk, posedge rst)begin
9         if(rst)begin
10             dclk<=1'b0;
11             count<=25'd0;
12         end
13
14         else if(count==25'd24000000)begin
15             dclk<=~dclk;
16             count<=25'd0;
17         end
18
19         else count<=count+25'd1;
20     end
21 end
22
23 endmodule
```

(圖十七) 除頻器模組

# 挑戰題 計時器

```
1 module timer(clk, rst, minSet, secSet, set, min, sec);
2     input clk, rst, set;
3     input [5:0]minSet, secSet;
4
5     output reg [5:0]min, sec;
6
7     always@(posedge clk or posedge rst or posedge set)begin
8         if(rst) begin min<=6'b0; sec<=6'b0;end
9
10        else if(set) begin
11            if(minSet<6'd60&&secSet<6'd60)begin min<=minSet;sec<=secSet;end
12            else if(minSet<6'd60)begin min<=minSet;sec<=secSet-6'd60;end
13            else if(secSet<6'd60)begin min<=(minSet-6'd60); sec<=secSet;end
14            else begin min<=minSet-6'd60; sec<=secSet-6'd60;end
15        end
16
17        else begin
18            if(sec==6'd59) begin
19                if(min!=6'd59) begin min<=min+6'd1; sec<=6'd0; end
20                else begin min<=6'b0; sec<=6'b0;end
21            end
22
23            else begin min<=min; sec<=sec+6'd1;end
24        end
25    end
26 end
27
28 endmodule
```

(圖十八)  
計時器模組



# 挑戰題 計時器

```
1  module binToBCD(in, outOne, outTen);  
2      input [5:0]in;  
3      output reg [3:0]outTen, outOne;  
4  
5      always@(*)begin  
6          outTen<=in/4'd10;  
7          outOne<=in%4'd10;  
8      end  
9  
10 endmodule
```

(圖十九) 二進位二進制轉二進位十進制模組

# 挑戰題 計時器

```
1 module sevenSeg(in, out);
2     input [3:0]in;
3
4     output[6:0]out;
5     reg[6:0]out;
6
7     always@(*)begin
8         case(in)
9             4'b0000:out=7'b0111111;
10            4'b0001:out=7'b0000110;
11            4'b0010:out=7'b1011011;
12            4'b0011:out=7'b1001111;
13            4'b0100:out=7'b1100110;
14            4'b0101:out=7'b1101101;
15            4'b0110:out=7'b1111101;
16            4'b0111:out=7'b0000111;
17            4'b1000:out=7'b1111111;
18            4'b1001:out=7'b1101111;
19            4'b1010:out=7'b1110111;
20            4'b1011:out=7'b1111100;
21            4'b1100:out=7'b0111001;
22            4'b1101:out=7'b1011110;
23            4'b1110:out=7'b1111001;
24            4'b1111:out=7'b1110001;
25            default:out=7'b0000000;
26        endcase
27    end
28 endmodule
```

(圖二十) 二進制轉  
七段顯示器模組

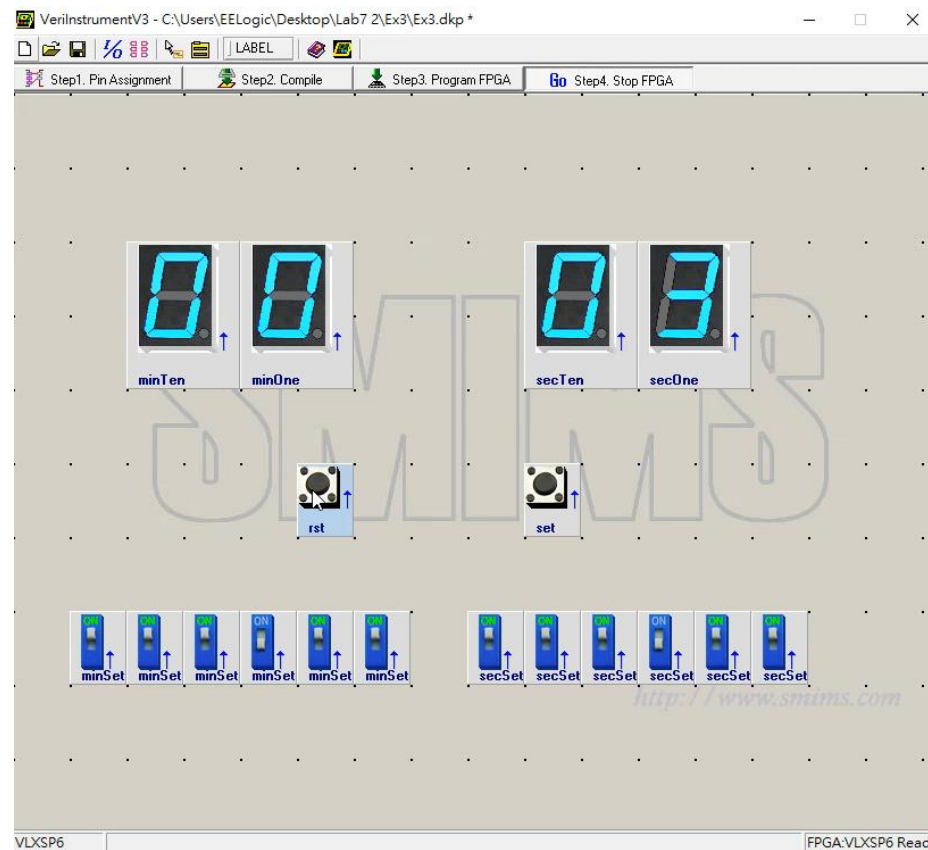
# 挑戰題 計時器

---

```
1 module system(clk, rst, set, minSet, secSet, minTen, minOne, secTen, secOne);
2     input clk, rst, set;
3     input[5:0] minSet, secSet;
4
5     output[6:0] minTen, minOne, secTen, secOne;
6
7     wire dclk;
8     wire[5:0] min, sec;
9     wire[3:0] minTenBin, minOneBin, secTenBin, secOneBin;
10
11     freqDevider FD(.clk(clk), .rst(rst), .dclk(dclk));
12
13     timer T0(.clk(dclk), .rst(rst), .minSet(minSet), .secSet(secSet), .set(set), .min(min), .sec(sec));
14     binToBCD BTBMin(.in(min), .outOne(minOneBin), .outTen(minTenBin));
15     binToBCD BTBSec(.in(sec), .outOne(secOneBin), .outTen(secTenBin));
16     sevenSeg minTenSS(.in(minTenBin), .out(minTen)), minOneSS(.in(minOneBin), .out(minOne)), secTenSS(.in(secTenBin), .out(secTen)), secOneSS(.in(secOneBin),
17     out(secOne));
18 endmodule
```

(圖二十一) 系統

# 挑戰題 計時器



(影片三)

VeriInstrument  
驗證

# 挑戰題 計時器

---

```
52
53     initial begin
54         clk=0;
55         rst=1;
56         set=0;
57         min_set=6'd22;
58         sec_set=6'd22;
59     end
60
61     initial begin
62         #1 rst=0;
63         #1 set=0;
64     end
65
66     always #1 clk=~clk;
67
68     initial begin
69         #800 set=1;
70         #100 rst=1;
71     end
72
73 endmodule
74
75
```

(圖二十二)  
testbench

# 挑戰題 計時器



## (圖二十二) testbench結果

# 心得

---

張軒：

做完這次實驗後，對c1k有更深入的了解，因為上一次的實驗沒有真的對c1k做什麼事情，所以也不知道他到底可以幹嘛。然後模組真是個好用的東西，很多東西都只需要寫一次，需要用到的時候再把很多模組組合成一個系統就好，不知道final project是不是也可以用很多模組組裝起來就好哈哈。

# 心得

---

章子嚴：這次的實驗還可以，看著他們寫verilog的時候，開始大概知道verilog這麼寫了，都是一部分寫完然後結合起來。用模組檢查好過用testbench檢查，模組操作起來也方便，檢查時也很方便，看verilog有沒有地方寫錯。



# 心得

---

魏晉成：

雖然上次黑特了VeriInstrutment一波，但因為這次要用到clk、七段顯示器以及指撥開關等物件，這些是在testbench中寫起來很痛苦的輸入輸出，所以這次終於體會到為何大家會一直誇讚VeriInstrument的好。