邏輯設計實驗Lab4結報

105060012張育菘

**1. Implement pre-lab1**:

**Cascade eight DFFs together as a shift register. Connect the output of the last DFF to the input of the first DFF as a ringer counter. Let the initial value of DFF output after reset be 01010101. Construct the Verilog RTL representation for the logics with verification.**

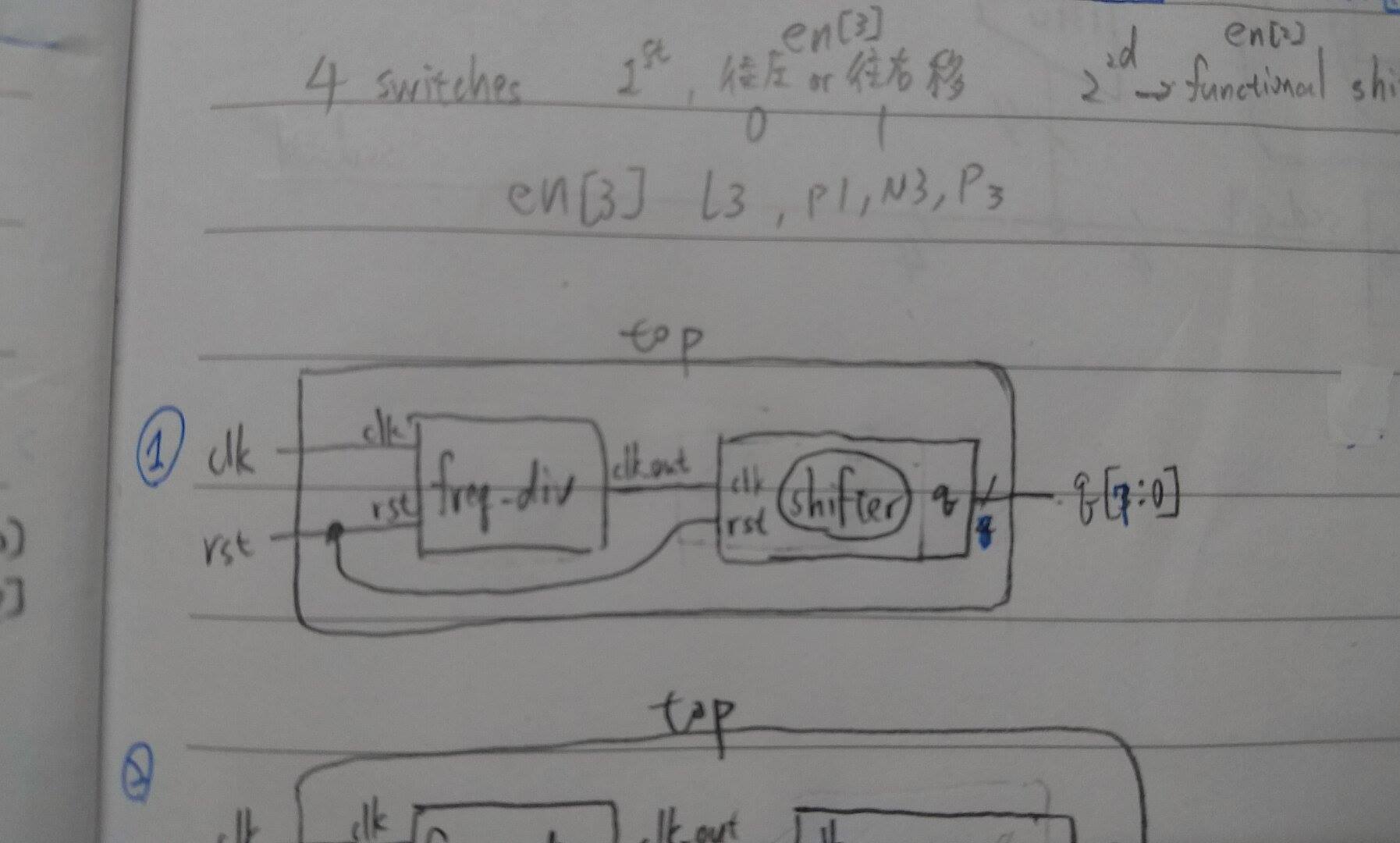
|  |  |
| --- | --- |
| **clk** | **rst** |
| **W5** | **V17** |

Design Specification

input clk,rst;

output [7:0]q;

block diagram :



Design Implementation

Logic function :

1. top：

此top module，專門拿來呼叫其他小moudule的。

並接一內線clk\_out作為shifter的clk。

1. freq\_div：

此為除頻的module，用來控制shifter的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=27'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的227倍。

1. shifter：

此作用就是做移位的工作，使所有的bit往左移一位，最高位則變成最低位。

此module就是同時有8的DFFs在作用，clk每從0🡪1時，即移位。

always@(posedge clk or negedge rst)

if(~rst) q<=8'b01010101; //設初始值為01010101

else

begin

q[0]<=q[7];

q[1]<=q[0];

q[2]<=q[1];

q[3]<=q[2];

q[4]<=q[3];

q[5]<=q[4];

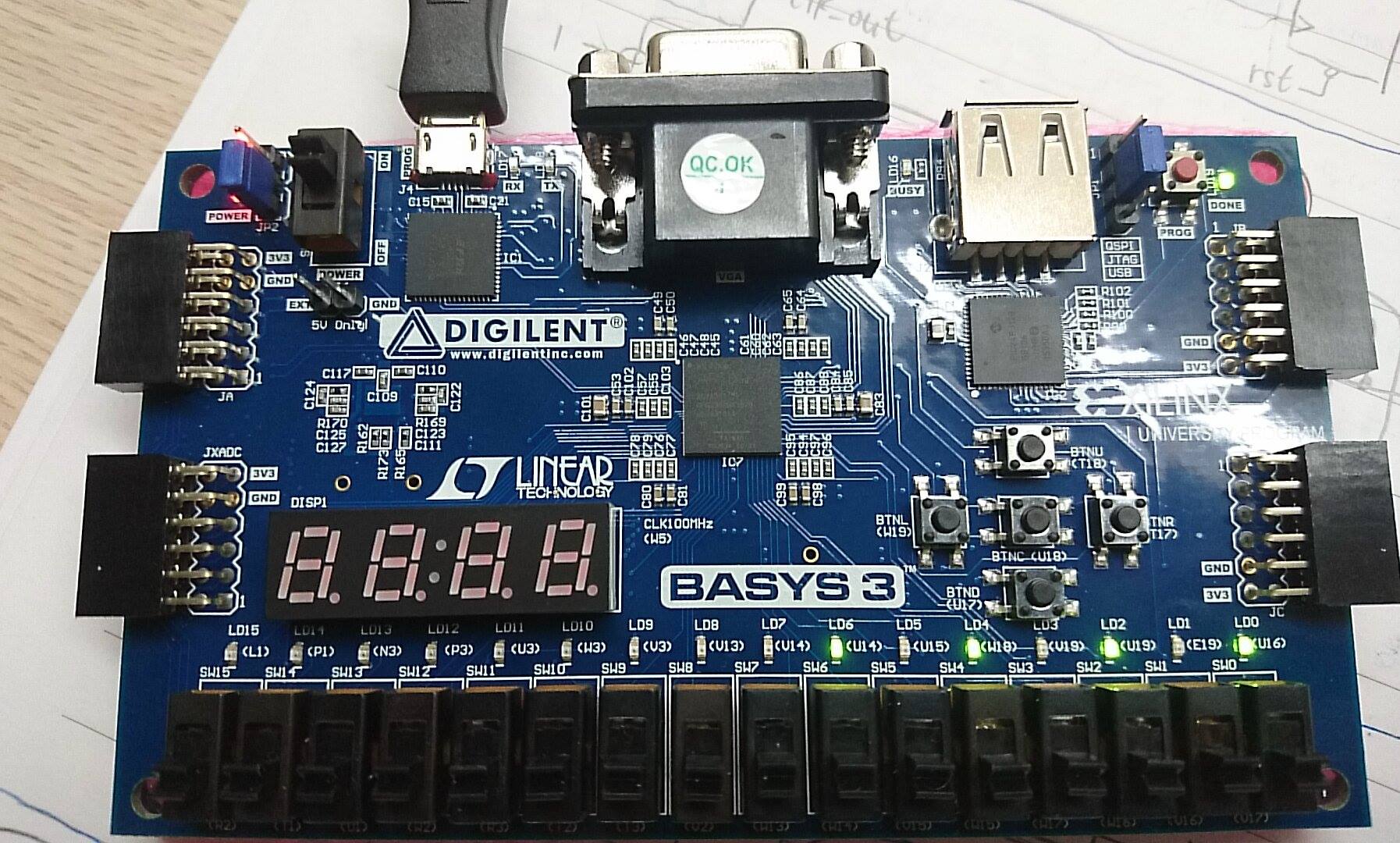
q[6]<=q[5];

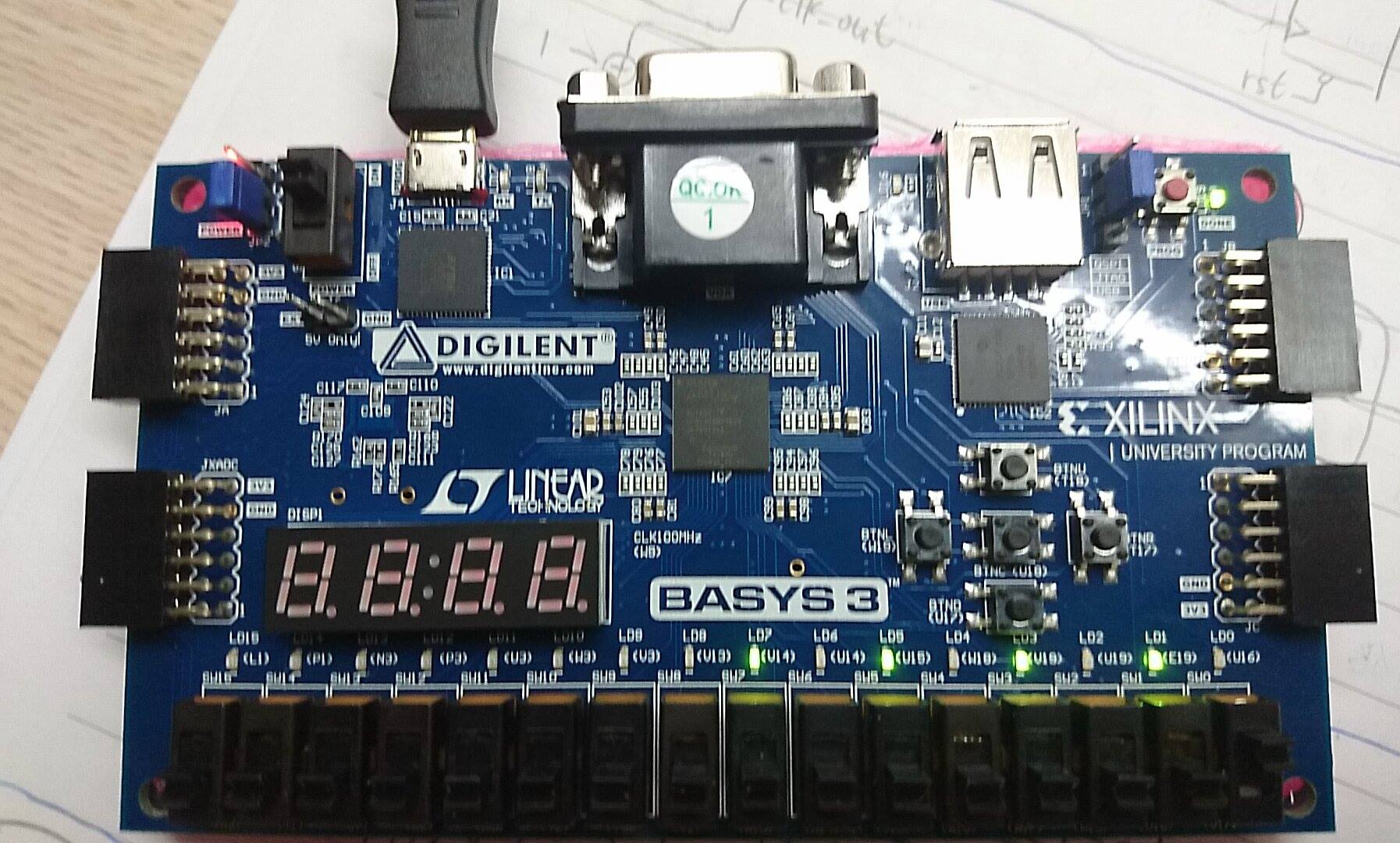
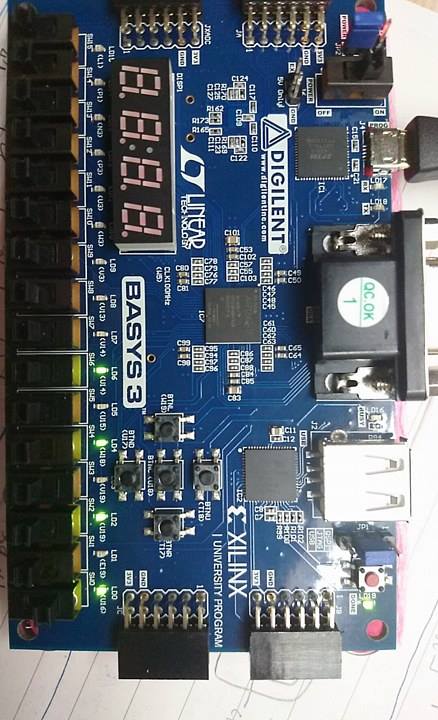
q[7]<=q[6];

end

Result

1. rst=0🡪初始化設為01010101
2. rst=1🡪LED燈來回在01010101與10101010之間轉換顯示。



Discussion

1. 這次實驗教會我如何操作shifter。

**2.** **Construct a ring counter similar to that of pre-lab1 but the initial value of the DFFs can be set randomly.**

|  |  |
| --- | --- |
| **clk** | **rst** |
| **W5** | **V17** |

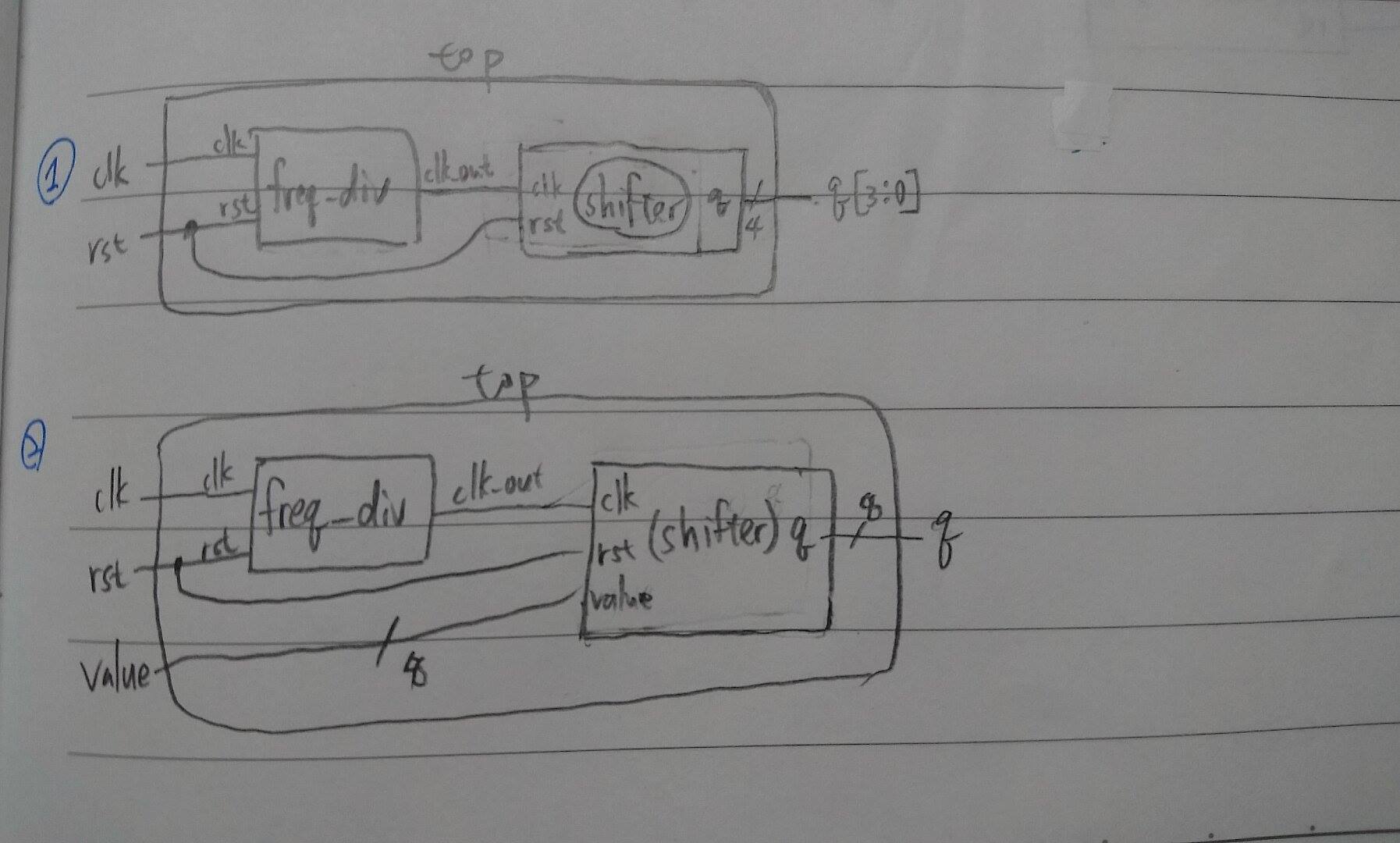
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **value[7]** | **value[6]** | **value[5]** | **value[4]** | **value[3]** | **value[2]** | **value[1]** | **value[0]** |
| **R2** | **T1** | **U1** | **W2** | **R3** | **T2** | **T3** | **V2** |

Design Specification

input : clk,rst;

output : [7:0]value, [7:0]q;

block diagram :



Design Implementation

Logic function :

1. top：

此top module，專門拿來呼叫其他小moudule的。

並接一內線clk\_out作為shifter的clk。

1. freq\_div：

此為除頻的module，用來控制shifter的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=27'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的227倍。

1. shifter：

此作用就是做移位的工作，使所有的bit往左移一位，最高位則變成最低位。

此module就是同時有8的DFFs在作用，clk每從0🡪1時，即移位。

always@(posedge clk or negedge rst)

if(~rst) q<=value; //初始值為自己輸入的value值

else

begin

q[0]<=q[7];

q[1]<=q[0];

q[2]<=q[1];

q[3]<=q[2];

q[4]<=q[3];

q[5]<=q[4];

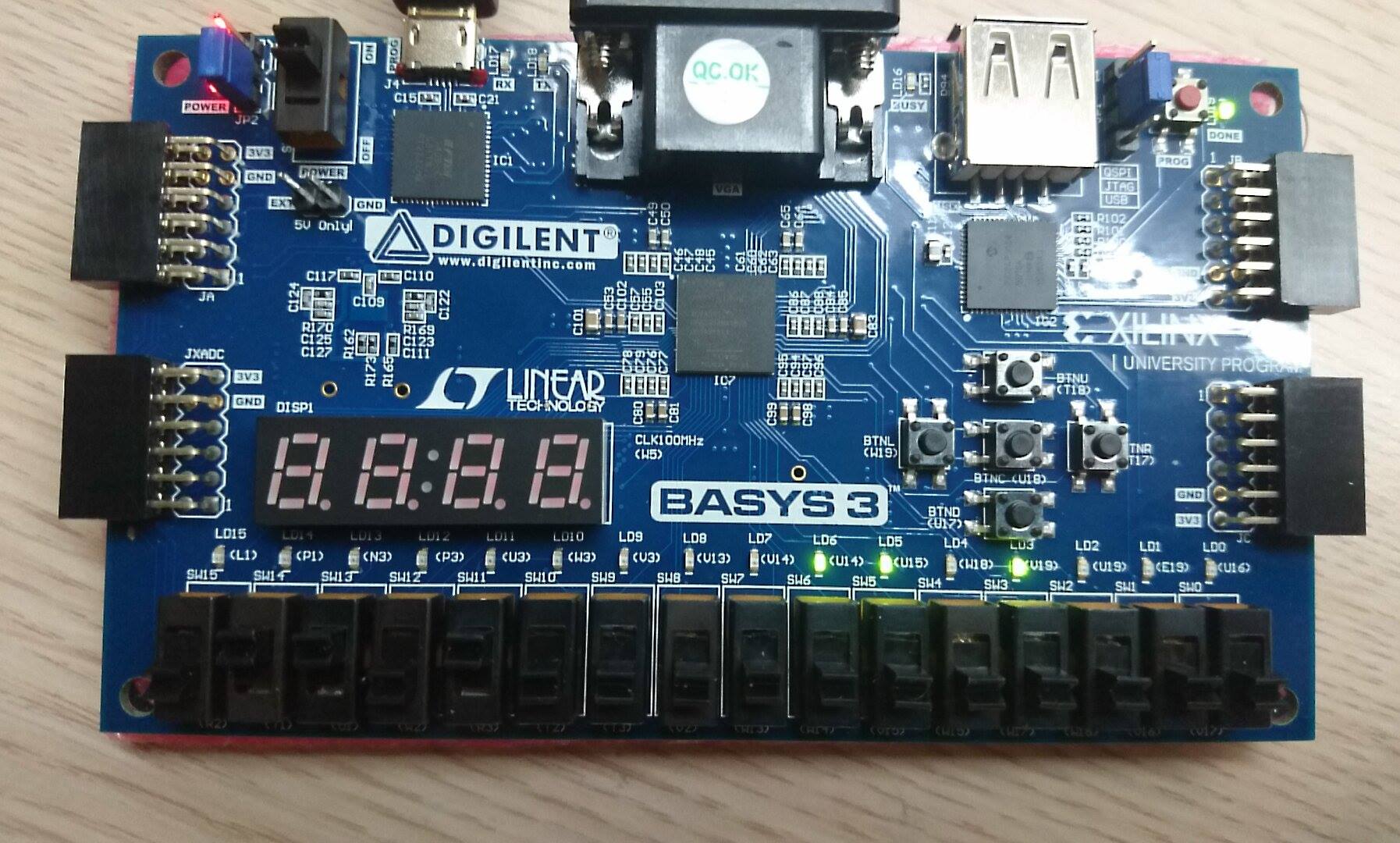
q[6]<=q[5];

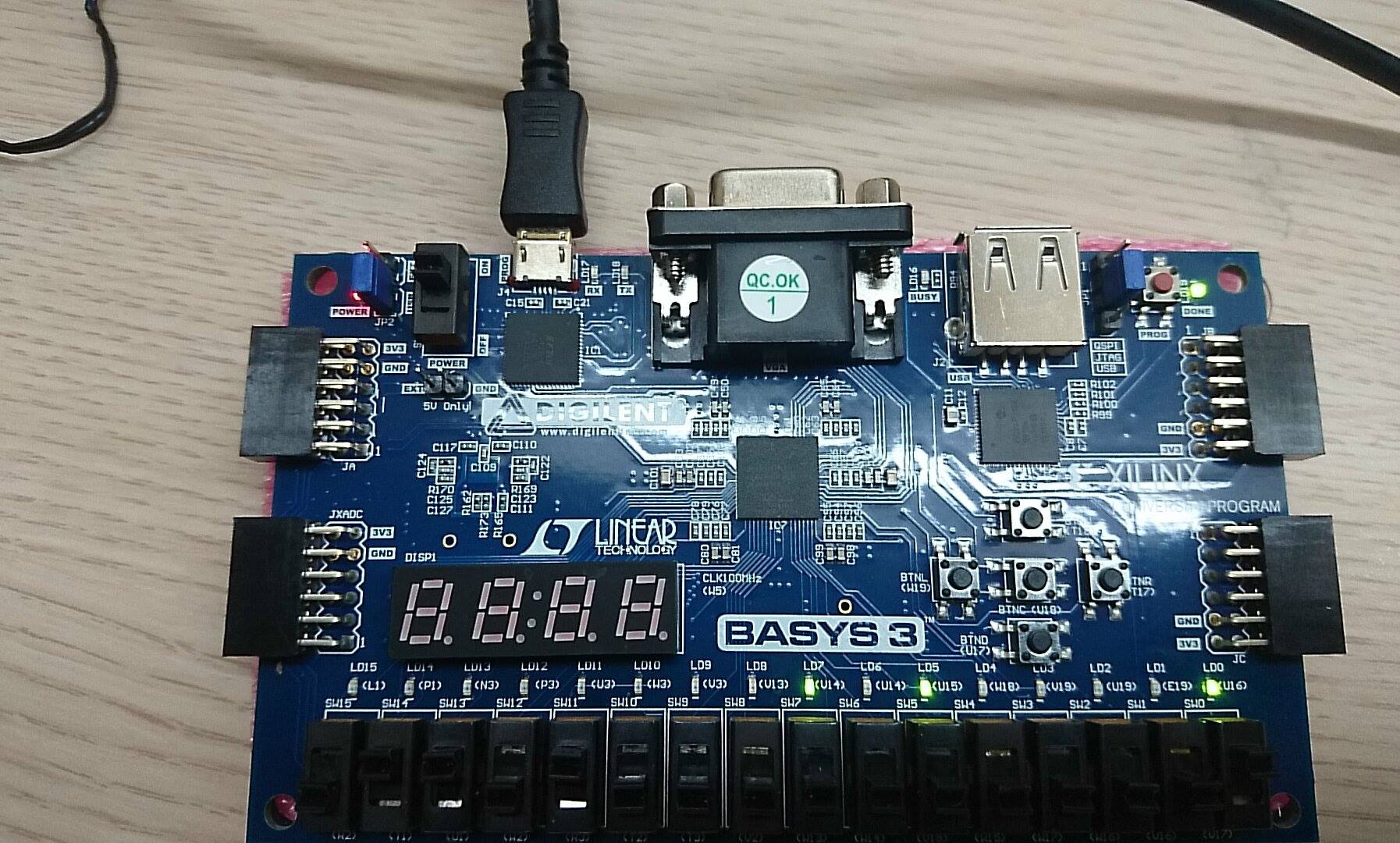
q[7]<=q[6];

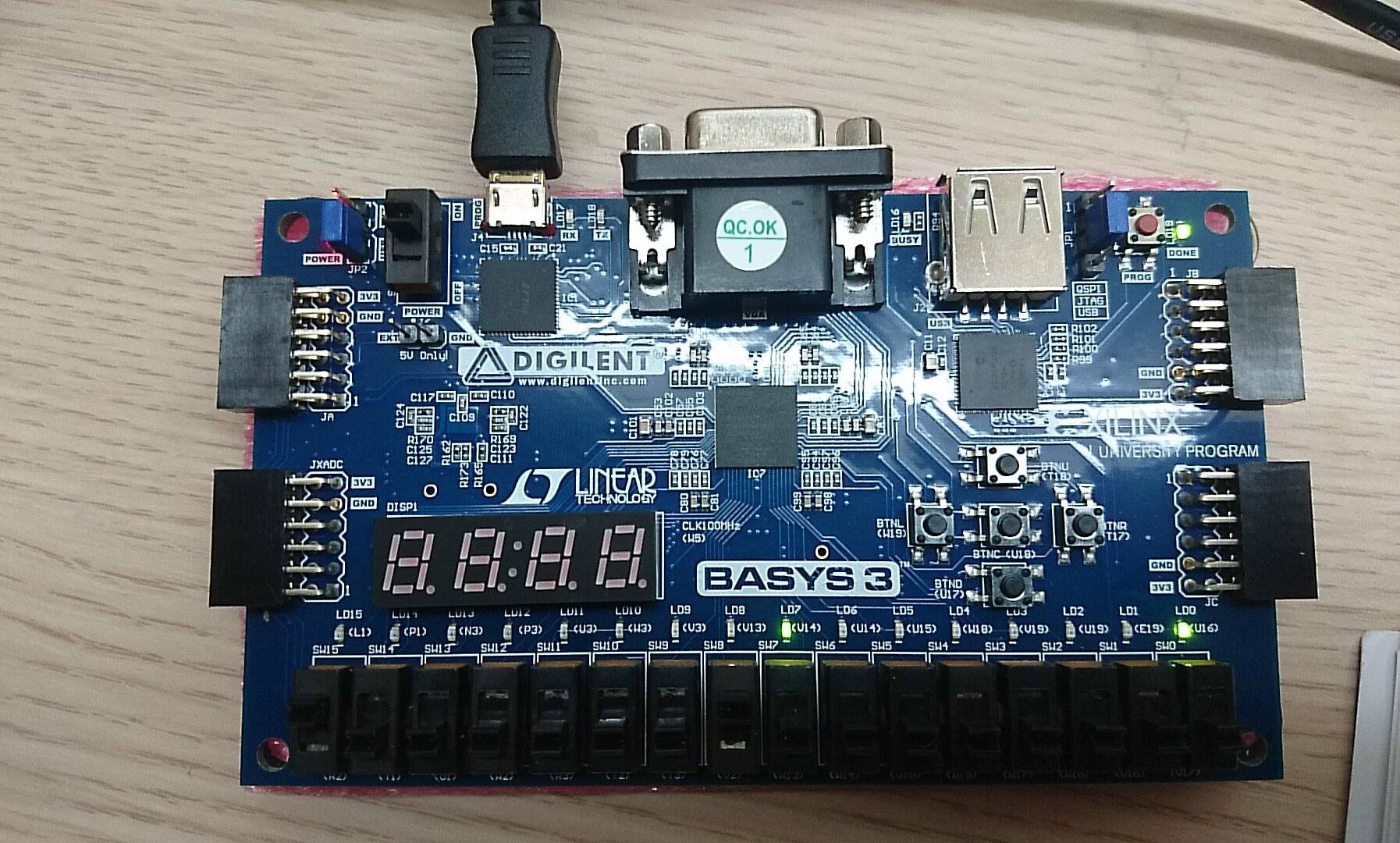
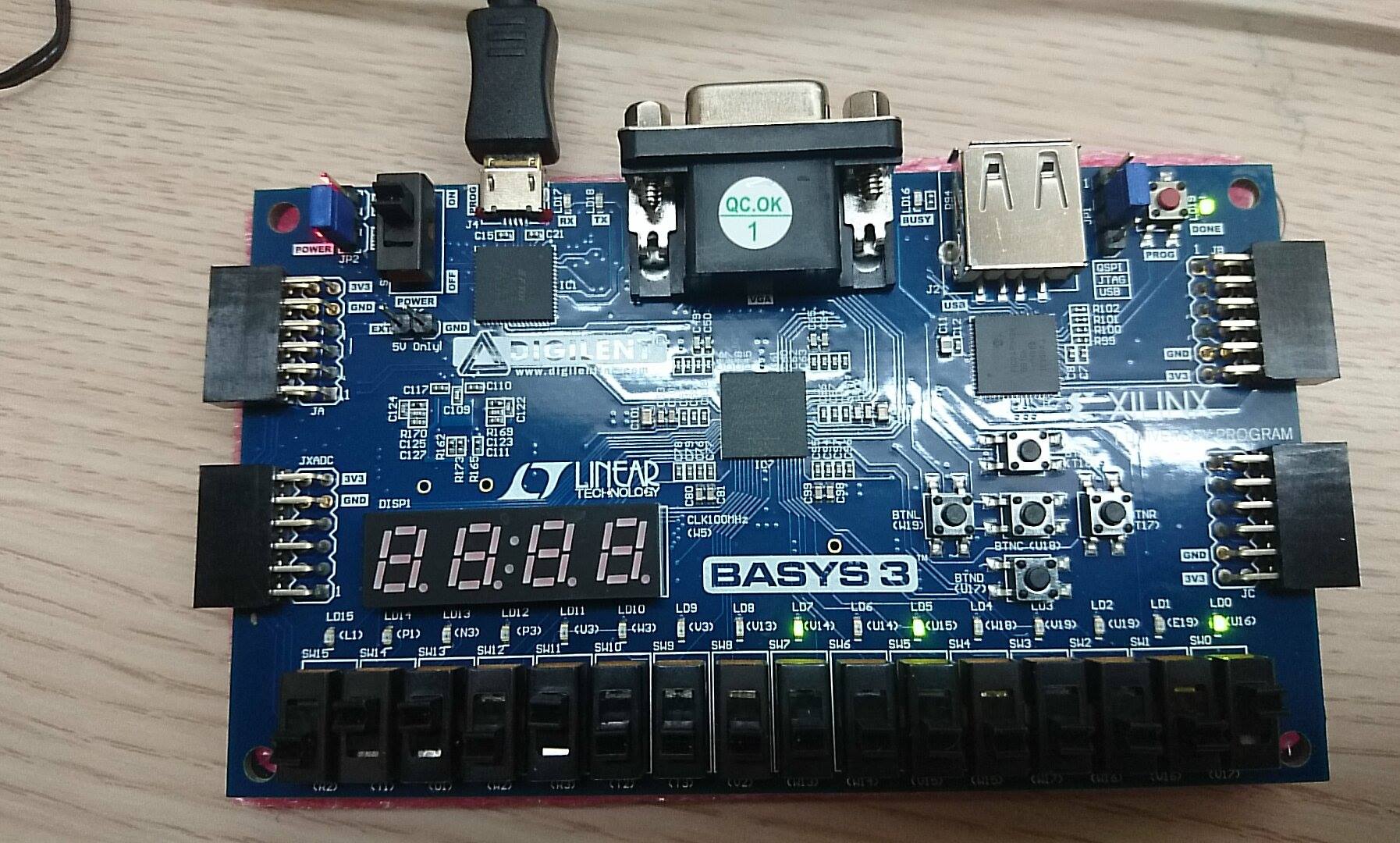
end

Result

1. 圖1 & 2即當value值為01101000時的情況，每經一個clk往昨移1個bit。
2. 圖3 & 4即當value值為10000001時的情況，每經一個clk往昨移1個bit。







Discussion

1. 這次實驗教會我如何操作shifter，並如何自己給定的輸入值。

**3.** **Use the idea from pre-lab1. We can do something on the seven-segment display. Assume we have the pattern of E, H, N, T, U for seven-segment display as shown below. Try to implement the scrolling pre-stored pattern NTHUEE with the four seven-segment displays.**

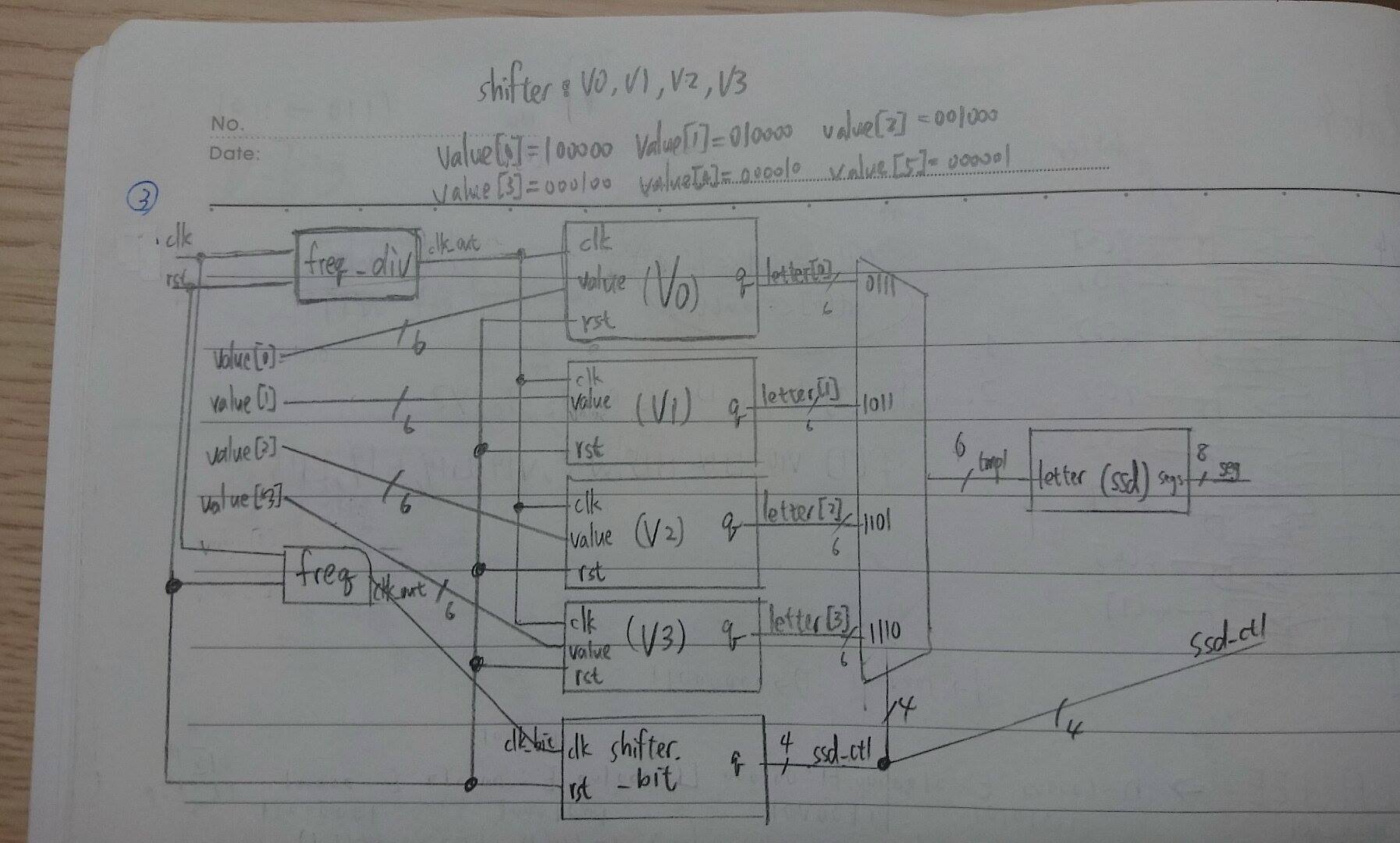
|  |  |
| --- | --- |
| **clk** | **clk\_out** |
| **W5** | **U16** |

Design Specification

input : clk,rst;

output : [3:0]ssd\_ctl, [7:0]seg;

block diagram :



Design Implementation

Logic function :

1. top：
2. 此top module，專門拿來呼叫其他小moudule的，並接一內線clk\_out作為shifter的clk。
3. 外加給定value值作為shifter的Input。

value[0] = 100000 ; value[1] = 010000 ; value[2] = 001000 ; value[3] = 000100

而此在ssd內，6-bit的one-hot會分別代表一個字母。

1. 還有內建一個mux，用由shifter\_bit得到的ssd\_ctl選擇七段顯示器要顯示何種字母。
2. freq\_div：

此為除頻的module，用來控制shifter的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=27'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的227倍。

1. freq：

此為除頻的module，用來控制shifter\_bit的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=16'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的216倍。

1. shifter：

此作用就是做移位的工作，使所有的bit往左移一位，最高位則變成最低位。

此module就是同時有6的DFFs在作用，clk每從0🡪1時，即移位。

always@(posedge clk or negedge rst)

if(~rst) q<=value; //此為在top給定的初始值

else

begin

q <= q>>1'b1; //每經一個clk，q往右移一個bit

q[5] <= q[0];

//ex. 010000🡪001000

End

1. shifter\_bit：

此作用就是做移位的工作，使所有的bit往左移一位，最高位則變成最低位。

此module就是同時有4的DFFs在作用，clk每從0🡪1時，即移位。

而此最重要的功能在於決定何時要顯示哪一個七段顯示器，因此會接上除頻後週期為clk的216倍的時間做為自己的clk，作用在於用即快速的交換顯示，使人眼誤以為是同時顯示的，因為七段顯示器一次只能顯示一種圖案，所以才要做這種處理。

always@(posedge clk or negedge rst)

if(~rst) q<= 4'b1110; //一開始只顯示最右邊的七段顯示器

else

begin

q <= q<<1'b1; ////每經一個clk，q往左移一個bit

q[0] <= q[3];

end

1. ssd：

當letter 為 6'd32: segs = 8'b11010101; //七段顯示器顯示n

當letter 為 6'd16: segs = 8'b11100001; //七段顯示器顯示t

當letter 為 6'd8: segs = 8'b10010001; //七段顯示器顯示H

當letter 為 6'd4: segs = 8'b10000011; //七段顯示器顯示U

當letter 為 6'd2: segs = 8'b01100001; //七段顯示器顯示E

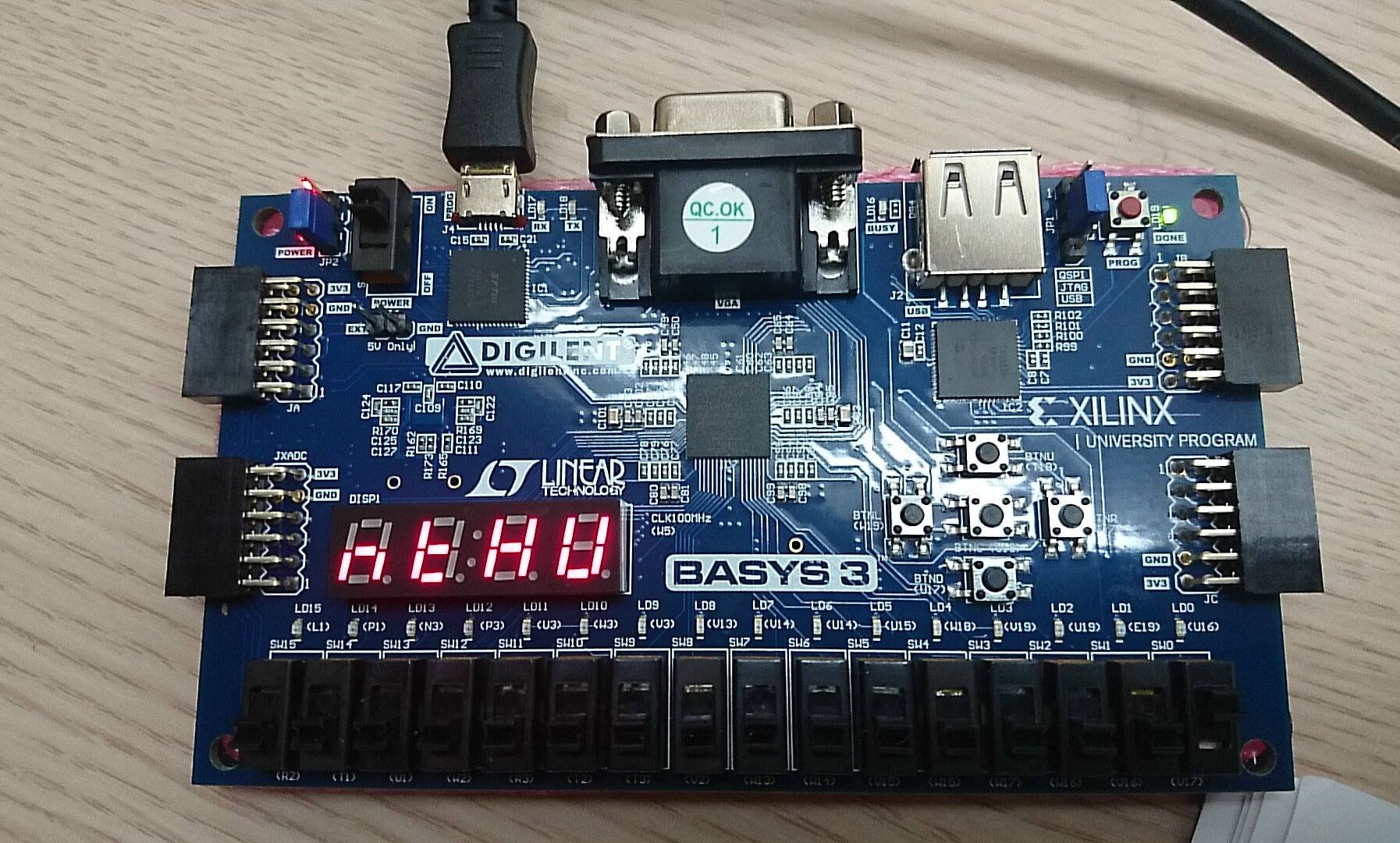
當letter 為 6'd1: segs = 8'b01100001; //七段顯示器顯示E

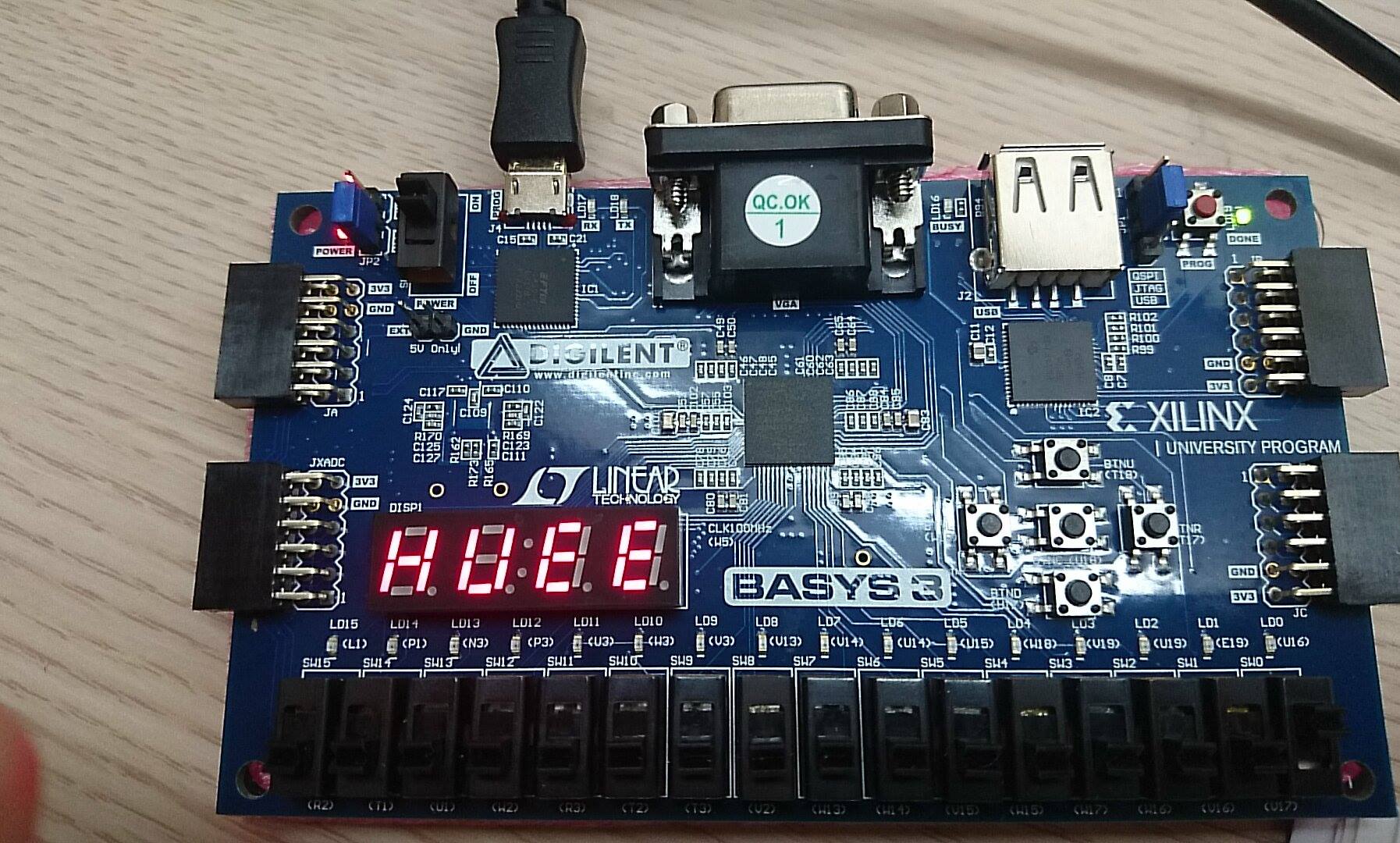
default: segs = 8'b00000000; //七段顯示器顯示8.

Result

1. rst=0(圖一)🡪其初始會顯示為”ntHU”，在只顯示最右邊的七段顯示器情況下只會顯示”U”。
2. rst=1(圖二&圖三)🡪為其左移的情形。

圖一

圖二

圖三

Discussion

1. 此設計想法是讓每一個七段顯示器各自擁有自己的shifter，只是初始值不同，使人眼產生字母是往左移的錯覺。
2. 可以用”>>”的方式代表位移，不用一個一個打出來

ex. q <= q>>1'b1 ; q[5] <= q[0] ; 即可代表右移一個bit的情形。

**4.** **(Bonus) Display 1010 in the seven-segment display. Use the DIP switch (one bit to indicate left/right shift, three bits with one hot to display the kind of shift operation) as the control input to implement the functional/arithmetic/barrel shifter. Use one push button to control the display of the number before/after the shift operation.**

|  |  |  |
| --- | --- | --- |
| **clk** | **clk\_out** | **push** |
| **W5** | **U16** | **T17** |

|  |  |  |  |
| --- | --- | --- | --- |
| **en[3]** | **en[2]** | **en[1]** | **en[0]** |
| **L3** | **P1** | **N3** | **P3** |

Design Specification

input : clk,rst;

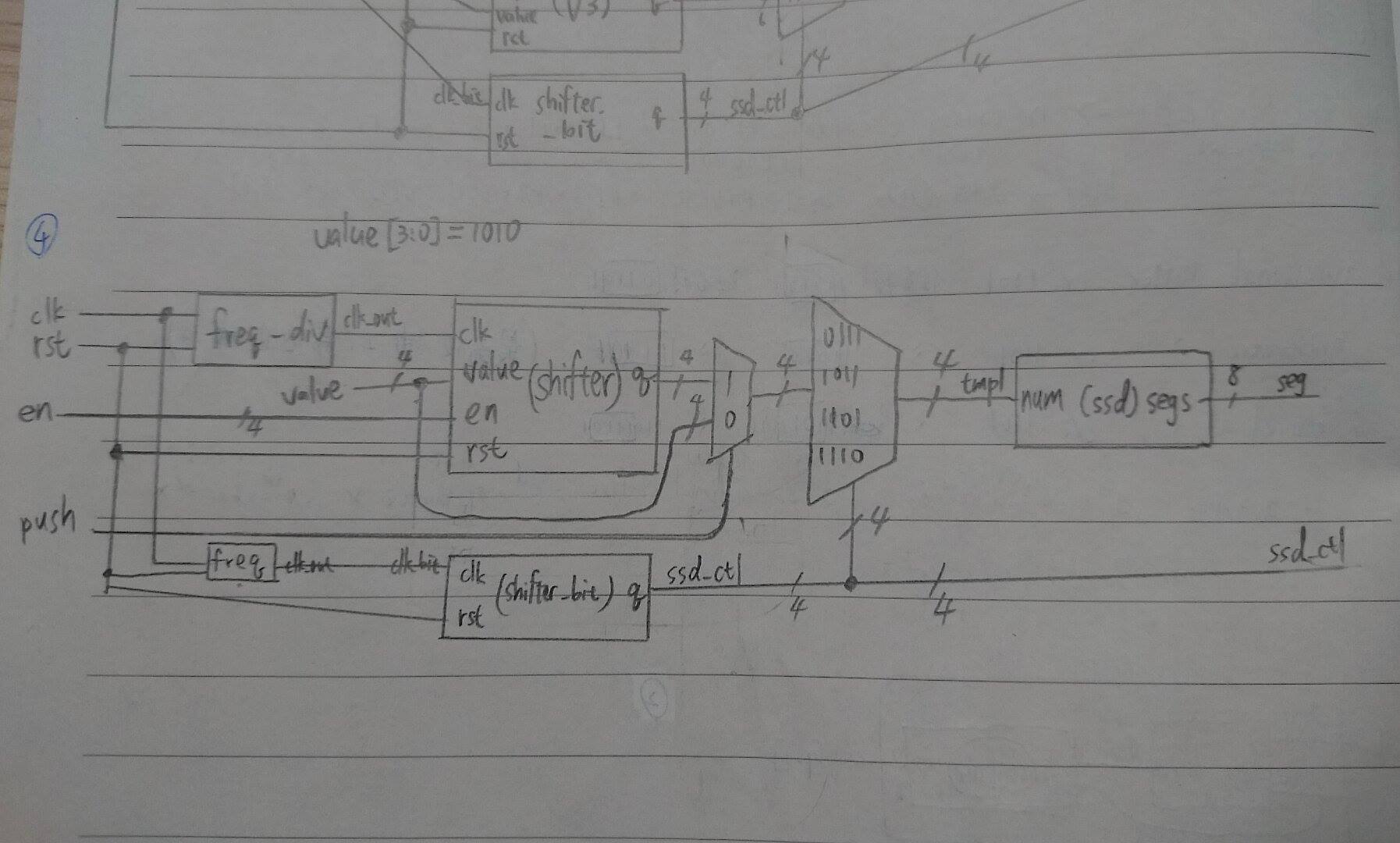
input [3:0]en; //DIP //en[0]控制往左往右移;en[1]為functional shifter

// en[2]為arithmetic shifter ; en[3]為barrel shifter

input :push; //按鈕

output : [3:0]ssd\_ctl, [7:0]seg;

block diagram :



Design Implementation

Logic function :

1. top：

1. 此top module，專門拿來呼叫其他小moudule的，並接一內線clk\_out作為shifter的clk。
2. 外加給定value值作為shifter的Input。value = 1010。
3. 還有內建二個mux，第一個用由shifter\_bit得到的ssd\_ctl選擇七段顯示器要顯示何種字母；第二個當push = 1(按下按鈕)七段顯示器顯示正在運作的值，push = 0七段顯示器顯示初始值1010，

2. freq\_div：

此為除頻的module，用來控制shifter的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=27'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的227倍。

1. freq：

此為除頻的module，用來控制shifter\_bit的速度。

always@\* cnt\_tmp={clk\_out,cnt}+1'b1;

always@(posedge clk or negedge rst) //DFFs

if(~rst){clk\_out,cnt}<=16'b0; //當rst=0時，{clk\_out,cnt} = 0(從0開始數)

else{clk\_out,cnt}<=cnt\_tmp;

//當rst=1 & clk從0🡪1時，把cnt\_tmp存入{clk\_out,cnt}

週期為clk的216倍。

1. shifter：

此作用就是做移位的工作，並用mux決定要產生何種功能。

//en[0] = 1表示往右移; en[0] = 0表示往左移

always@(posedge clk or negedge rst)

if(~rst) q <= value;

else

begin

//en[3] = 1做barrel shifter。當往左移時，最低位補最高位的數；往右移時，最高位補最低位的數

if(en==4'b1001)

begin

q <= q>>1'b1;

q[3] <= q[0];

end

else if(en==4'b0001)

begin

q <= q<<1'b1;

q[0] <= q[3];

end

//en[2] = 1做arithmetic shifter。當往左移時，最低位補”0”；往右移時，最高位補最高位的數

else if(en==4'b1010)

begin

q <= q>>1'b1;

q[3] <= q[3];

end

else if(en==4'b0010)

begin

q <= q<<1'b1;

q[0] <= 0;

end

//en[1] = 1做functional shifter。當往左移時，最低位補”0”；往右移時，最高位補”0”

else if(en==4'b1100)

begin

q <= q>>1'b1;

q[3] <= 0;

end

else if(en==4'b0100)

begin

q <= q<<1'b1;

q[0] <= 0;

end

else

begin

q <= q;

end

end

1. shifter\_bit：

此作用就是做移位的工作，使所有的bit往左移一位，最高位則變成最低位。

此module就是同時有4的DFFs在作用，clk每從0🡪1時，即移位。

而此最重要的功能在於決定何時要顯示哪一個七段顯示器，因此會接上除頻後週期為clk的216倍的時間做為自己的clk，作用在於用即快速的交換顯示，使人眼誤以為是同時顯示的，因為七段顯示器一次只能顯示一種圖案，所以才要做這種處理。

always@(posedge clk or negedge rst)

if(~rst) q<= 4'b1110; //一開始只顯示最右邊的七段顯示器

else

begin

q <= q<<1'b1; ////每經一個clk，q往左移一個bit

q[0] <= q[3];

end

1. ssd：

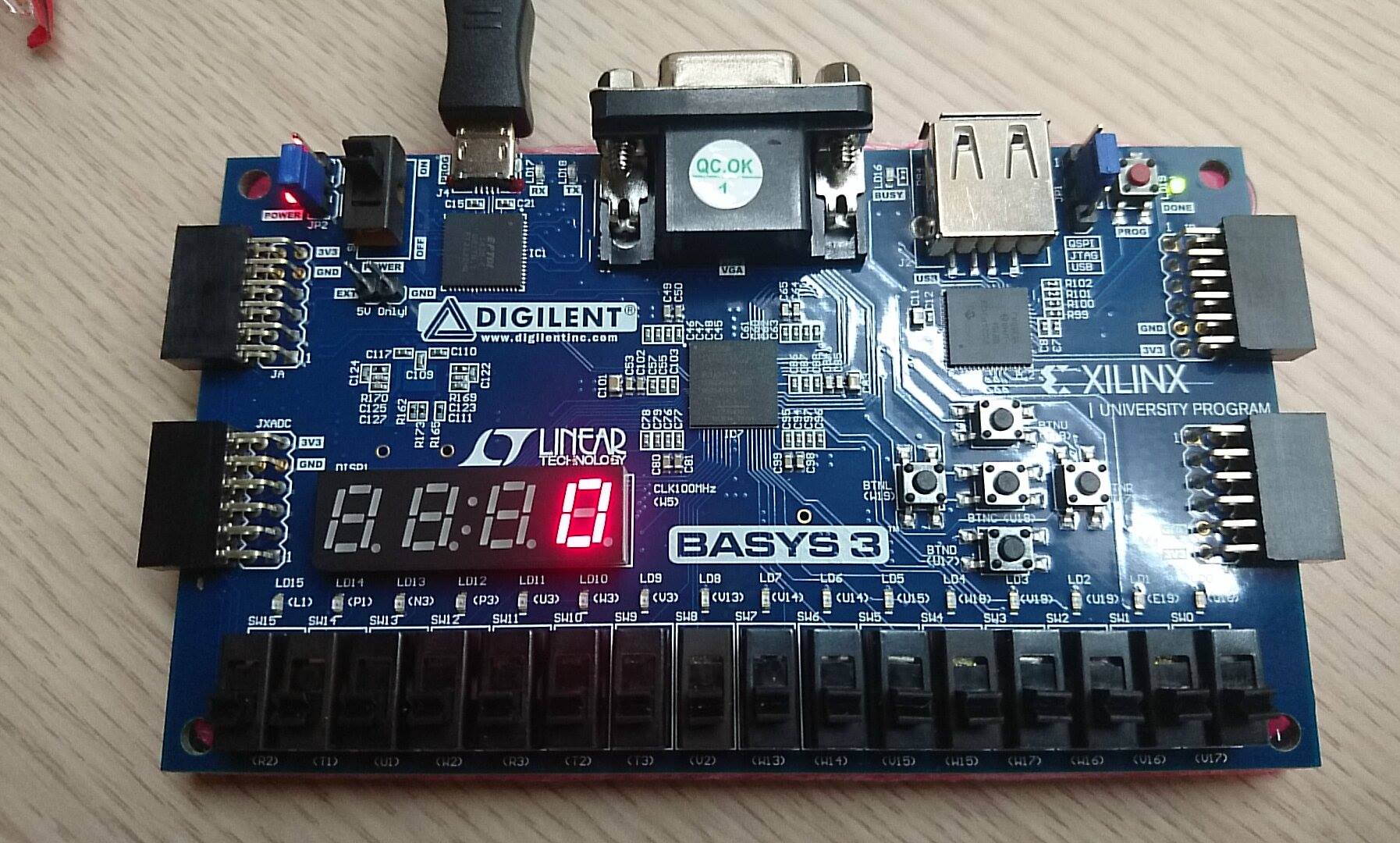
always@\*

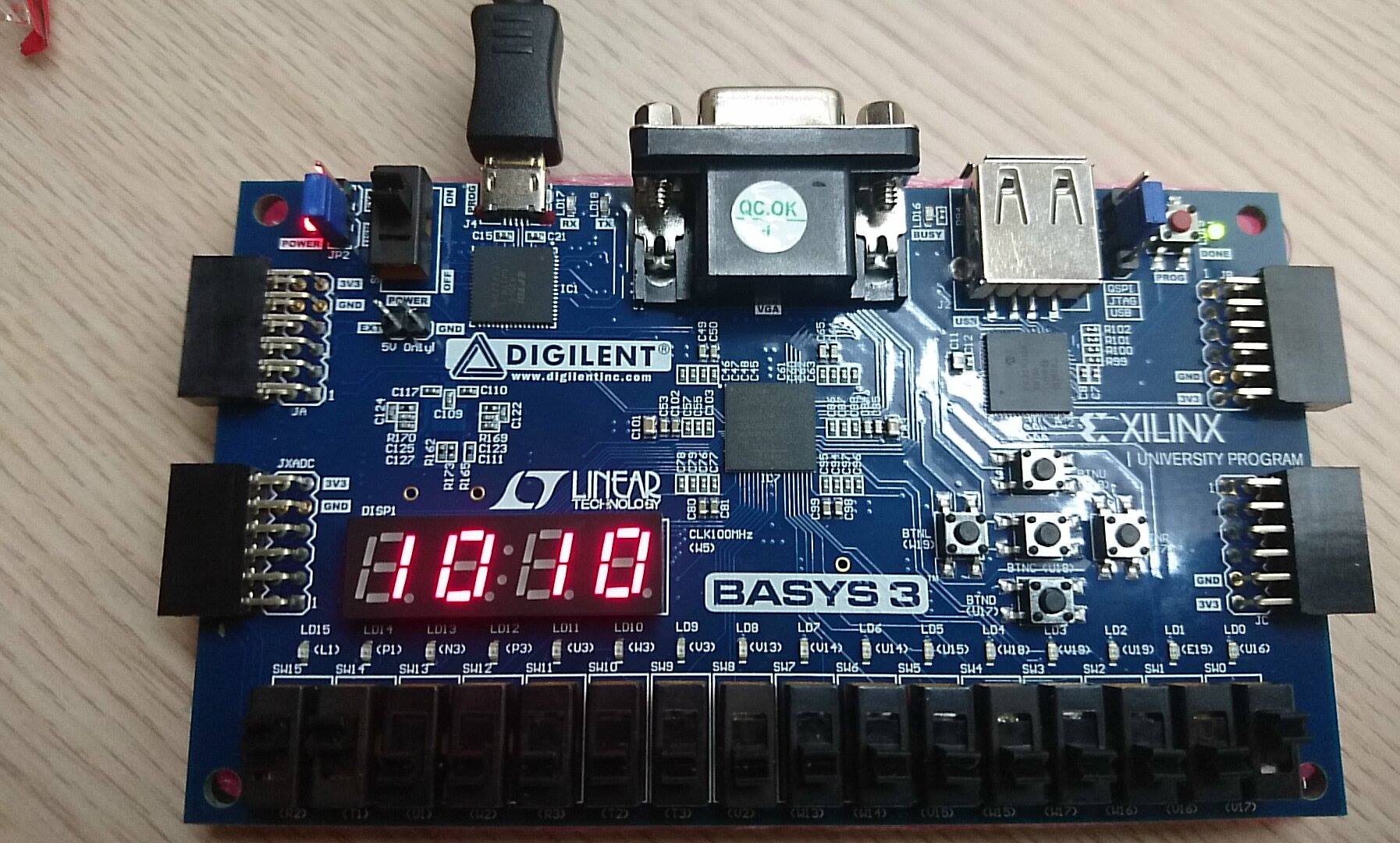
if(num==1'b0) segs = 8'b00000011; //當num為0七段顯示器顯示0

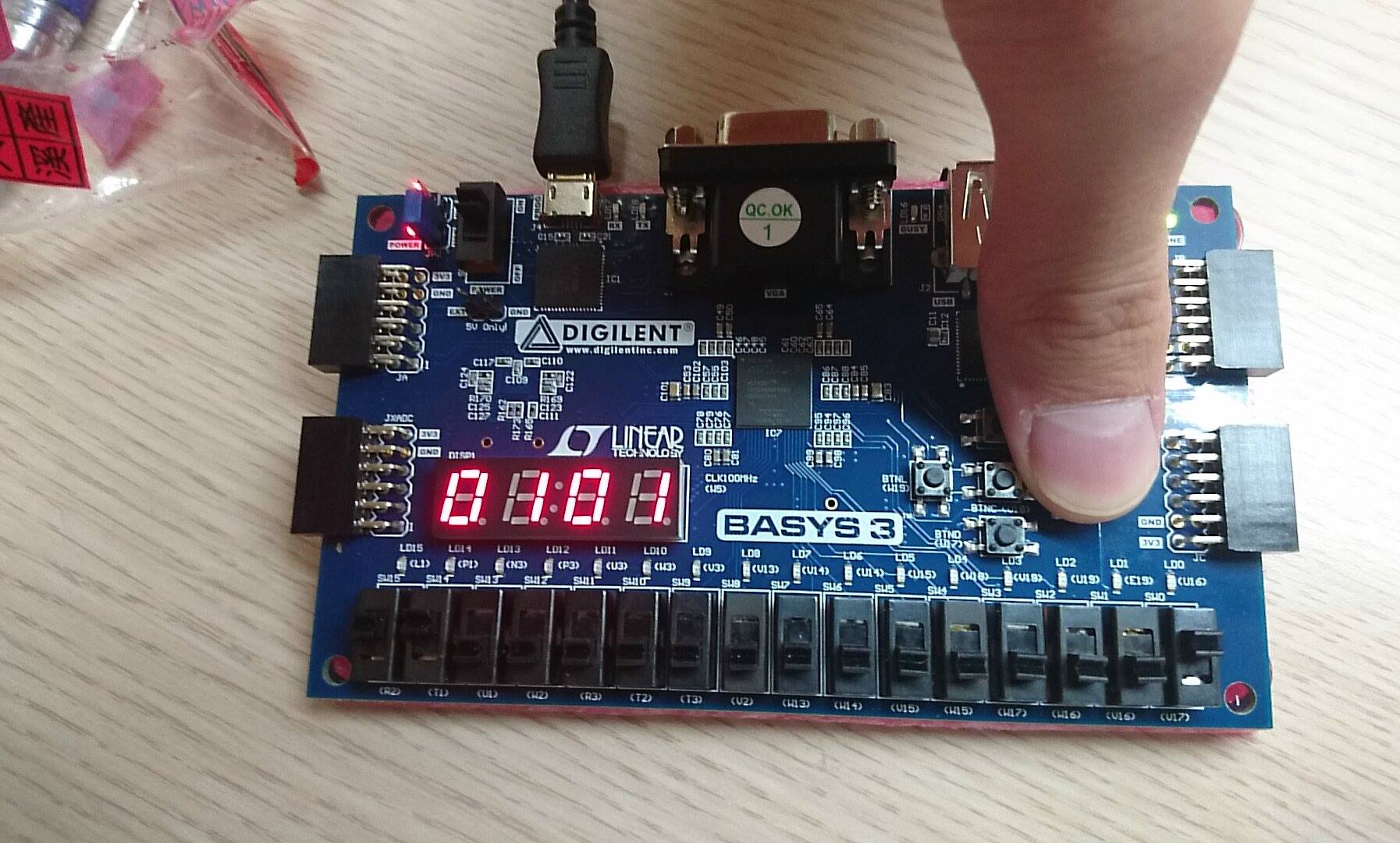
else segs = 8'b10011111; //當num為1七段顯示器顯示1

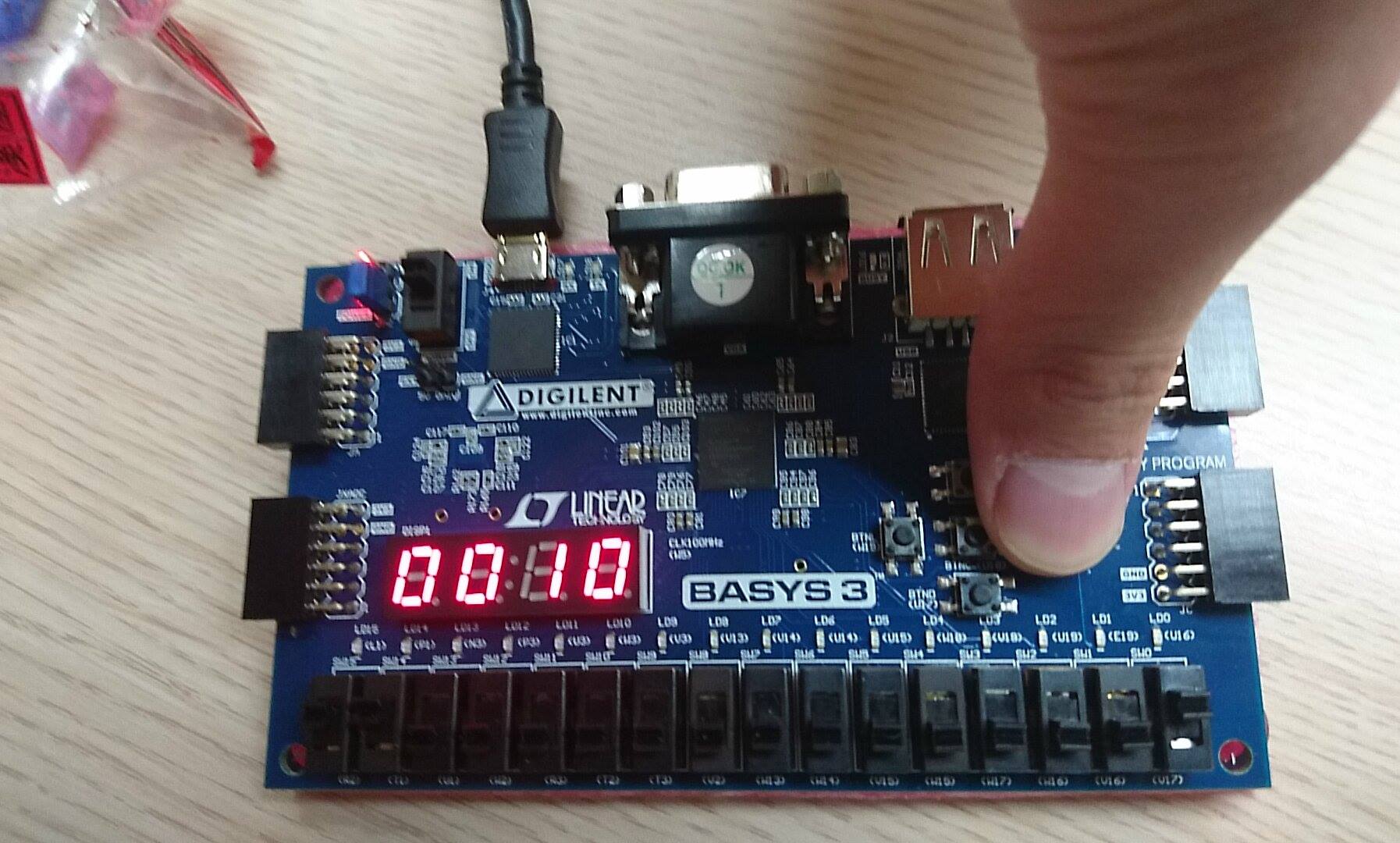
Result(顯示器只有第3個數字亮)

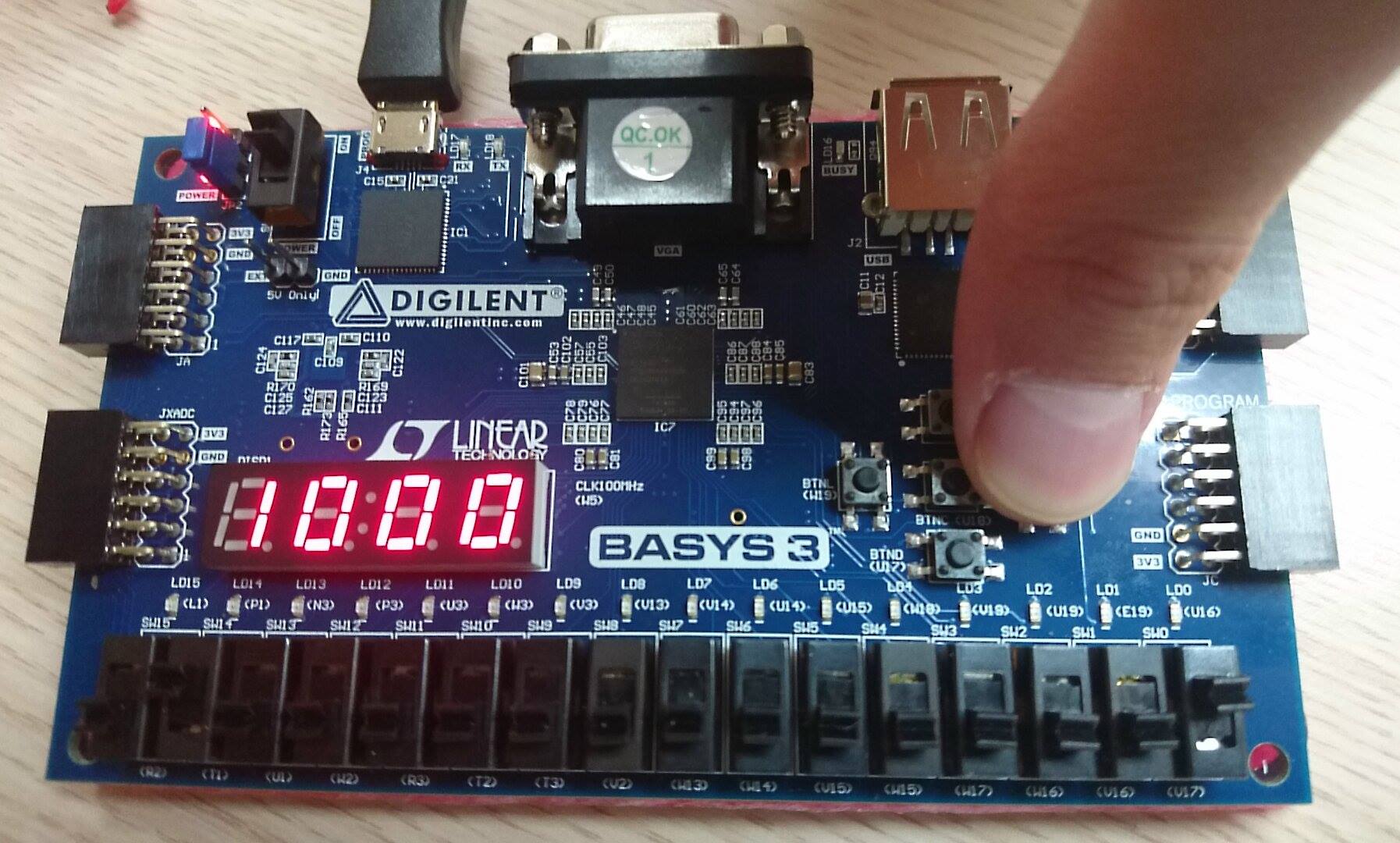
* 1. rst=0(圖一)🡪其初始會顯示為”1010”，在只顯示最右邊的七段顯示器情況下只會顯示”0”。
  2. 圖二：當push=0時(未按下按鈕)，不管怎麼開開關，仍然顯示初始值”1010”。
  3. 圖三、四、五：當en[1] 為1時做functional shifter，當往左移時(en[0]=0)，最低位補”0”；往右移時(en[0]=1)，最高位補”0”。
  4. 圖六、七：當en[2] 為1時做arithmetic shifter，當往左移時，最低位補”0”；往右移時，最高位補最高位的數。
  5. 圖八：當en[3]為1時做barrel shifter，當往左移時，最低位補最高位的數；往右移時，最高位補最低位的數。

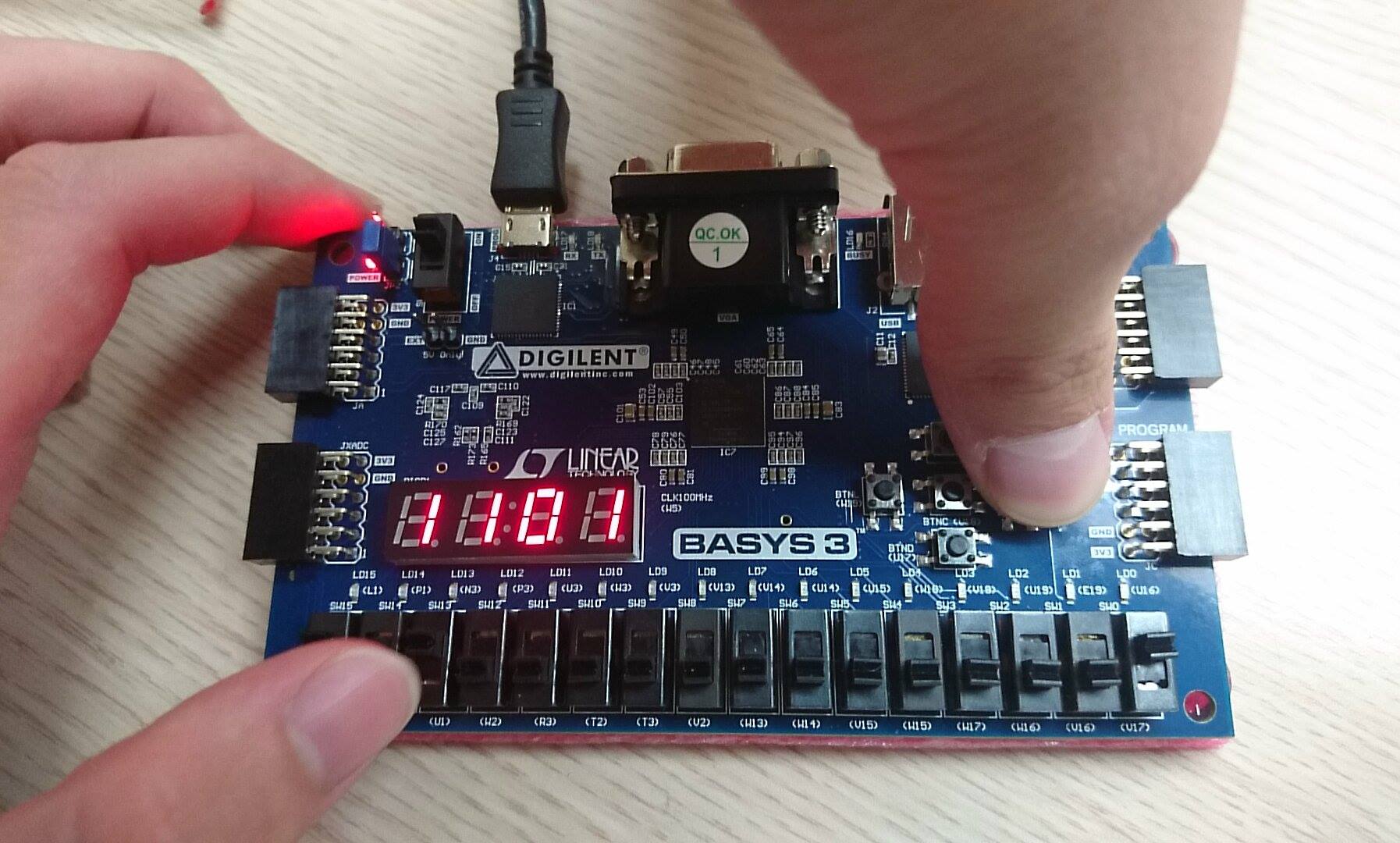
圖一

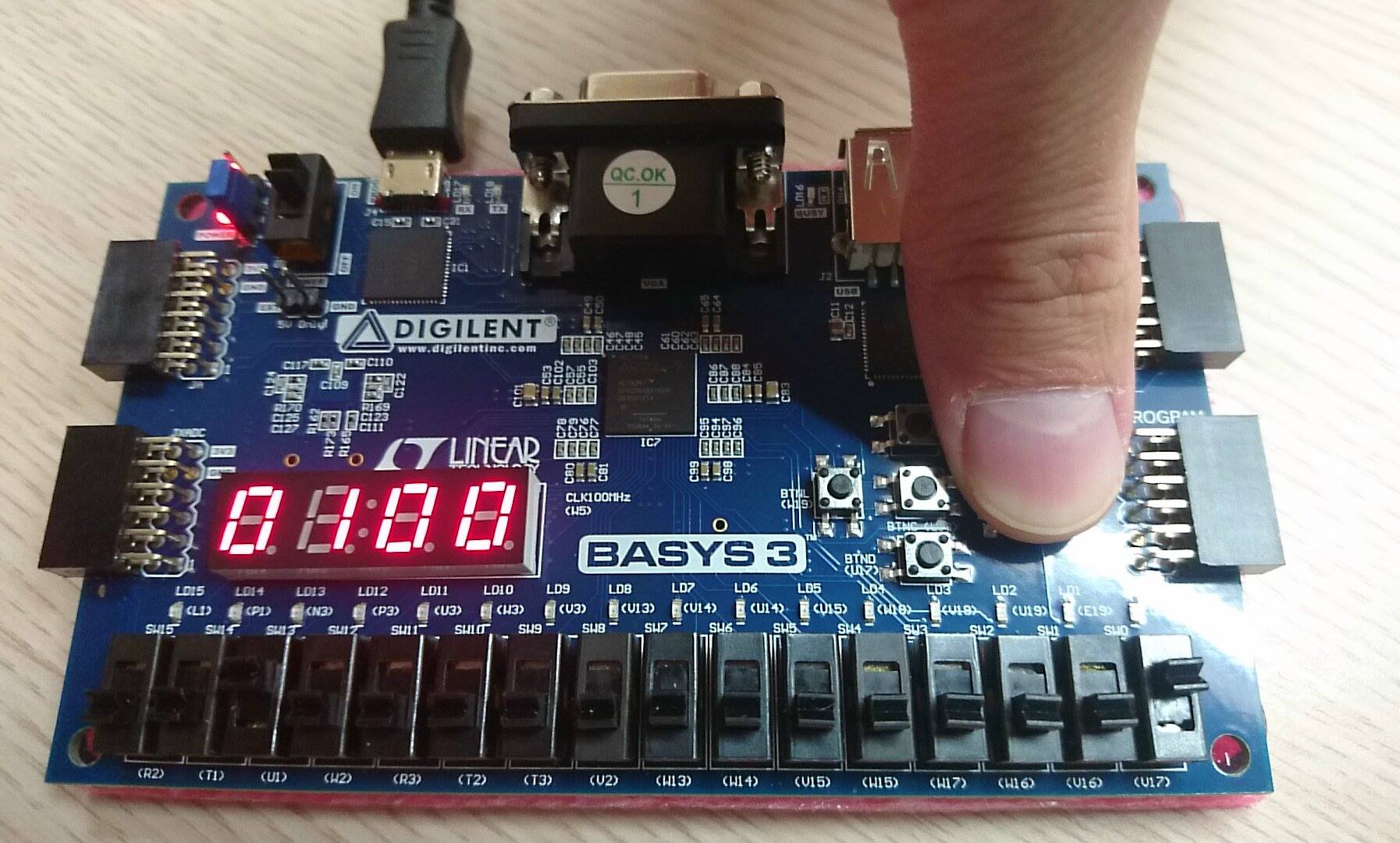
圖二

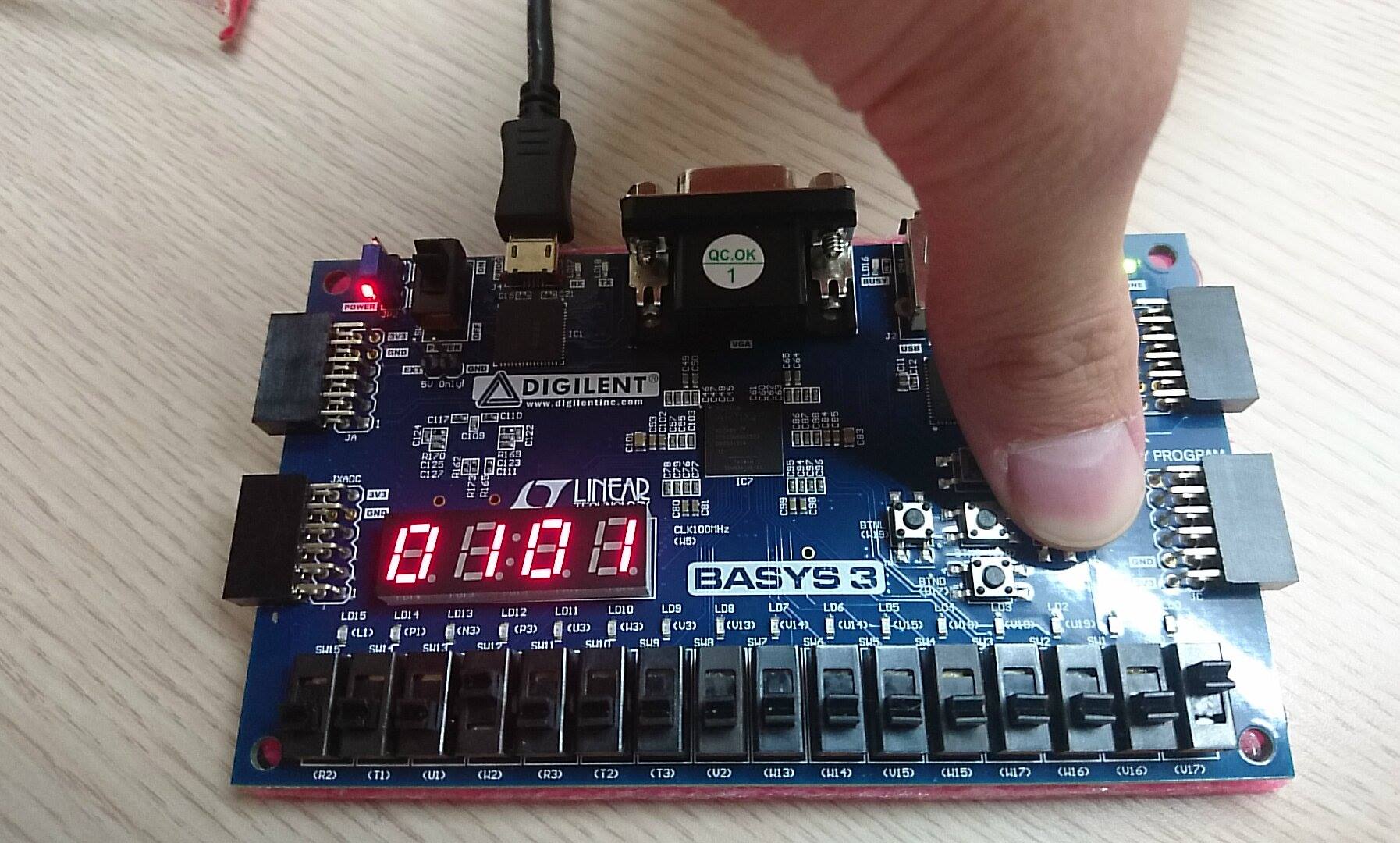
圖三

圖四

圖五

圖六

圖七

圖八

Discussion

1. functional shifter：當往左移時(en[0]=0)，最低位補”0”；往右移時(en[0]=1)，最高位補”0”。
2. arithmetic shifter：當往左移時，最低位補”0”；往右移時，最高位補最高位的數。
3. barrel shifter，當往左移時，最低位補最高位的數；往右移時，最高位補最低位的數。

**Conclusion**：

這次還蠻好玩的，感覺可以把上次Lab3學到的東西做應用，還可以外加學一些功能，像是跑馬燈其實是每個七段顯示器內部自己在做循環顯示而已，真的還蠻實用的。

Reference：

馬席彬教授的上課講義