國立高雄第一科技大學
National Kaohsiung First University of Science and Technology

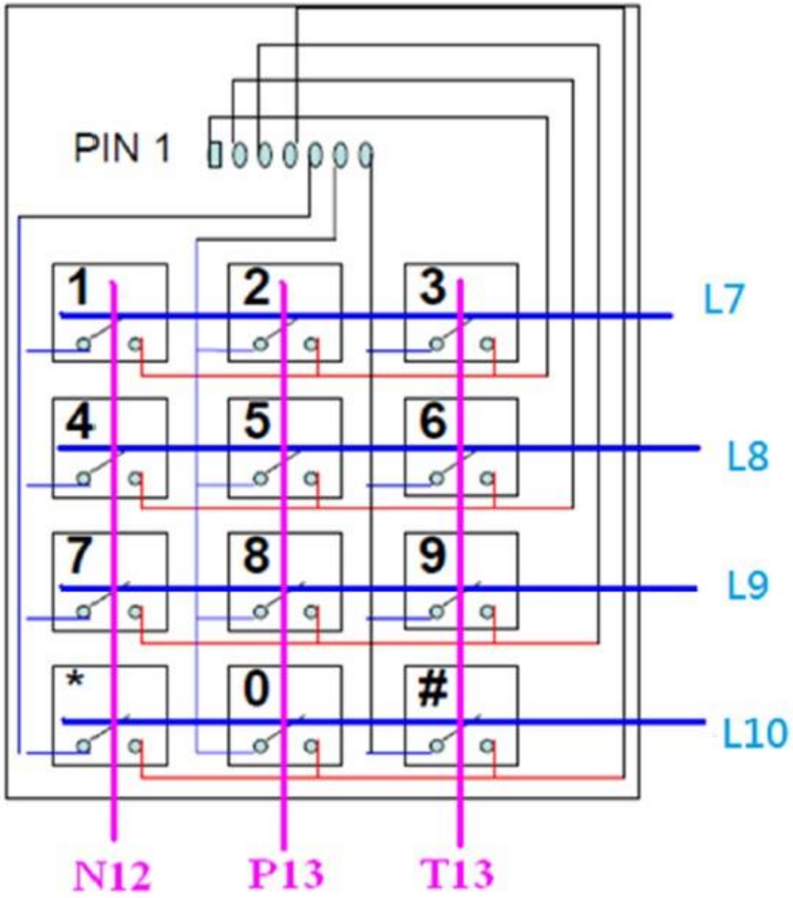
基本FPGA實習

Lab7 Keypad

Keypad

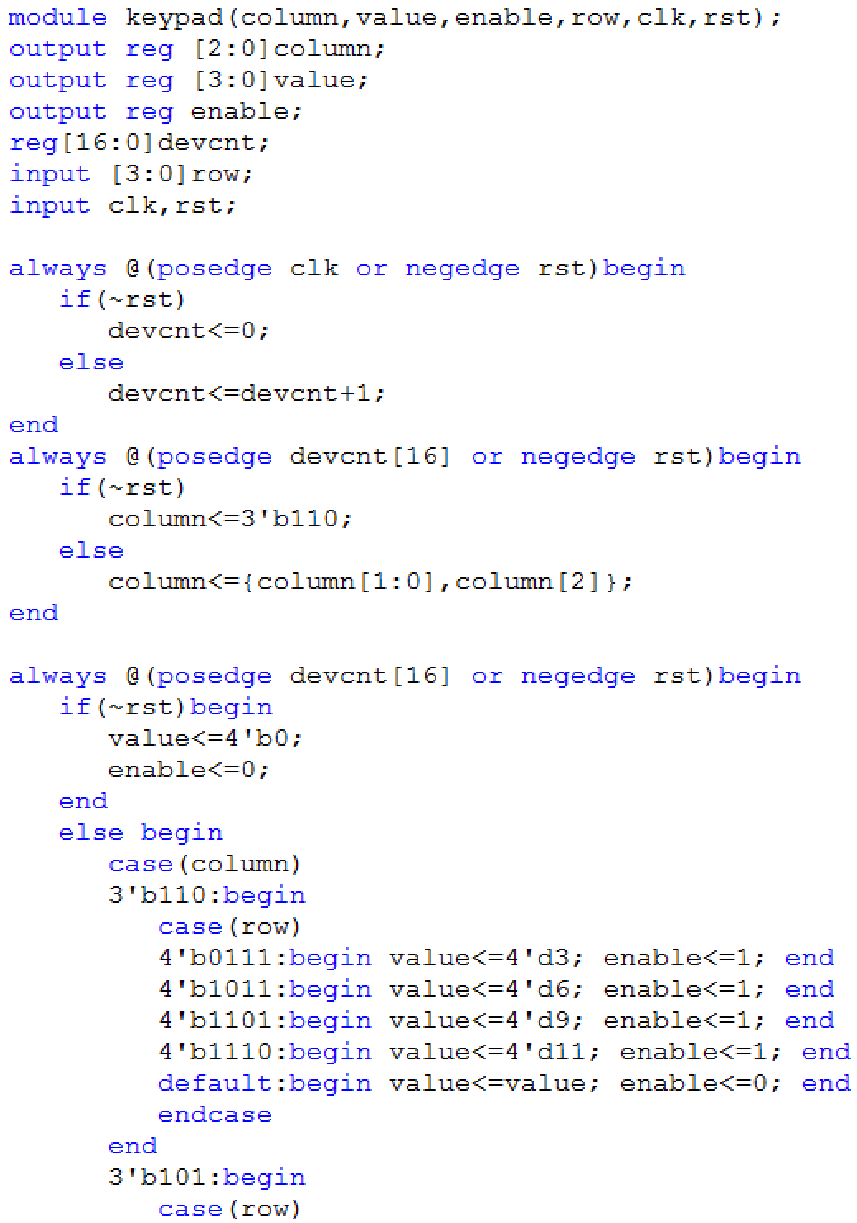
此圖Row 4Bit為Output端，Column 3Bit為Input端，利用解碼方式得到按鍵值，如右圖所示:

當Row Output輸出0，則Keypad被按下的Column會讀到相對應的值，如Row當下為011，則當按下1時，Column會讀到0111的訊號，因此當Row為011且Column為0111時則可得知是按到1。

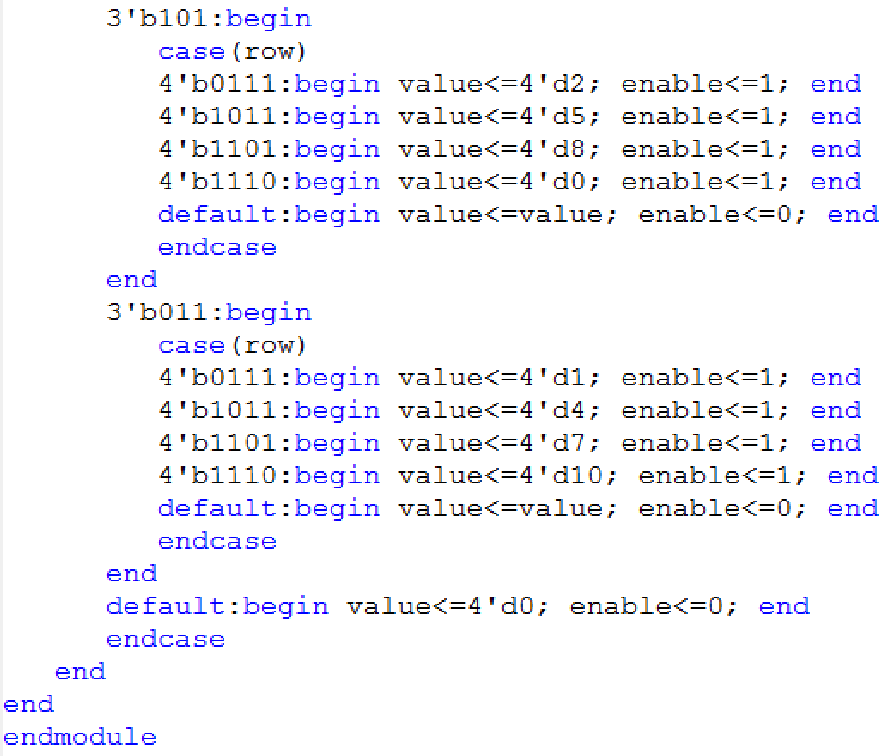


Keypad對應解碼程式如右圖

請將enable與value利用LED做輸出

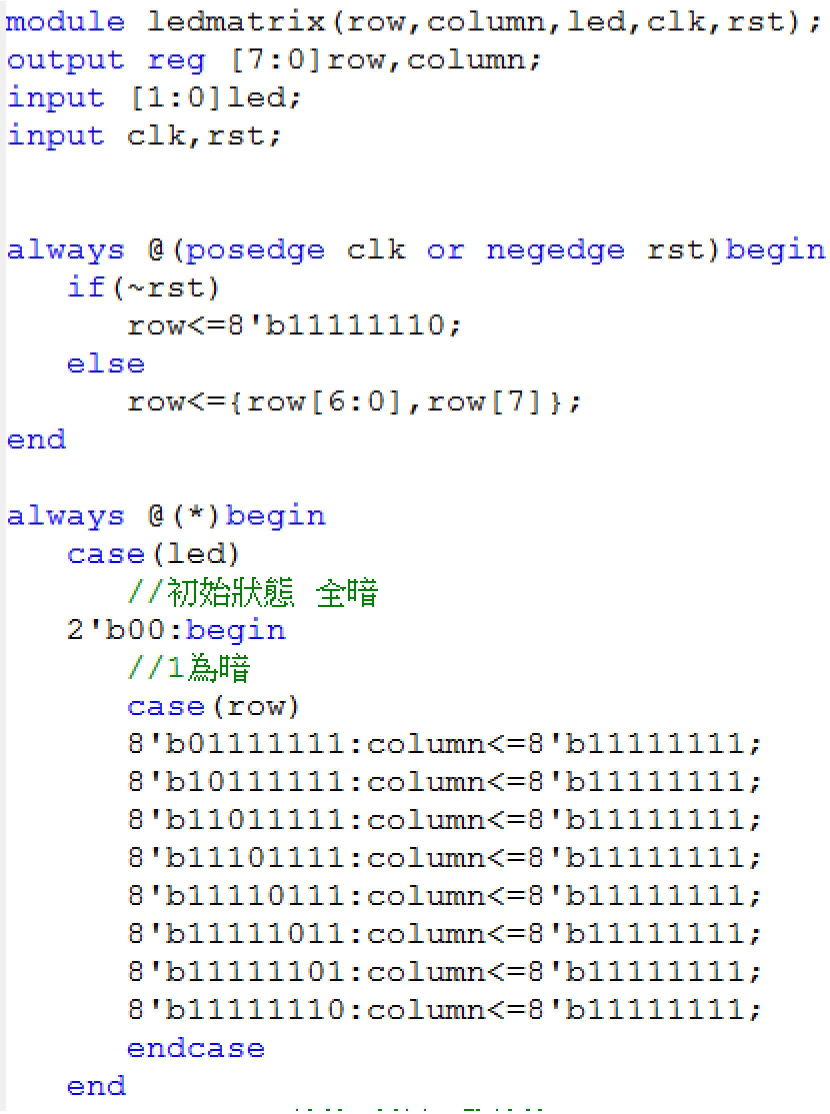


```
module keypad(column,value,enable,row,clk,rst);
output reg [2:0]column;
output reg [3:0]value;
output reg enable;
reg[16:0]devcnt;
input [3:0]row;
input clk,rst;

always @(posedge clk or negedge rst)begin
   if(~rst)
      devcnt<=0;
   else
      devcnt<=devcnt+1;
end
always @(posedge devcnt[16] or negedge rst)begin
   if(~rst)
      column<=3'b110;
   else
      column<={column[1:0],column[2]};
end

always @(posedge devcnt[16] or negedge rst)begin
   if(~rst)begin
      value<=4'b0;
      enable<=0;
   end
   else begin
      case(column)
      3'b110:begin
         case(row)
         4'b0111:begin value<=4'd3; enable<=1; end
         4'b1011:begin value<=4'd6; enable<=1; end
         4'b1101:begin value<=4'd9; enable<=1; end
         4'b1110:begin value<=4'd11; enable<=1; end
         default:begin value<=value; enable<=0; end
         endcase
      end
      3'b101:begin
         case(row)
```



```
      3'b101:begin
         case(row)
         4'b0111:begin value<=4'd2; enable<=1; end
         4'b1011:begin value<=4'd5; enable<=1; end
         4'b1101:begin value<=4'd8; enable<=1; end
         4'b1110:begin value<=4'd0; enable<=1; end
         default:begin value<=value; enable<=0; end
         endcase
      end
      3'b011:begin
         case(row)
         4'b0111:begin value<=4'd1; enable<=1; end
         4'b1011:begin value<=4'd4; enable<=1; end
         4'b1101:begin value<=4'd7; enable<=1; end
         4'b1110:begin value<=4'd10; enable<=1; end
         default:begin value<=value; enable<=0; end
         endcase
      end
      default:begin value<=4'd0; enable<=0; end
      endcase
   end
end
endmodule
```

ledmatrix



```
module ledmatrix(row,column,led,clk,rst);
output reg [7:0]row,column;
input [1:0]led;
input clk,rst;


always @(posedge clk or negedge rst)begin
   if(~rst)
      row<=8'b11111110;
   else
      row<={row[6:0],row[7]};
end

always @(*)begin
   case(led)
      //初始狀態 全暗
   2'b00:begin
      //1為暗
      case(row)
      8'b01111111:column<=8'b11111111;
      8'b10111111:column<=8'b11111111;
      8'b11011111:column<=8'b11111111;
      8'b11101111:column<=8'b11111111;
      8'b11110111:column<=8'b11111111;
      8'b11111011:column<=8'b11111111;
      8'b11111101:column<=8'b11111111;
      8'b11111110:column<=8'b11111111;
      endcase
   end
```

以下請自行撰寫

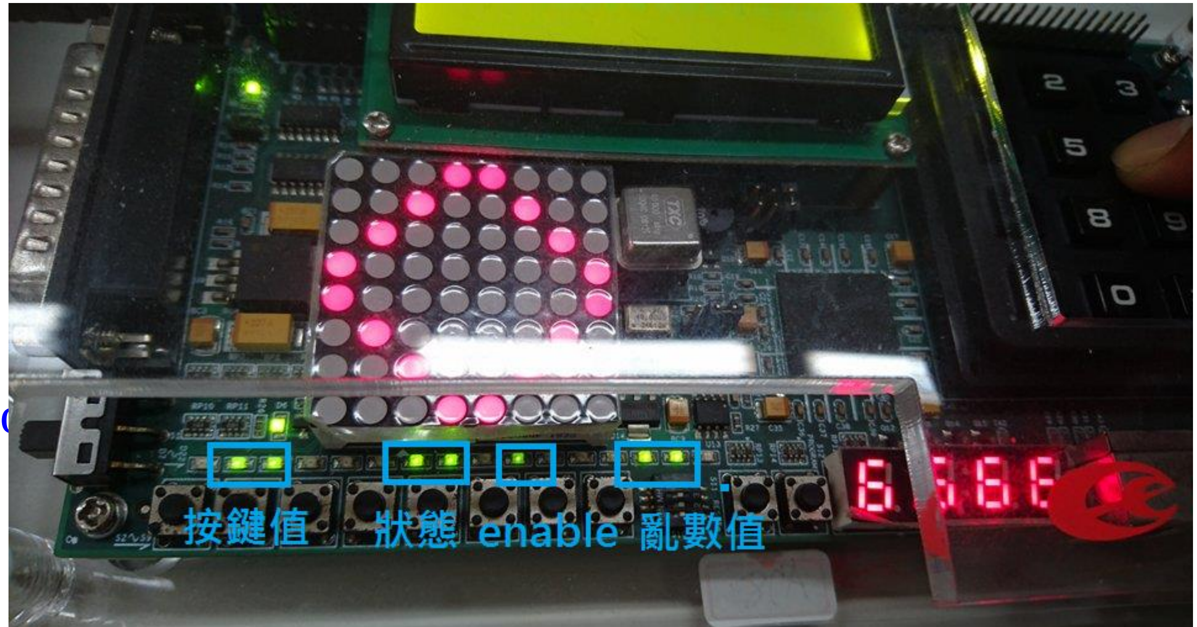
作業題目：

試做一個猜數字的小遊戲，當按下按鈕時，取得一亂數值作為答案，並led顯示出來，隨後當按下鍵盤與答案做判斷，並用七段顯示鍵盤數字

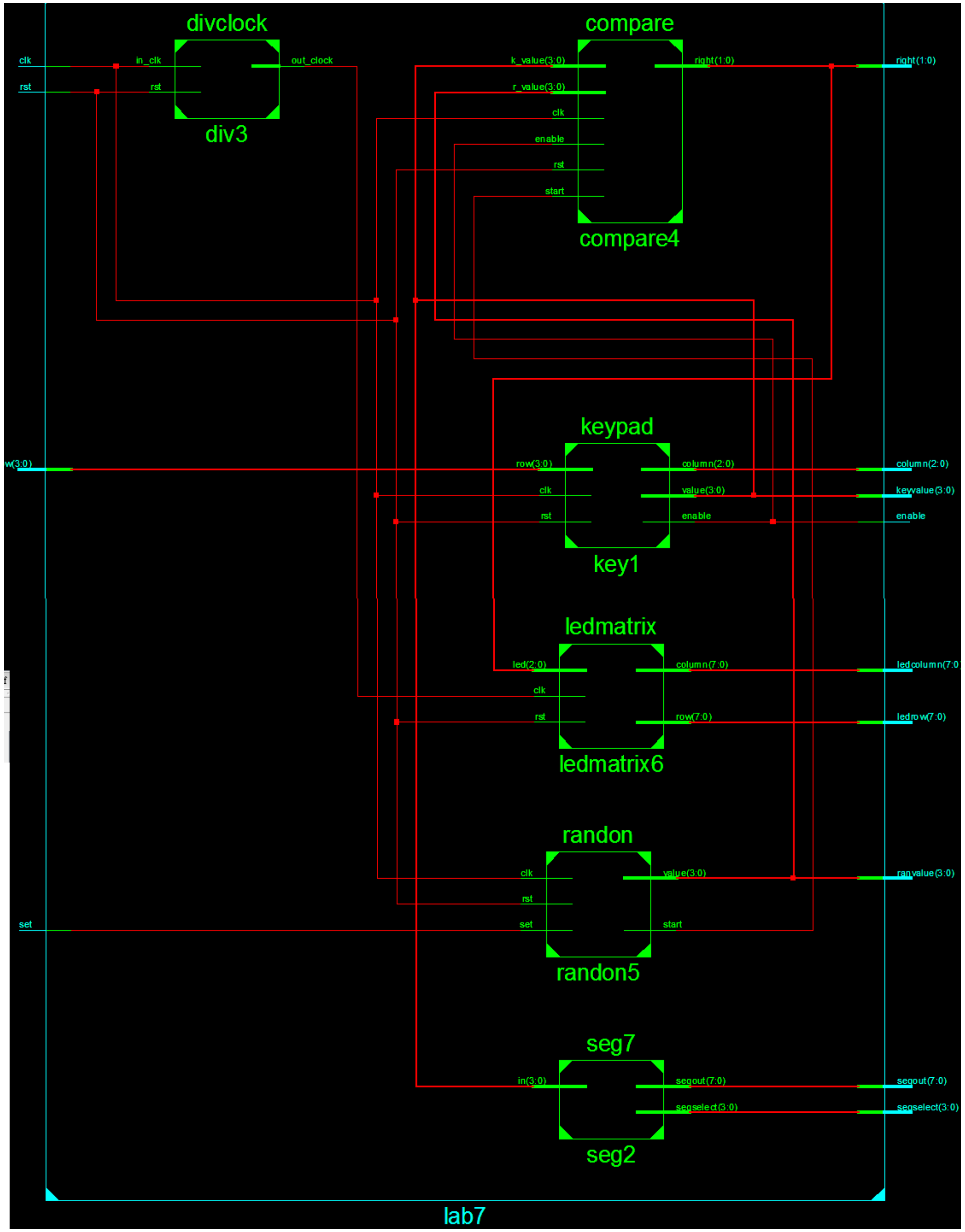
當按下數字小於答案，在矩陣顯示↑
按下數字大於答案，在矩陣顯示↓
按下數字大於答案，在矩陣顯示O

需用led 顯示enable 燈亮表示按下
燈暗表示沒按下

作業內容需包含：
1.題目
2.Verilog Code
3.RTL View
4.TestBench Setting & Wavef
5.SMIMS Simulator Result



電路參考圖


divclock
clk
in_clk
out_clock
rst
rst
div3
compare
k_value(3:0)
right(1:0)
right(1:0)
r_value(3:0)
clk
enable
rst
start
compare4
keypad
w(3:0)
row(3:0)
column(2:0)
column(2:0)
clk
value(3:0)
keyvalue(3:0)
rst
enable
enable
key1
ledmatrix
led(2:0)
column(7:0)
ledcolumn(7:0
clk
rst
row(7:0)
ledrow(7:0)
ledmatrix6
randon
clk
value(3:0)
ranvalue(3:0)
rst
set
set
start
randon5
seg7
in(3:0)
segout(7:0)
segout(7:0)
segselect(3:0)
segselect(3:0)
seg2
lab7

Ucf參考

```
NET "set" CLOCK_DEDICATED_ROUTE = FALSE;

NET "column[2]" LOC = N12;
NET "column[1]" LOC = P13;
NET "column[0]" LOC = T13;
NET "clk" LOC = P9;
NET "enable" LOC = J12;
NET "rst" LOC = K5;
NET "row[3]" LOC = L7;
NET "row[2]" LOC = L8;
NET "row[1]" LOC = L9;
NET "row[0]" LOC = L10;
NET "segout[7]" LOC = T11;
NET "segout[6]" LOC = T10;
NET "segout[5]" LOC = P10;
NET "segout[4]" LOC = N10;
NET "segout[3]" LOC = M10;
NET "segout[2]" LOC = R9;
NET "segout[1]" LOC = T9;
NET "segout[0]" LOC = N9;
NET "segselect[3]" LOC = N11;
NET "segselect[2]" LOC = P11;
NET "segselect[1]" LOC = P12;
NET "segselect[0]" LOC = R13;
```

```
NET "set" LOC = G11;
NET "ranvalue[3]" LOC = J16;
NET "ranvalue[2]" LOC = K16;
NET "ranvalue[1]" LOC = H14;
NET "ranvalue[0]" LOC = J14;
NET "keyvalue[3]" LOC = M14;
NET "keyvalue[2]" LOC = M13;
NET "keyvalue[1]" LOC = K13;
NET "keyvalue[0]" LOC = L13;
NET "ledcolumn[7]" LOC = H16;
NET "ledcolumn[6]" LOC = H15;
NET "ledcolumn[5]" LOC = F16;
NET "ledcolumn[4]" LOC = G16;
NET "ledcolumn[3]" LOC = G14;
NET "ledcolumn[2]" LOC = H13;
NET "ledcolumn[1]" LOC = F15;
NET "ledcolumn[0]" LOC = E16;
NET "ledrow[7]" LOC = F14;
NET "ledrow[6]" LOC = G13;
NET "ledrow[5]" LOC = F13;
NET "ledrow[4]" LOC = E14;
NET "ledrow[3]" LOC = D15;
NET "ledrow[2]" LOC = D16;
NET "ledrow[1]" LOC = D14;
NET "ledrow[0]" LOC = E13;
```

步驟說明:

1.確定keypad 值 可以正常顯示在七段

2.產生亂數值 led燈顯示

3.Compare 電路比較2個數值 狀態是否有變化
可用測試程式觀看

4.矩陣除頻後 是否正常顯示