



*Department of Computer Science and Information Engineering*

*National Cheng Kung University*

# ***LAB - 11***

陳培殷

國立成功大學 資訊工程系





*Department of Computer Science and Information Engineering*

*National Cheng Kung University*

# ***Lab : Keypad Controller***

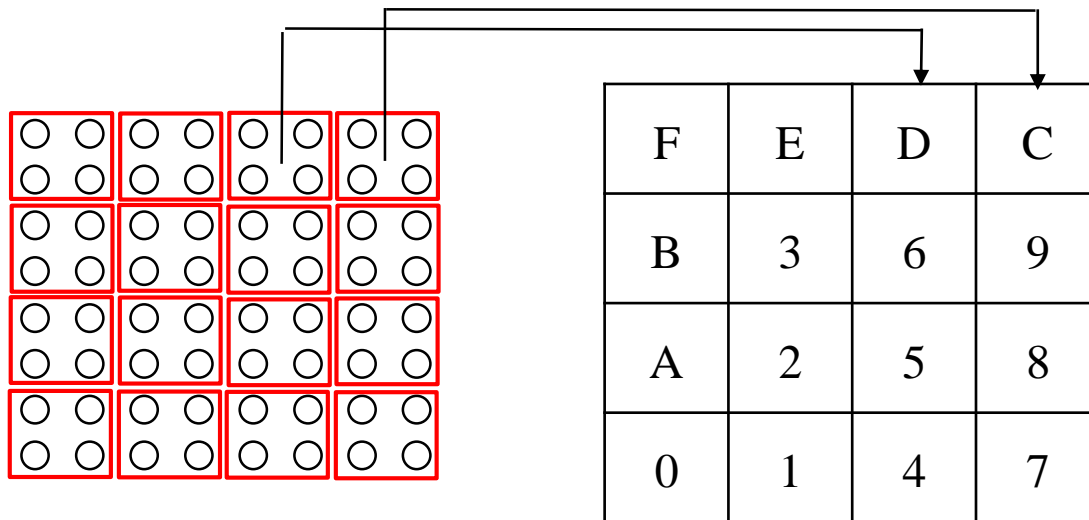
---

# Keypad controller (1/2)

- Please design a **Keypad controller** by using the following components:
  - ❑ 1 LED Dot Matrix Displays
  - ❑ 1 keypad
  - ❑ 1 reset button
  - ❑ 1 seven-segment display

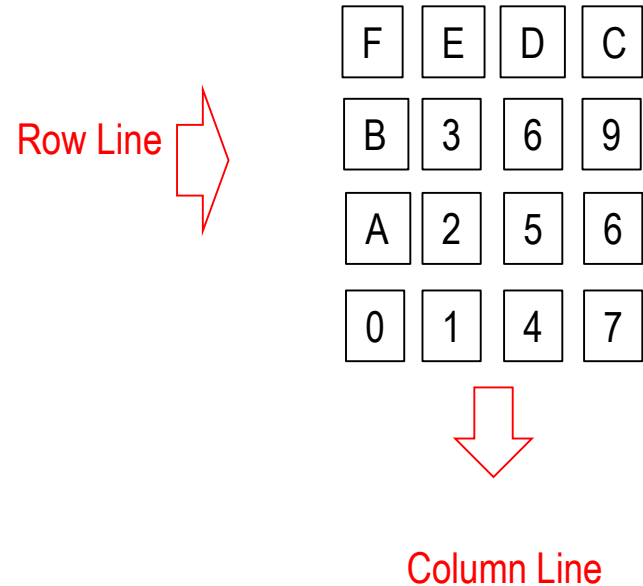
# Dot matrix controller (2/2)

- Clock Frequency : 100 Hz (button detection)
- Clock Frequency : 2500 Hz (dot matrix)
- Basic requirements :
  - When the reset button is pressed, dot matrix need to set off and seven-segment display needs to set 0.
  - When any keypad be pressed, the corresponding area in dot matrix needs to set on and seven-segment display needs to display the corresponding value.



# Keypad(1/8)

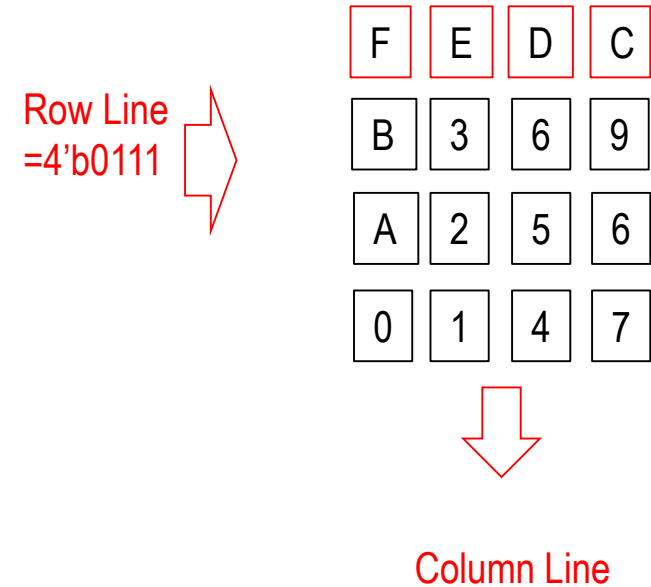
- DE0-CV的external board提供一組keypad
- 一組keypad需要8條控制線
  - 四條輸入到keypad的Row Line
  - 四條輸出到FPGA的Column Line
- Row Line決定哪個Row被開啟
  - Row Line為0時，該row被開啟
  - Row Line為1時，該row被關閉
- Column Line決定該Row上，哪一個按鍵被按下去
  - Column Line為0時，按鍵有按
  - Column Line為1時，按鍵沒有按



# Keypad(2/8)

## ■ Example

- 當 Row Line = 4'b0111時，就可以偵測按鍵F、E、D、C




# Keypad(3/8)

## ■ Example

- 當Row Line = 4'b0111時，就可以偵測按鍵F、E、D、C
- 如果Column = 4'b1111時，則代表按鍵F、E、D、C都沒有被按下去

Row Line  
=4'b0111



F	E	D	C
B	3	6	9
A	2	5	6
0	1	4	7




Column Line

# Keypad(4/8)

## ■ Example

- 當Row Line = 4'b0111時，就可以偵測按鍵F、E、D、C
- 如果Column = 4'b1111時，則代表按鍵F、E、D、C都沒有被按下去
- 如果Column = 4'b0111時，則代表按鍵1被按下去

Row Line  
=4'b0111



F	E	D	C
B	3	6	9
A	2	5	6
0	1	4	7



Column Line




# Keypad(5/8)

## ■ Example

- 當Row Line = 4'b0111時，就可以偵測按鍵F、E、D、C
- 如果Column = 4'b1111時，則代表按鍵F、E、D、C都沒有被按下去
- 如果Column = 4'b0111時，則代表按鍵1被按下去
- 如果Row Line = 4'b1011，就可以偵測按鍵B、3、6、9

Row Line  
=4'b1011



F	E	D	C
B	3	6	9
A	2	5	6
0	1	4	7




Column Line

# Keypad(6/8)

## ■ Example

- 當Row Line = 4'b0111時，就可以偵測按鍵F、E、D、C
- 如果Column = 4'b1111時，則代表按鍵F、E、D、C都沒有被按下去
- 如果Column = 4'b0111時，則代表按鍵1被按下去
- 如果Row Line = 4'b1011，就可以偵測按鍵B、3、6、9
- 如果Column = 4'b1011時，則代表按鍵3被按下去

Row Line  
=4'b1011



F	E	D	C
B	3	6	9
A	2	5	6
0	1	4	7



Column Line

# Keypad(7/8)

- 如果希望一直偵測每一排是否有按鍵被按下，則需要藉由快速切換來達到目的
- 每次偵測一排要給多少個clock cycle？
  - 如果太少(1 clock cycle)，則掃描速度太快，沒有debounce，可能會使結果不如預期輸入
  - 如果太多(50M clock cycle)，則掃描速度太慢，debounce過長，等待時間太久
- 因此需要一個除頻器(clock divisor)，產生一個比較適當的速度，來控制keypad
- 對每一個FPGA，適當的速度都不一樣，因此要個別測試(此次使用 100 Hz 即可)

# Keypad(8/8)

```
module checkkeypad(clk, rst, keypadRow, keypadCol);
    input clk, rst;
    input [3:0]keypadCol;
    output [3:0]keypadRow;

    reg [3:0]keypadRow;
    reg [3:0]keypadBuf;
    reg [31:0]keypadDelay;

    always@(posedge clk)
    begin
        if(!rst)
        begin
            keypadRow <= 4'b1110;
            keypadBuf <= 4'b0000;
            keypadDelay <= 31'd0;
        end
        else
        begin
            if(keypadDelay == `TimeExpire_KEY)
            begin
                keypadDelay = 31'd0;
                case({keypadRow, keypadCol})
                    8'b1110_1110 : keypadBuf <= 4'h7;
                    8'b1110_1101 : keypadBuf <= 4'h4;
                    8'b1110_1011 : keypadBuf <= 4'h1;
                    8'b1110_0111 : keypadBuf <= 4'h0;
                    8'b1101_1110 : keypadBuf <= 4'h8;
                    8'b1101_1101 : keypadBuf <= 4'h5;
                    8'b1101_1011 : keypadBuf <= 4'h2;
                    8'b1101_0111 : keypadBuf <= 4'ha;
                    8'b1011_1110 : keypadBuf <= 4'h9;
                    8'b1011_1101 : keypadBuf <= 4'h6;
                    8'b1011_1011 : keypadBuf <= 4'h3;
                    8'b1011_0111 : keypadBuf <= 4'hb;
                    8'b0111_1110 : keypadBuf <= 4'hc;
                    8'b0111_1101 : keypadBuf <= 4'hd;
                    8'b0111_1011 : keypadBuf <= 4'he;
                    8'b0111_0111 : keypadBuf <= 4'hf;
                    default      : keypadBuf <= keypadBuf;
                endcase
                case(keypadRow)
                    4'b1110 : keypadRow <= 4'b1101;
                    4'b1101 : keypadRow <= 4'b1011;
                    4'b1011 : keypadRow <= 4'b0111;
                    4'b0111 : keypadRow <= 4'b1110;
                    default: keypadRow <= 4'b1110;
                endcase
            end
            else
                keypadDelay <= keypadDelay + 1'b1;
            end
        end
    end
endmodule
```

當keypadDelay = TimeExpire時

1.keypadDelay歸零

2.依照目前偵測的row，檢查是否該row是否有按鍵被按

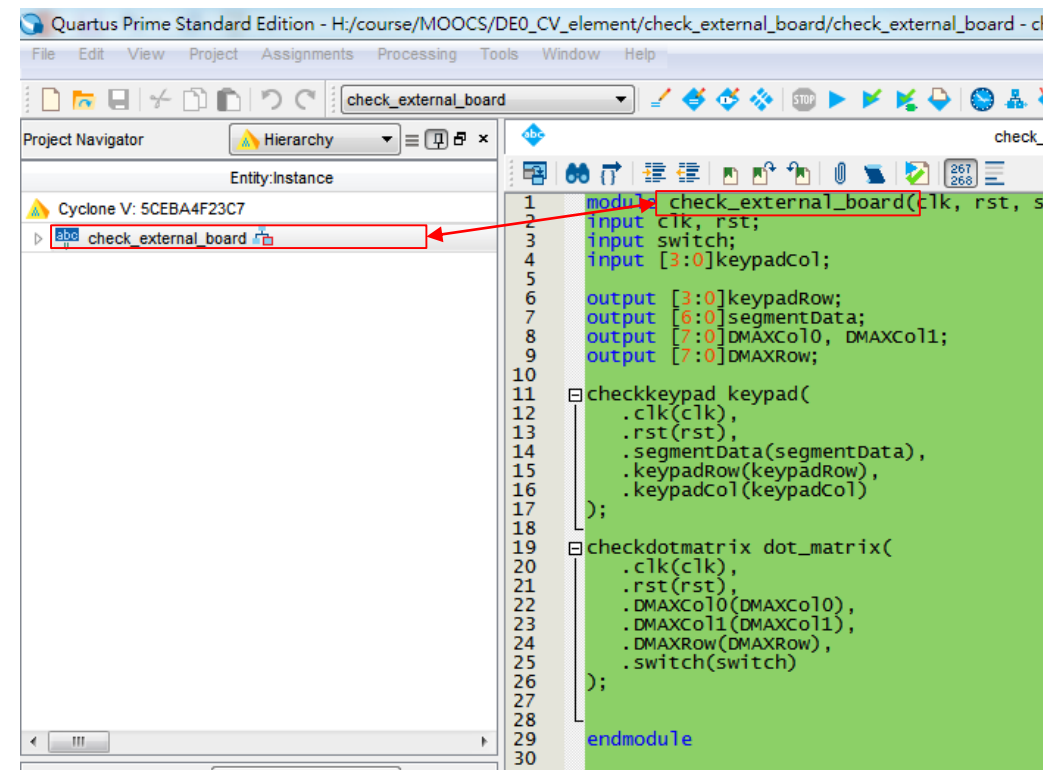
3.切換到下一列

# Notice

- wire and reg type define
  - `always begin ...裡面變數... end`，宣告 **reg** type
  - `always begin ..... end` 外面變數，宣告 **wire** type
    - 需搭配 **assign** 使用
- `reg == register`
  - 在**組合**電路中使用 `reg type`，合成 → **線 (net)**
  - 在**循序**電路中使用 `reg type`，合成 → **Flip-flop (register)**
- Inferred latch
  - 在組合電路中，`case`、`if...else...`若**沒有寫滿**，合成後會產生latch

# Notice

- 請勿命名中文資料夾或數字開頭資料夾
- 請確認 Device family 是否與 FPGA 晶片符合
  - Family: **Cyclone** / Device: **5CEBA4F23C7**
- top module name & project name 需要一致
- 燒錄檔案至 FPGA 前，Double-check **Pin Assignment**
  - 設定錯誤的 Pin，會導致 FPGA 無法正確執行
- 連接 FPGA 板後，請先確認是否可以正常燒錄與動作
  - **USB Blaster**，指定到 USB Blaster Driver 目標資料夾 C:\altera\16.0\quartus\drivers\usb-blaster



# Number Representation

- May be represented using
  - Binary, decimal, hexadecimal,
- Format
  - `<size>'<base_format> <number>`
  - `base_format`:
  - `b, d, h,`
- Example
  - `4'b1111; 16'd255`
  - `23456` (32-bit decimal # by default); `'hc3` (32 bit)
  - `12'b1111_0000_1010`