

Final Project Report

Group members:

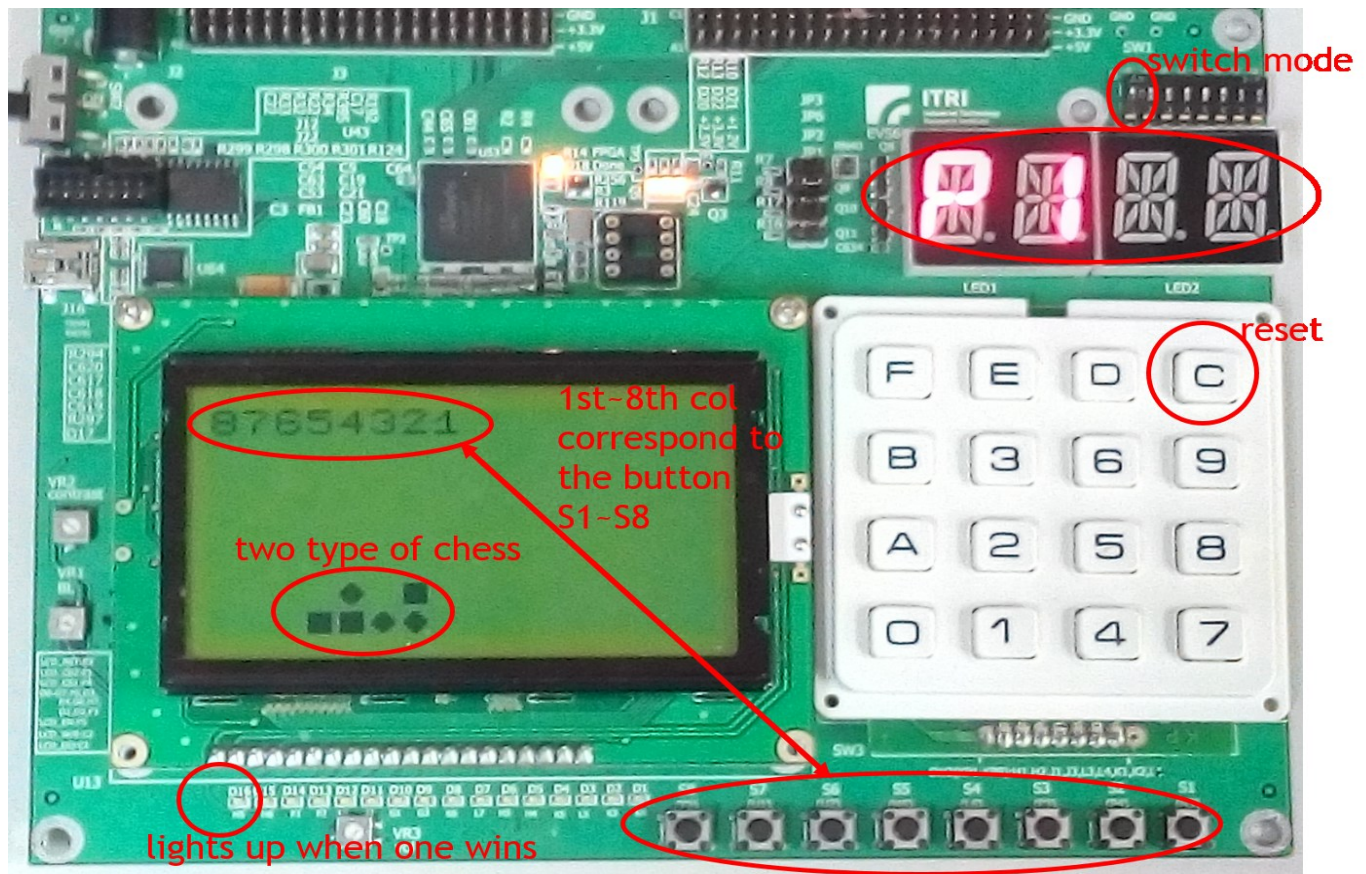
1. Department of Math. 101021120 鄭宇翔

2. Department of Math. 101021110 林力暉

Name of our design: LCD 四子棋 (LCD Connect Four)

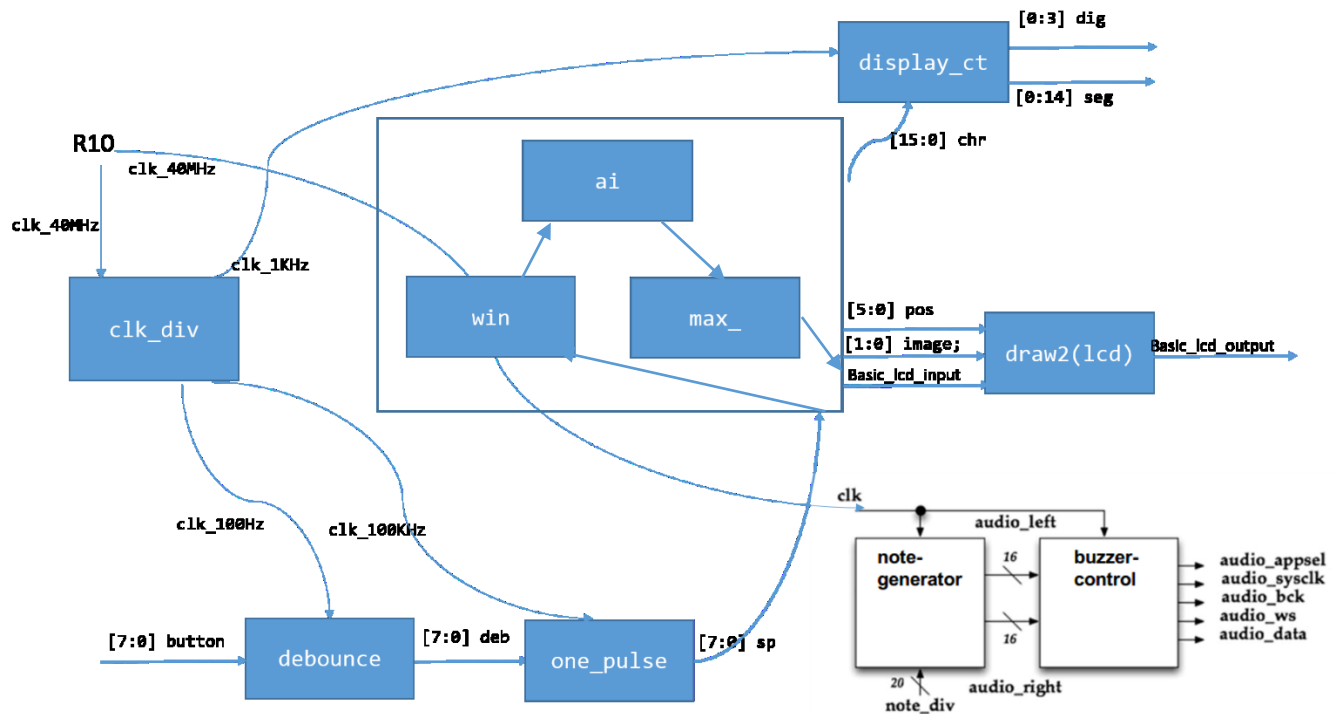
Part I Specification

- Two modes (controlled by the switch T1)
 - 1-player mode (玩家(P1) vs. 電腦(AI))
(T1 = 0, switch up)
 - 2-player mode (玩家(P1) vs. 玩家(P2))
(T1 = 1, switch down)
- 共 7 * 8 (row * col) 格 (on LCD)
- 14-seg LED displays whose turn and the winning message.
- Press the key C on the keypad to reset the game any time.
- Click one of the buttons S1~S8 to play (toss) chess.
- Play sound as you toss your chess and as you win.
- LED D16 lights up when one wins.



Part II *Implementation details (explanation with block diagram, state transition diagram, etc.)*

Block diagram



win block:

function:

1. count the num. of conti. chess.
2. Judge whether someone wins.

ai block:

function: 計算每個位置的 score

input: 上下、左右、斜線的 win block output.

output: The “score” of every position.

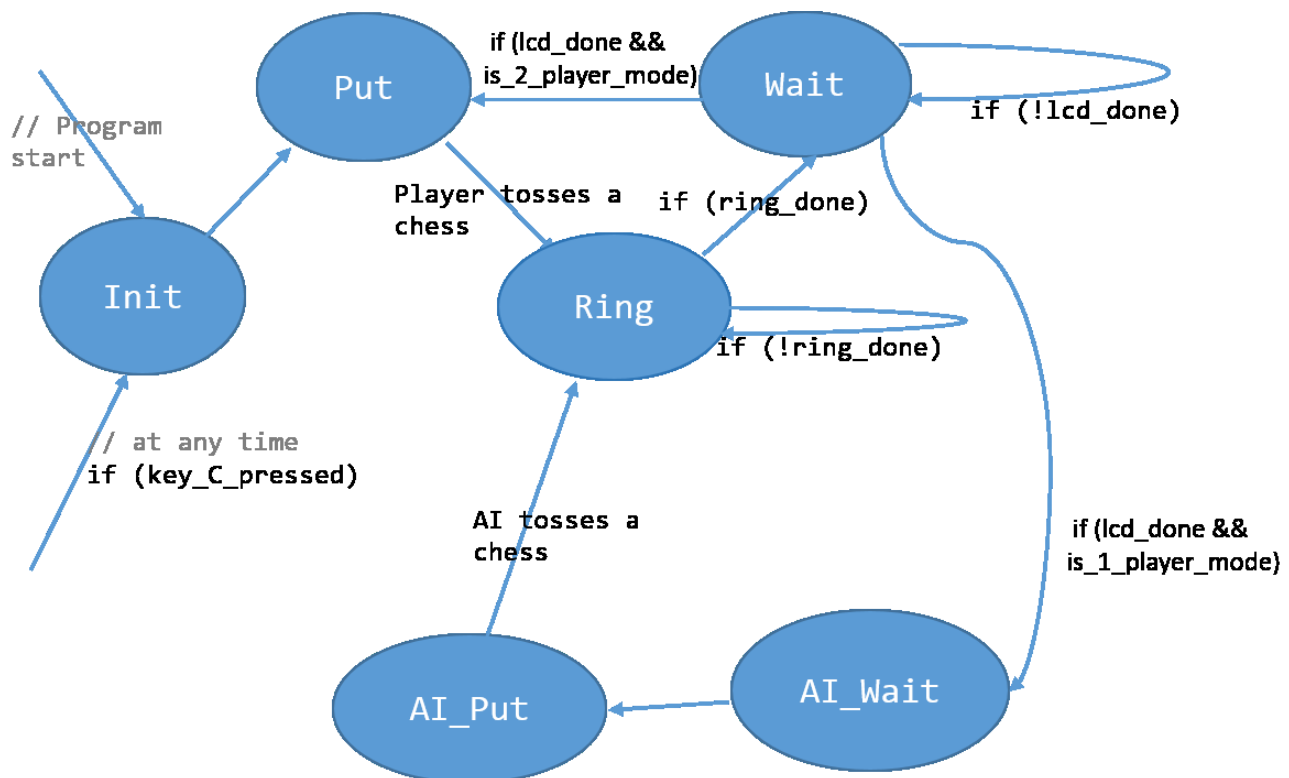
max_ block:

function: 計算最高的 score (AI 就下那步)

input: output of ai block (score).

output: the maximum score of all and its position.

State transition diagram



The function of win.v (in module win.v)

$\phi : 3\text{bits}(\text{連續片段}) \rightarrow 2\text{bits}(\text{連線棋子個數})$

0xx \mapsto 00

10x \mapsto 01

110 \mapsto 10

111 \mapsto 11

How to implement the above function:

transform to K-map

	0	1	
0	1	0	00
0	1	0	01
0	1	0	10
0	0	1	11

1. 拆成左右 part

左：

	0	1
0	1	00
0	1	01
0	1	10
0	0	11

右：

	0	1
0	1	00
0	1	01
0	1	10
0	0	11

3. transform to code

```
module win(a,c);
input [2:0]a;
output [2:0]c;
wire w1,w2;
and (w1,a[0],a[2]) ,(w2,a[0],~a[1]) ,(c[1],a[0],a[1]);
or (c[0],w1,w2);
assign c[2]=1'b0;
endmodule
```

The AI algorithm (in module ai.v)

原理：給每個位置一個分數，取最高分數的位置為 AI 要下的位置。

給分方式：

1. 棋子到最上方為 0 分
2. 其餘每個位置至少 2 分
3. AI 一下下去就會贏的位置(附近已有 3 珠連線)，分數 += 2^{14}
4. 玩家(P1)一下下去就會贏的位置(附近已有 3 珠連線)，分數 += 2^{13}
5. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若超過 7 個，分數 += 2^{10}
6. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 7 個，分數 += 2^9
7. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 6 個，分數 += 2^8
8. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 5 個，分數 += 2^7

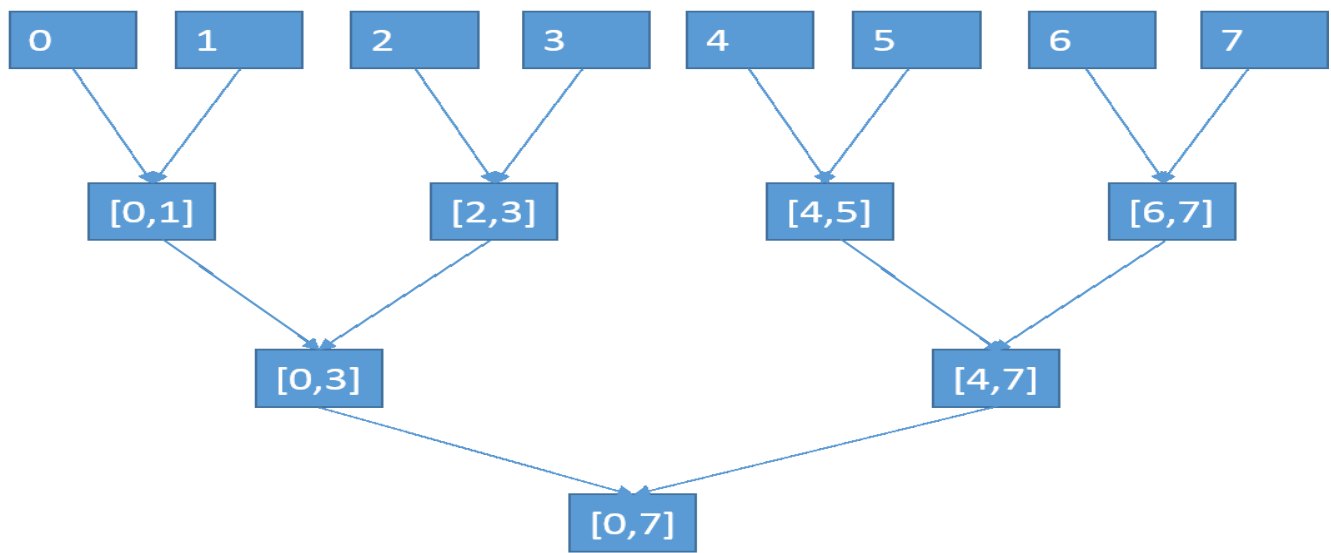
9. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 4 個，分數 += 2^6
10. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 3 個，分數 += 2^5
11. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 2 個，分數 += 2^4
12. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 1 個，分數 += 2^3
13. 在該處畫一個「米」字，遇到空白或對方棋子就停止該方向計算，計算「AI 連續棋子總個數 + 玩家連續棋子總個數」，若等於 0 個，分數 += 2^2

In ai.v:

```
module
ai(bottom,in11,in21,in31,in41,in12,in22,in32,in42,in13,in23,in33,in43,score);
input [2:0]bottom;
input [3:0]in11,in21,in31,in41,in12,in22,in32,in42,in13,in23,in33,in43;
output [15:0]score;
assign score[14]=bottom?(in11>=3||in21>=3||in31>=3||in41>=3):0;
assign score[13]=bottom?(in12>=3||in22>=3||in32>=3||in42>=3):0;
assign score[10]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42>=8):0;
assign score[9]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==7):0;
assign score[8]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==6):0;
assign score[7]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==5):0;
assign score[6]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==4):0;
assign score[5]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==3):0;
assign score[4]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==2):0;
assign score[3]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==1):0;
assign score[2]=bottom?(in11+in21+in31+in41+in12+in22+in32+in42==0):0;
assign score[1]=bottom?1:0;
endmodule
```

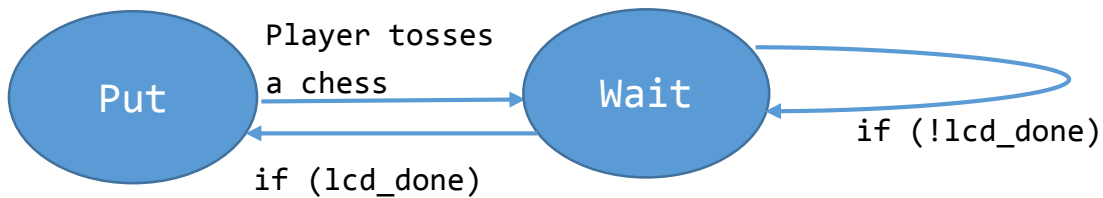
How to count max score (in module top.v and max .v)

原理：兩兩比較位置[0,1],[2,3],[4,5],[6,7]的 score 高低，取勝者為 x01, x23, x45, x67，再兩兩比較[x01,x23],[x45,x67]的 score 高低，取勝者為 x03,x47，再兩兩比較[x03,x47]的 score 高低，取最後勝利者為 x07。



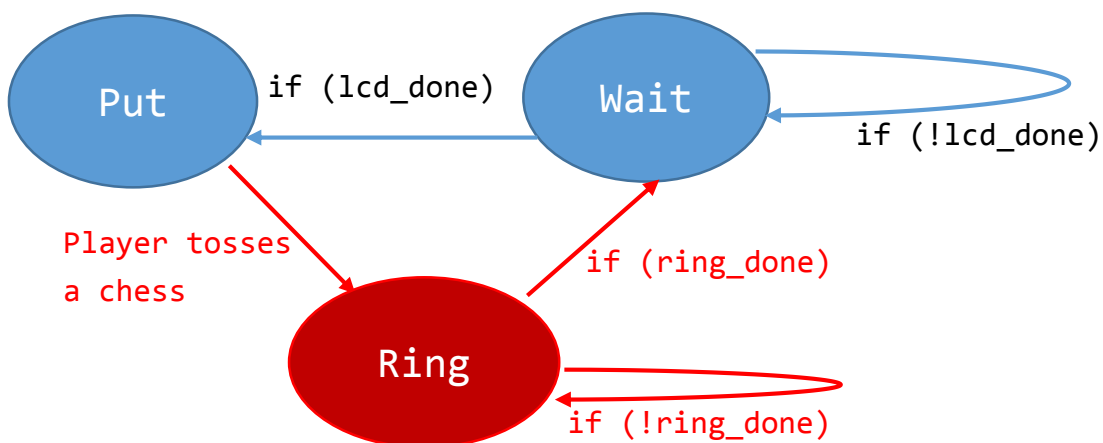
Part III *Issues and problems worthy of note and how you solved them*

1. 問題： Want to play sound as you toss a chess, but the state transition of playing chess was like following:



Nowhere to play sound.

解決方式：I solved it by adding an intermediate state “Ring” like following:



2. 問題：寫 AI 時遇到 LUT 不夠。
解決方式：把 AI 寫簡單一點(很多)。
3. 問題：不知道按鍵對應到 LCD 上確切的位置，有時會看錯。
解決方式：第一 row 加上數字 1~8。

Part IV A list of reference materials or manual
that you read (if any)

1. Wikipedia
2. Lecture notes
3. Sample codes from TAs
4. Our talent

Part V Suggestions for this class (if any)

1. Labs are a little too easy.(The part of provided code is too much)
2. Should give more state construction and transition lab practices.
3. Should teach us the coding convention earlier.
4. Should give the complete code after every lab submission.
5. Can teach us how to connect two boards and how to make them communicate to each other.
6. Can teach us how to connect own peripheral devices.
7. Can teach us how to download code to ROM rather than RAM.
8. Can teach us how to adjust the volume and the bright and contrast of the LCD.