

# 遊戲：十點半

---

## 目錄

- [遊戲介紹](#)
- [Lab介紹](#)
- [七段顯示器](#)
- [LED](#)
- [檢查答案](#)
- [評分標準](#)
- [參考資料](#)

## 需要使用工具

- 除頻器
- FSM
- LUT(會事先提供，作為抽撲克牌用)
- 3顆LED燈
- 兩顆七段顯示器

## 遊戲介紹:

十點半是撲克遊戲的一種。經常被用來賭博。遊戲分為莊家與玩家，玩家目標為拿到總點數合大於莊家拿到總點數合，反之則由莊家獲勝，點數均以不超過十點半為原則

本Lab總共兩人玩，分別為玩家與莊家，點數大者即獲勝。

起始玩家會拿到一張牌，拿到的牌會從1到13不等，其中1~10即代表對應的數值

11、12、13則代表半點，**累積兩個半點可以進位成一點**

玩家可以自行選擇要補牌與否，同理莊家，補牌上限為**五張**(包含起始的一張牌)

## 遊戲決定勝負方式

舉例: 玩家起始牌: 3、莊家起始牌: 9

<範例1> 玩家補牌補到7，表示目前累積點數10，此時莊家如果不補牌，則由玩家獲勝

<範例2> 玩家補牌補到5，表示目前累積點數8，此時莊家如果不補牌，則由莊家獲勝

<範例3> 玩家補牌補到6，表示目前累積點數9，此時莊家如果不補牌，則由莊家獲勝(平手算莊家獲勝)

<範例4> 玩家補牌補到6，表示目前累積點數9，此時莊家如果補到11，表示目前累積點數9.5，則由莊家獲勝

<範例5> 玩家補牌補到6，表示目前累積點數9，此時莊家如果補到3，表示目前累積點數12，由於莊家點數超過10.5，判定玩家勝出

<範例6> 玩家補牌補到3，又在補到6，表示目前累積點數12。此時莊家不論補到多少，均算莊家勝出(在本lab莊家仍可補牌，但不論最終補多少，均判定莊家獲勝)

為了簡化問題，本lab不考慮過五關的問題

## 本Lab遊戲玩法

每回合從按下btn\_m開始遊戲，按下後會進入抽牌階段

抽牌階段:

由玩家先抽一張牌，接著由莊家抽1張牌，抽牌需要使用**LUT.v**

首先將此LUT instantiate進tenthty.v，當需要抽牌時，拉起pip訊號，此時**number**訊號會延遲一個cycle後送出，此**number**訊號即表示抽到的牌。完成後即進入補牌階段，由玩家先開始補牌

**重點提示：** 在此階段無須特別使用七段顯示器顯示補牌情況

補牌階段:

同樣使用LUT.v補牌，方式同抽牌階段介紹

按下btn\_m即表示補牌，如果玩家決定不補牌，則按下btn\_r表示換莊家補牌

若玩家決定補牌，一旦補牌超過十點半，則會自動換莊家，在未超過十點半的情況下，玩家最多可以補四張牌  
同理莊家補牌階段，按下btn\_m即表示補牌，按下btn\_r表示進入比較大小階段

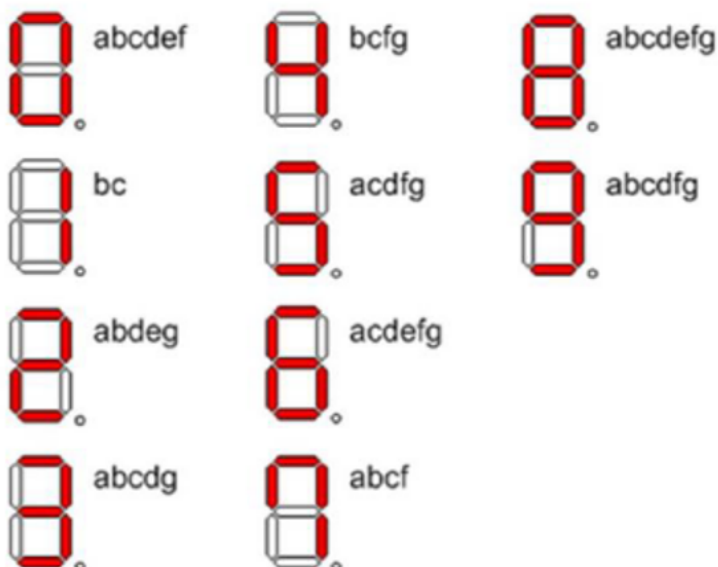
比較大小階段:

- 以不超過10.半為原則，比對莊家與玩家的點數大小，點數大者獲勝
- 在莊家與玩家同點的情況下，判定莊家獲勝
- 玩家超過10.半的情況下，判定莊家獲勝

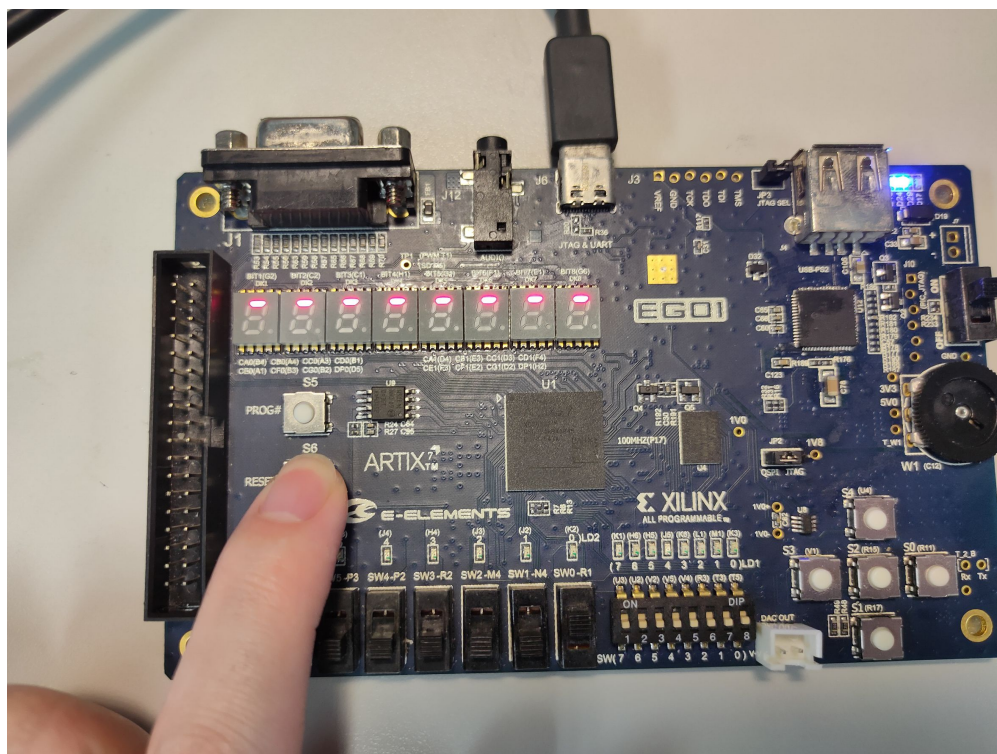
比大小完即為完成一個回合，按下btn\_r即可開始下個回合，同樣按下**btn\_m**才會開始遊戲  
遊戲一共進行**四個**回合，四個回合後狀態機須切換至DONE STATE

## 七段顯示器

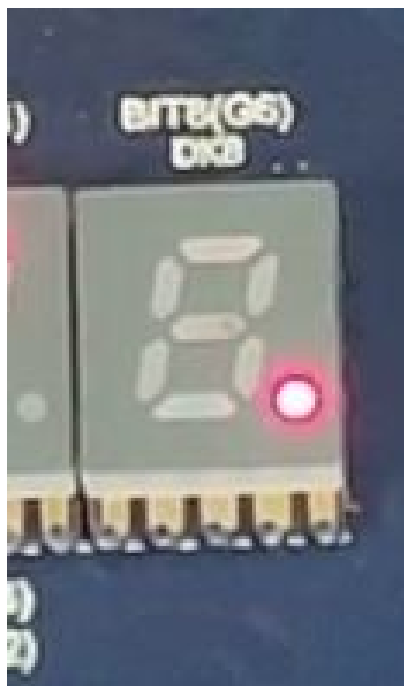
七段顯示器顯示數值如下：



七段顯示器Reset後則顯示如下：



若抽到11、12、13(半點的情形)，則顯示如下圖：

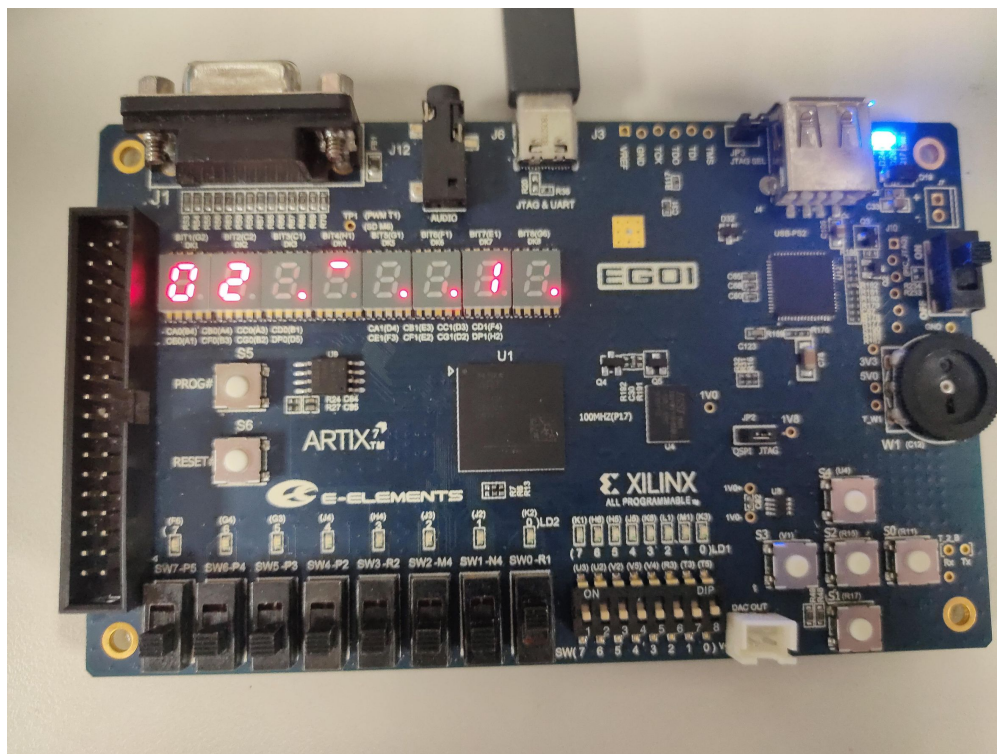


抽牌階段：

由於此階段佔時極短，無須使用七段顯示器顯示抽牌情況

### 補牌階段：

在玩家補牌階段，右邊5顆七段顯示器會由右往左顯示手牌資訊，最先出現的牌將顯示再最右邊，示意圖如下：

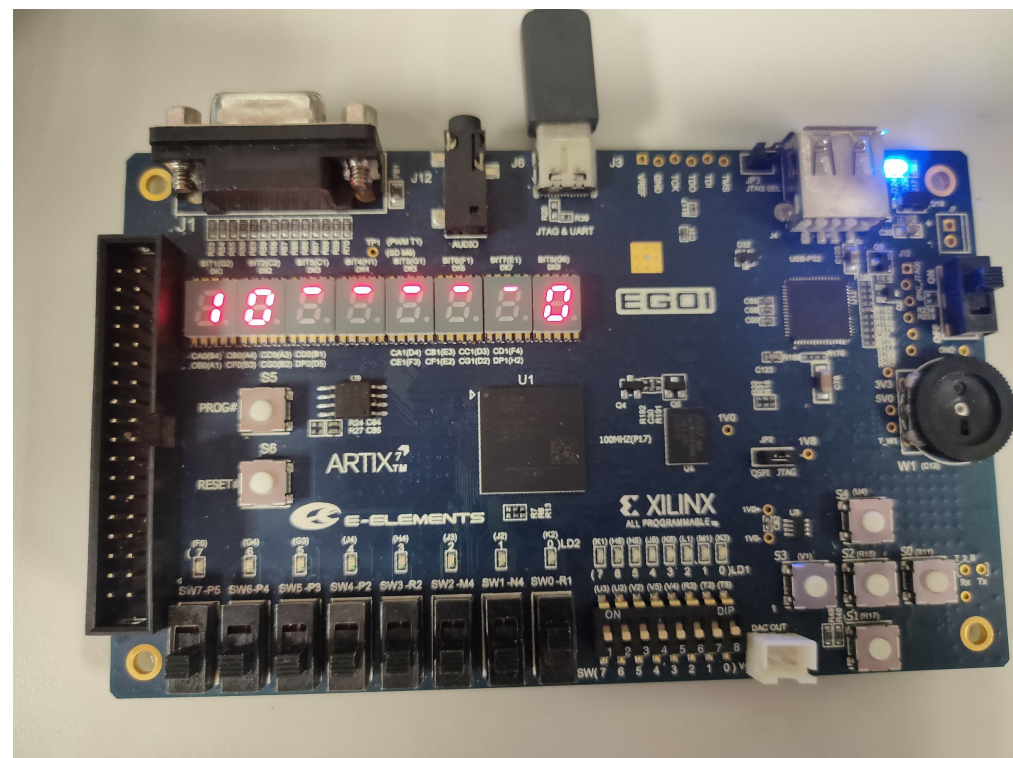


起始手牌為半點，接著補到1，再補到兩次半點，累加數值為2.5，總共補3張牌，總共四張牌

左邊三顆七段顯示器則顯示累加數值，如上圖左三顆七段顯示器。最左邊表十位數數值，中間表個位數，最右邊表示半點的情況，若無半點，則維持reset後情形，若有，則如上圖所示



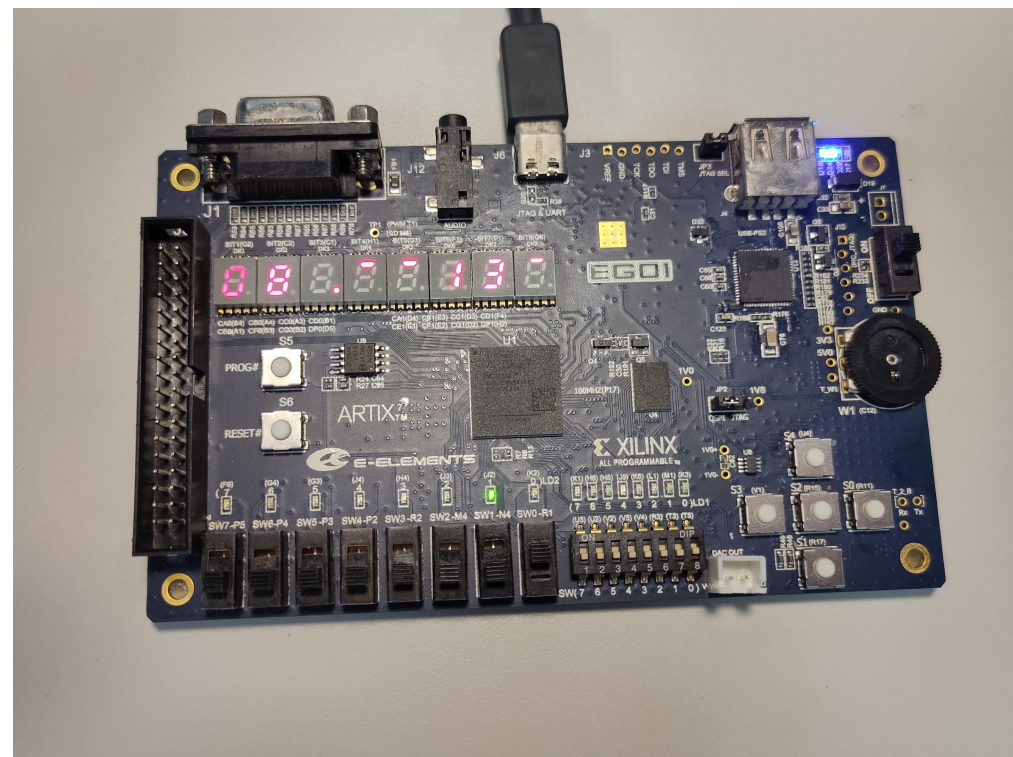
若遇到手牌出現10的情形，則直接顯示0即可 示意圖如下圖最右邊的七段顯示器：



同理在莊家補牌階段，顯示莊家手牌與累計數值

比較大小階段：

左三顆顯示莊家累積點數，右三顆顯示玩家累積點數(不論是否爆牌均需顯示)，示意圖如下：



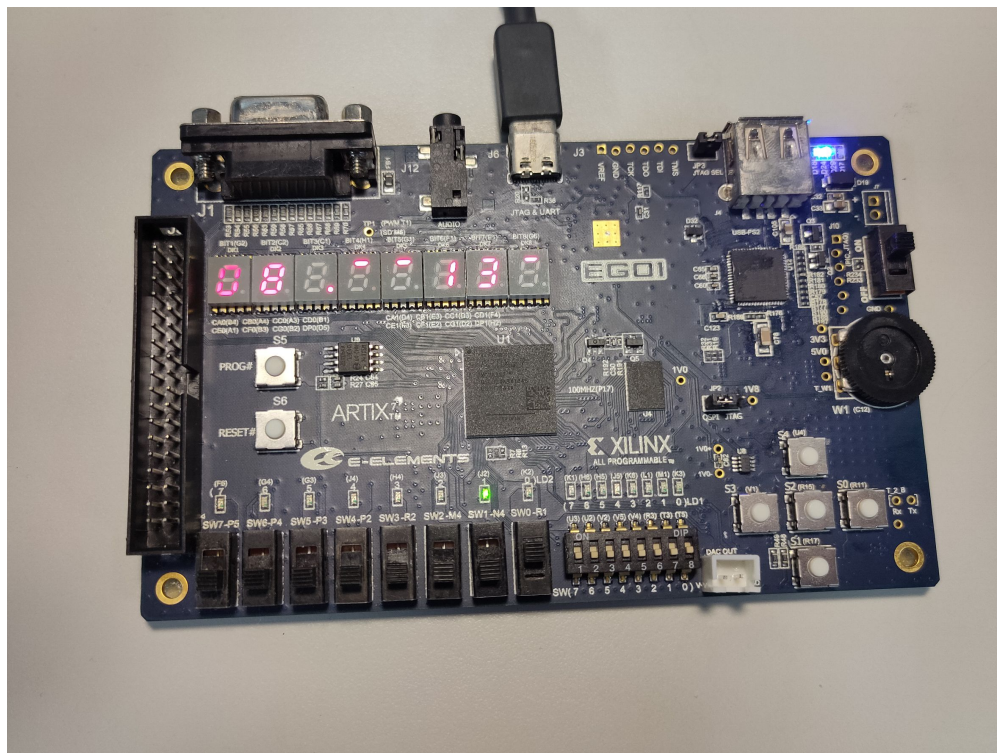
右三顆顯示方式與左三顆相同。最左邊表十位數數值，中間表個位數，最右邊表示半點的情況，若無半點，則維持reset後情形

## LED

比較大小階段：

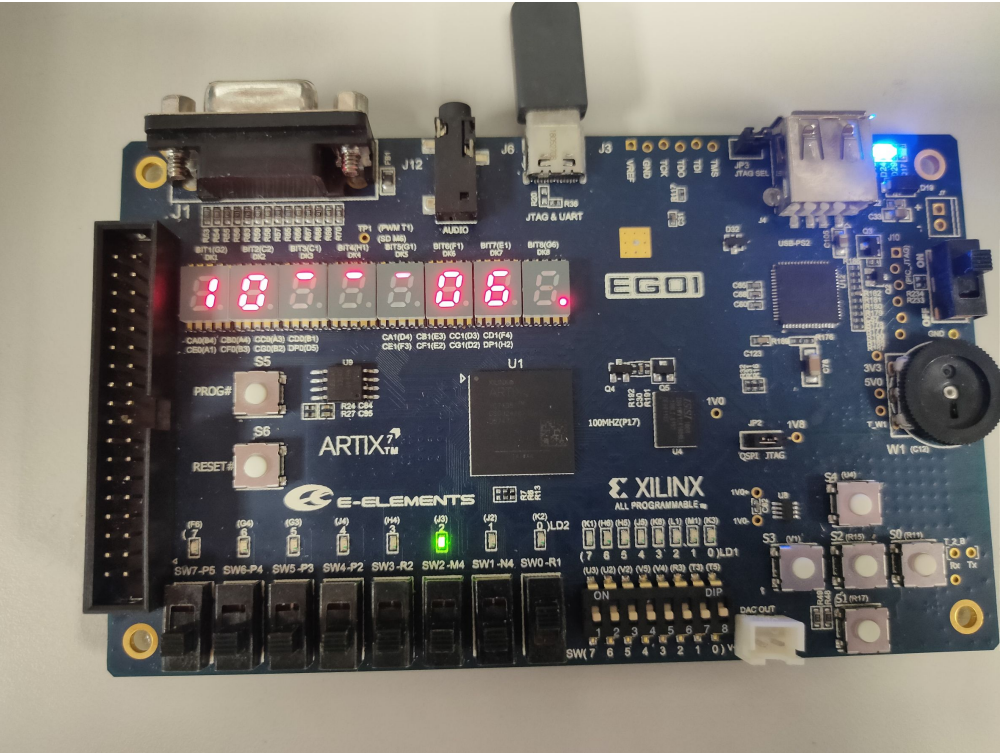
需要亮起LED燈表示完成，亮燈規則如下

led[0]：玩家贏；led[1]：莊家贏；(下圖左邊為莊家，右邊為玩家)



上圖由於玩家補牌超過十點半，因此判定莊家獲勝，led[1]亮起

led[2]則等四個回合均完成後拉起(即DONE STATE)，示意圖如下：



其餘時刻LED燈均不可亮起



## Data Config

- LUT(look up table)

Signal Name	I/O	Width	Simple Description
clk	I	1	Posedge triggered Clock
rst_n	I	1	Asynchronous negedge Reset
pip	I	1	Pip 訊號拉起時，會將 number 訊號延遲一個 cycle 後送出
number	O	4	表示撲克牌數值，數值從 1~13

- TenThirty

Signal Name	I/O	Width	Simple Description
clk	I	1	Posedge triggered Clock
rst_n	I	1	Asynchronous negedge Reset
btn_m	I	1	表示 S2 按鍵
btn_r	I	1	表示 S0 按鍵
seg7_sel	O	8	七段顯示器選擇器，控制八顆七段顯示器何時亮
Seg7	O	8	右邊四顆七段顯示器(DK5~DK8)
Seg7_l	O	8	左邊四顆七段顯示器(DK1~DK4)
led	O	3	LD2(0、1、2)

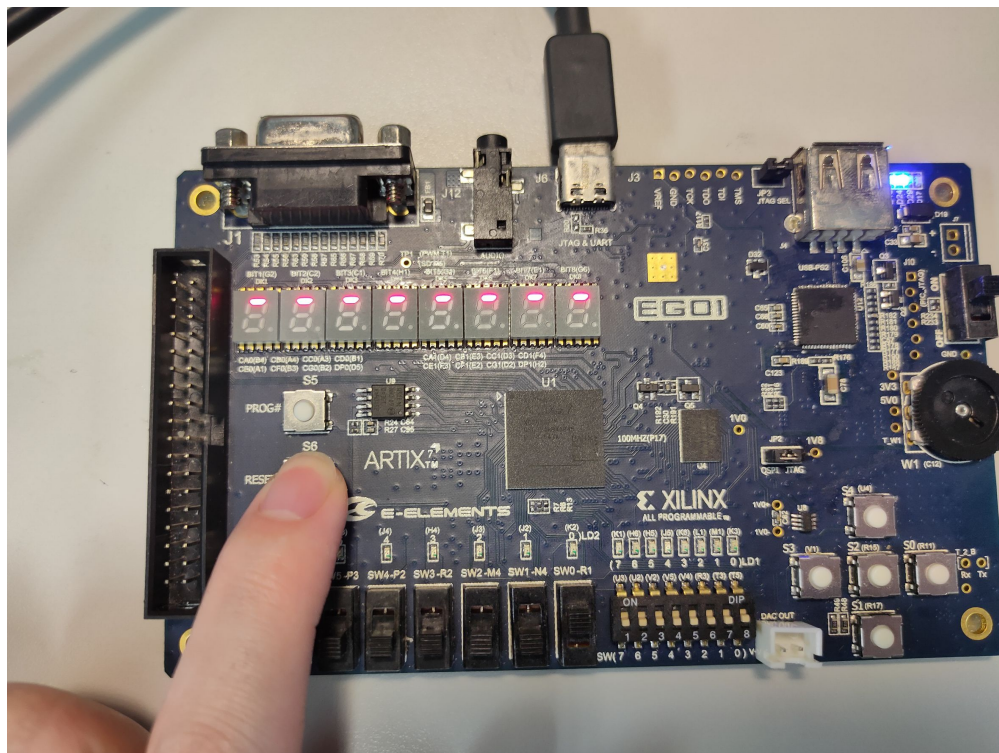
示範影片如下:

<https://www.youtube.com/watch?v=MHQ68WXCOEY&list=PLn0-Y9IYJqqvGrmoE9heed0lfZplog0h0&index=1>

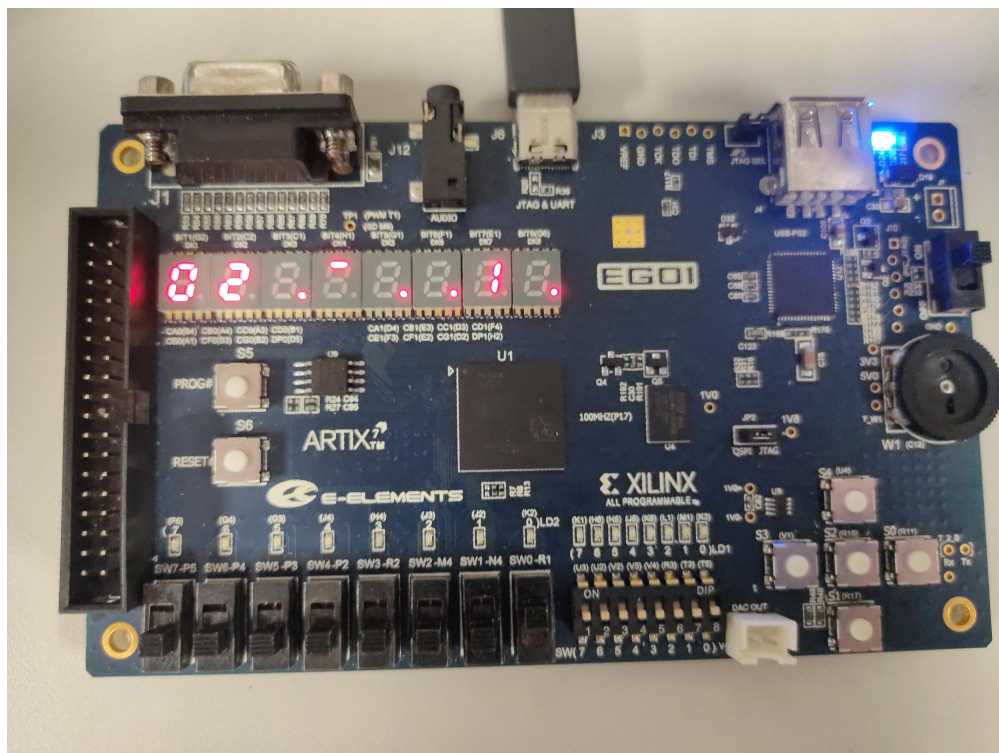


## 檢查答案

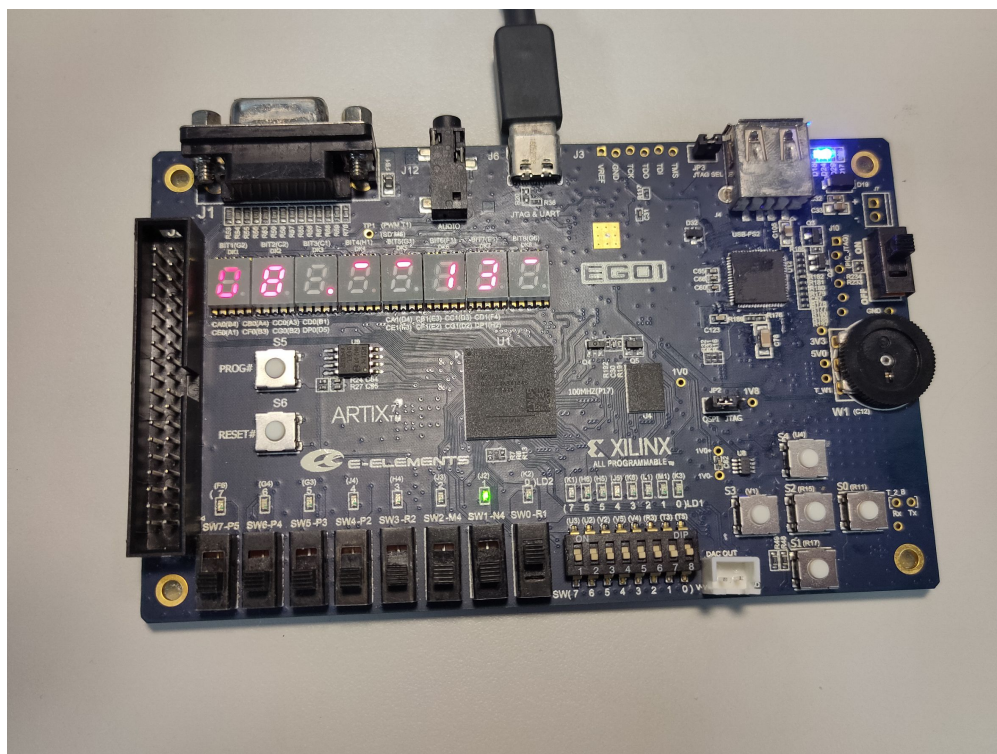
- Case 1 : Reset



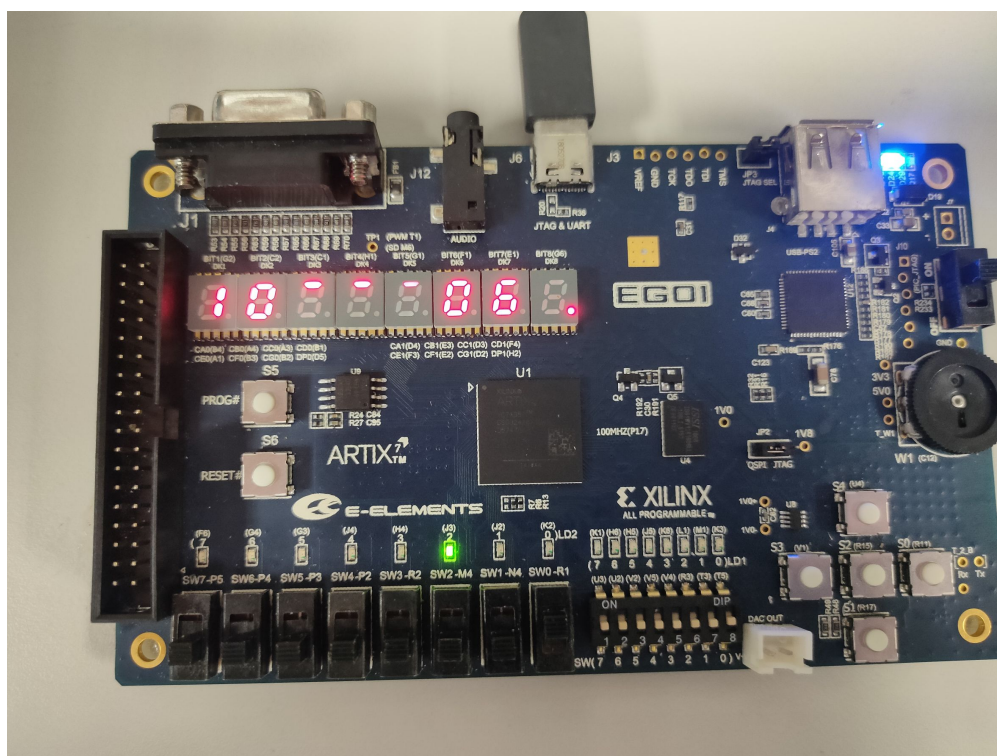
- Case 2 : 顯示手牌資訊與累積點數 舉例：起始牌:半點，補牌補到1、半點、半點



- Case 3 : 顯示結果(包含LED燈亮起與玩家、莊家累積點數) 範例:如圖



- Case 4 : 四回合後led[2]是否亮起 範例:如圖





## 評分標準

1. Reset後七段顯示器符合標準(10%)
2. btn\_m按下後完成抽牌階段，並成功顯示補牌階段的起始值(10%)
3. 第一回合玩家手牌與顯示正確，並按下btn\_r進入莊家補牌(10%)
4. 第一回合莊家手牌與顯示正確，並按下btn\_r進入比較大小階段(10%)
5. 第一回合比較階段led燈顯示正確(10%)
6. 玩家與莊家能成功補牌，且手牌與累積點數七段顯示器顯示正確(爆點要顯示爆點數值)(10%)
7. 補牌五張成功(10%)
8. 爆牌能自動切換下個狀態(莊家爆進入比較階段，玩家爆進入莊家補牌階段)(10%)
9. 當有爆牌情況，能成功判定獲勝者(10%)
10. 能完整進行四個回合，並進入DONE\_STATE(10%)

### 額外加分

有使用one\_shot\_pulse(10%)

總分：100分，最高110分

**\*\*本lab規定的IO以及LUT.v不可以改，違反者0分\*\***

**\*\*一定要使用FSM，違反者分數8折\*\***

**\*\*不可以更動原始.v檔，違反者分數8折\*\***

**\*\*不可以抄襲，違反者0分，抓到兩次者，請退選\*\***

**\*\*如果太多人做不出來，會依完成的程度適當給分，請大家盡力寫\*\***

## HINT

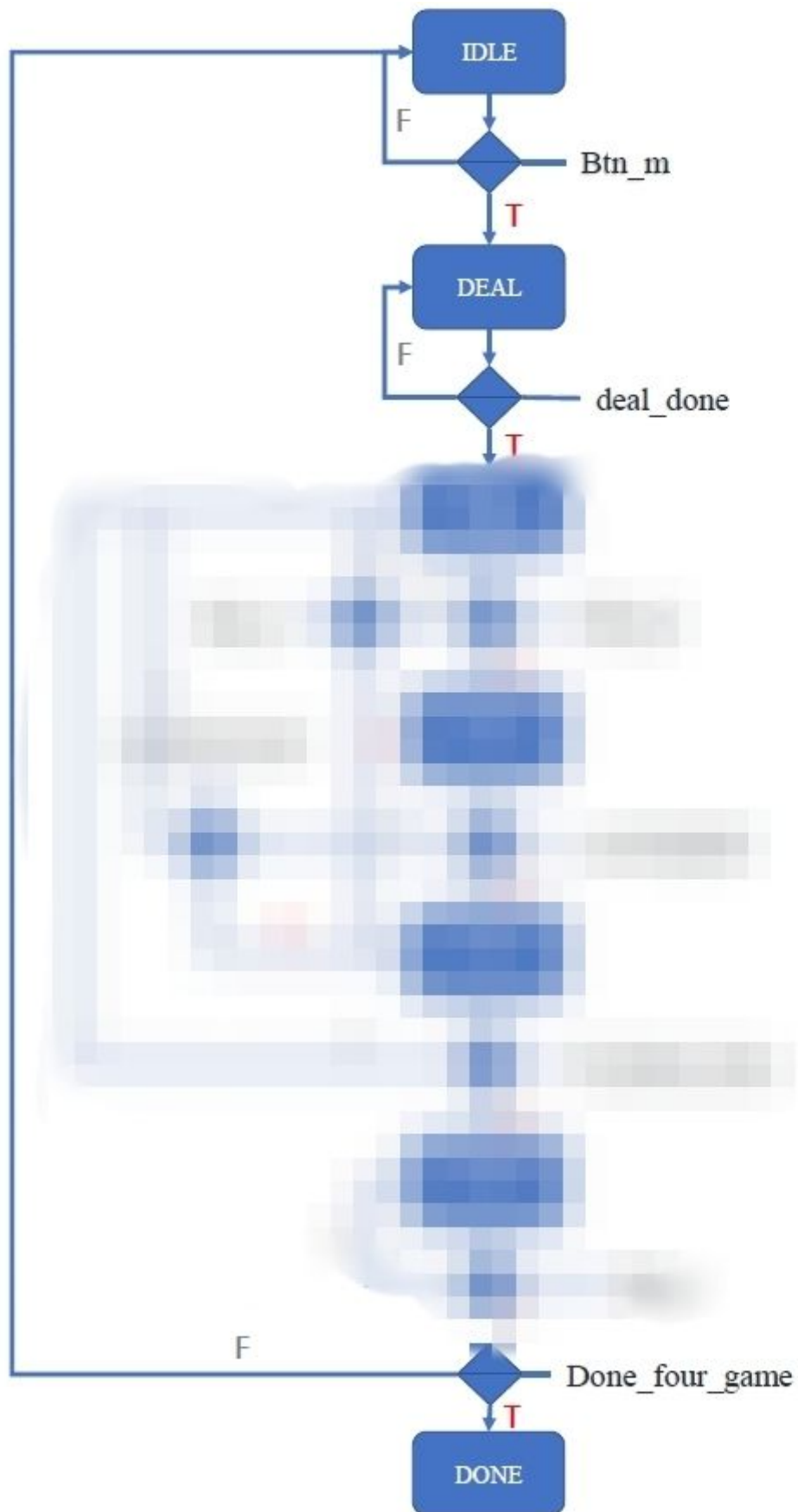
1. dis\_clk需要比d\_clk頻率更快
2. **seg7\_temp**暫存器是用來寫8顆七段顯示器
3. 你可能需要另外開暫存器存手牌資訊與累加點數(勿使用seg7\_temp)
4. 可以多花一個bit儲存半點資訊，兩次半點進位至一點即可

## 參考資料

[1] 十點半 維基百科 <br>  
<https://zh.wikipedia.org/wiki/%E5%8D%81%E9%BB%9E%E5%8D%8A>

[2] Markdown 語法大全<br>  
<https://ed521.github.io/2019/08/hexo-markdown/>

## 示範FSM





## TB example

```
initial begin
    rst_n = 1;
    set_initiaial;
    gap = $urandom_range(1,5);
    repeat(gap)@(negedge clk);
    rst_n = 0;
    repeat(gap)@(negedge clk);
    rst_n = 1;
    repeat(gap)@(negedge d_clk);
    btn_m = 1;
    repeat(2)@(negedge d_clk);
    btn_m = 0;
    repeat(2)@(negedge d_clk);
    btn_r = 1;
    repeat(2)@(negedge d_clk);
    btn_r = 0;
    btn_m = 1;
    repeat(1)@(negedge d_clk);
    btn_m = 0;
    repeat(2)@(negedge d_clk);
    btn_m = 1;
    repeat(6)@(negedge d_clk);
    $finish;
end
```

重點：在使用tb時，記得將tenthty.v的d\_clk改成跟d\_clk=counter[5]

tb需要自行設計，可以自行決定btn\_m跟btn\_r的拉起時間，助教這僅提供範例做參考

tb主要是拿來驗自己的FSM是否正確，無法驗七段顯示器

## LUT 示範

如下圖所示，在時脈正緣pip拉起，number訊號將在下一個cycle送出

