

Digital System Design (Fall 2021)

Final Report of Term Project

Group Number (組別): 第 2 組

Group Member 1 (組員 1): Student ID 0711239 Name 李勝維

Contribution (貢獻度) 50%

Group Member 2 (組員 2): Student ID 0711250 Name 吳奕霆

Contribution (貢獻度) 50%

Title (標題): 電子飛鏢機

A. Problem Description (問題敘述):

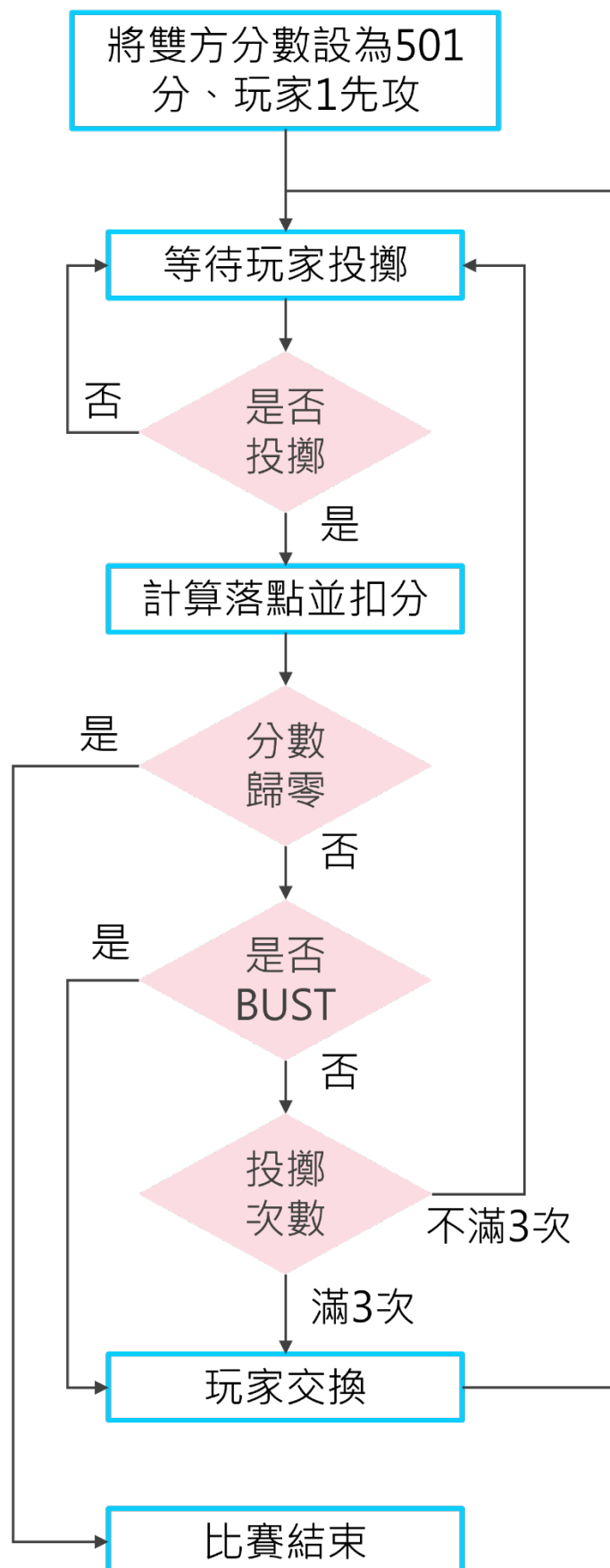
飛鏢是一種講求精準度的運動，近幾年來，隨著電子飛鏢靶興起，飛鏢成為了大家的娛樂休閒之一，在世界各地運動會也可以看到飛鏢的身影。而在本次期末專題，將試著利用 Verilog HDL 來模擬酒吧裡面常見的電子飛鏢機，並了解飛鏢機的運作原理。

這次專題，我們將挑選最受歡迎的「01 遊戲」來作為模擬的對象，01 遊戲中包含了 301、501、701 等，因為皆以 01 結尾故得名，玩法為扣分制，最先將得分剛好扣至 0 分者獲勝，若扣分超過目前的分數，則會「BUST」，並回到上一回合的分數，因此精準度將會是最重要的關鍵，本次專題將以 501 為例。

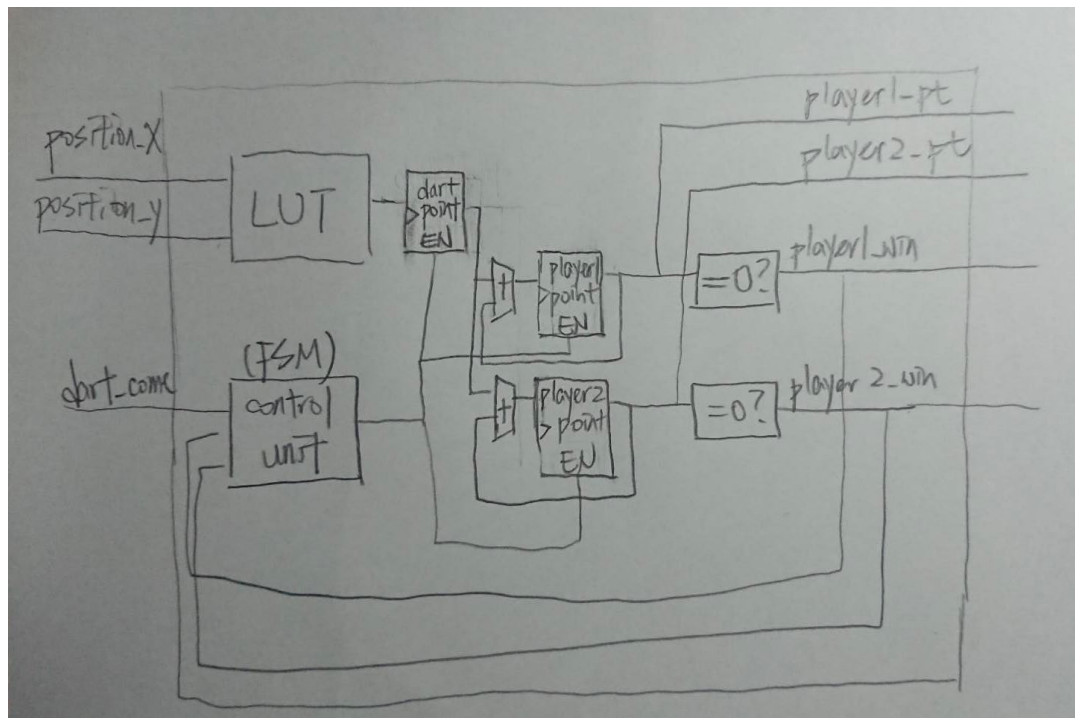
本次期末專題將著重於電子飛鏢機的三大功能:

1. 判斷飛鏢落點:判斷飛鏢落在幾分、幾倍區，以方便進行分數判讀
2. 計算分數:在確認飛鏢落點後，進行分數計算，更新兩邊玩家的分數
3. 判斷比賽是否結束，在更新完分數後，判斷比賽是否結束，或是判斷此次丟擲是否觸發「BUST」(ex:當剩下 3 分時，丟到 5 分區，因為超過 3 分，因此此次丟擲無效，下一回合將重新回到 3 分)

B. Flowchart or Procedure:



C. Block Diagram :



各模組功能：

- dart.v: 根據各項輸入，進行飛鏢比賽並輸出結果，為主要功能模組
- pattern.v: 內部存有不同之 testcase (比賽過程)，為測試之用
- test.v: 將 dart.v 和 pattern.v 進行串接，為 testbench

D. Definition of Inputs, Outputs, Control Signals, and Status Signals:

For module dart (dart.v):

Inputs:

- clk : system clk
- reset : system reset
- dart come : 飛鏢是否來了
- dart position x : 飛鏢在靶上之 x 座標
- dart position y : 飛鏢在靶上之 y 座標

Outputs:

- player 1 done: 判斷 player1 的回合是否結束
- player 2 done: 判斷 player2 的回合是否結束
- player 1 win: 判斷 player1 是否分數歸零
- player 2 win: 判斷 player2 是否分數歸零
- player 1 pt : player1 當前分數
- player 2 pt : player2 當前分數
- game set : 判斷比賽結束，即是否有人分數歸零

Control signals:

- dart point : 本次投擲分數
- counter : 計算目前玩家已投擲次數(丟三次才換人)
- who turn : 目前是哪個玩家的回合

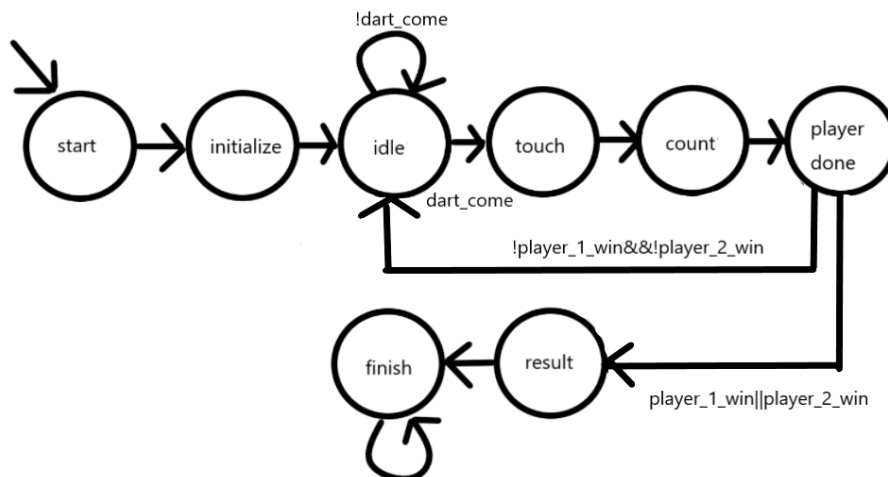
Status Signals:

- state : currnet state.
- next state : next state.

E. State Machine Chart (SM Chart) or State Graph:

各個 state 執行動作：

- Start：初始 state，比賽開始
- Initialize：初始化分數(將兩玩家分數設定為 501 分)
- Idle：等待飛鏢
- Touch：判斷飛鏢落點和此次丟擲分數
- Count：計算分數，並判斷丟擲是否有效(觸發 BUST)
- Player done：判斷比賽是否結束並換人
- Result：將比賽結果輸出
- Finish：結束比賽



F. Description of Verilog Code (Verilog 電路模組說明):

dart.v :

本次專題所實現的飛鏢機模組命名為 dart.v，細項說明如下:

1.流程:

飛鏢機運行模式大略如下:

- (1)將玩家分數初始化
- (2)等待飛鏢投擲
- (3)根據飛鏢落點判斷本次丟擲分數
- (4)判斷是否觸發 BUST/判斷遊戲是否結束/更新玩家分數
- (5)回到(2)繼續

2.判斷分數:

在本次專題中，使用了 LUT 來判斷分數，將飛鏢之 X、Y 整數座標輸入 LUT 內後，可查詢對應的落點分數和其是否為雙倍、三倍區。實作上我們使用了 $31 * 31$ 的 LUT，而 LUT 的內容產生則是透過撰寫 python 腳本 (generate mesh value.py)，配合測量實際的標靶尺寸並調整參數比例來產生。

3.判斷 BUST/判斷遊戲是否結束:

會先將玩家分數和該次丟擲分數做比較。若丟擲分數 > 玩家分數，代表觸發 BUST，玩家分數不更新。若丟擲分數 = 玩家分數，更新玩家分數為 0，代表遊戲結束，該玩家獲勝。若丟擲分數 < 玩家分數，則直接更新玩家分數。

G. Description of Test Bench (Verilog 測試模組說明):

1. test.v: 將 pattern.v 和 dart.v 各自實例化並接線，程式碼主要部分如下：

```
//submodule declaration
dart d1 (
    .game_set_o(game_set),
    .player_1_done_o(player_1_done),
    .player_2_done_o(player_2_done),
    .player_1_win_o(player_1_win),
    .player_2_win_o(player_2_win),
    .player_1_pt_o(player_1_pt),
    .player_2_pt_o(player_2_pt),
    .dart_come_i(dart_come),
    .dart_position_x_i(dart_position_x),
    .dart_position_y_i(dart_position_y),
    .clk(clk),
    .reset(reset)
);

pattern p1 (
    .dart_come_o(dart_come),
    .dart_position_x_o(dart_position_x),
    .dart_position_y_o(dart_position_y),
    .game_set_i(game_set),
    .player_1_done_i(player_1_done),
    .player_2_done_i(player_2_done),
    .player_1_win_i(player_1_win),
    .player_2_win_i(player_2_win),
    .player_1_pt_i(player_1_pt),
    .player_2_pt_i(player_2_pt),
    .clk(clk),
    .reset(reset)
);
```

2. pattern.v: 程式碼主要有兩部分，第一部分為使用 display, monitor 等函數將飛鏢遊戲進行的過程印出來，方便我們瞭解電路的運作，程式碼大致如下：

宣佈遊戲開始並及時更新分數：

```
#35$display("*****\n*Game Start!\n*****");  
$monitor("(%4d) Scoreboard [%d : %d]", $time,player_1_pt_i, player_2_pt_i);
```

等待遊戲結束後宣佈贏家：

```
initial begin  
    @(posedge game_set_i)  
    #100  
    $write("\n*****\n*");  
    if (player_1_win_i)  
        $write("Game set, player 1 won!");  
    else  
        $write("Game set, player 2 won!");  
    $display("*\n*****");  
    $finish;  
end
```

第二部分為模擬遊戲的進行，在將每次投擲的結果輸入進主要的電路模組進行運算，並透過 "@()" 的語法等待電路模組完成運算後，更新投擲位置，程式碼大致如下：

```
// player 1  
dart_position_x_o = 13;  
dart_position_y_o = 2;  
@(posedge player_1_done_i or posedge player_2_done_i)  
dart_position_x_o = 14;  
dart_position_y_o = 2;  
@(posedge player_1_done_i or posedge player_2_done_i)  
dart_position_x_o = 14;  
dart_position_y_o = 3;  
@(posedge player_1_done_i or posedge player_2_done_i)  
#10$display("player 1 done.");
```


遊戲進行的流程則如下表，流程產生方式則是實際進行一場遊戲後記錄：

	1	2	3	4	5	6	7	8	9	10	11
1		P1 score	P2 score	Player1 dart (score)	X	Y		Player2 dart (score)	X	Y	測試備註
2		501	501	NA	NA	NA		NA	NA	NA	
3	1	481	501	20	13	2					
4	2	461	501	20	14	2					
5	3	401	501	60	14	3					Test Triple
6	4	401	481					20	13	2	
7	5	401	445					36	24	4	Test Double
8	6	401	425					20	13	2	
9	7	341	425	60	14	3					
10	8	291	425	50	13	15					
11	9	271	425	20	13	2					
12	10	271	375					50	13	15	
13	11	271	325					50	14	16	
14	12	271	275					50	13	16	
15	13	211	275	60	14	3					
16	14	151	275	60	14	3					
17	15	131	275	20	13	2					
18	16	131	224					51	19	26	
19	17	131	164					60	14	3	
20	18	131	104					60	14	3	
21	19	91	104	40	13	1					
22	20	51	104	40	14	1					
23	21	34	104	17	18	23					
24	22	34	100					4	19	11	
25	23	34	50					50	14	15	
26	24	34	47					3	15	21	
27	25	-17	47	51	20	26					Player 1 BUST
28	26	34	7					40	15	0	
29	27	34	-7					14	8	28	Player 2 BUST
30	28	0	47	34	20	29					Game Set

測試中我們主要測試了 1. 單倍區分數計算、2. 兩倍區分數計算、3. 三倍區分數計算、4. 雙方玩家皆觸發 BUST 和 5. 判斷遊戲結束。

H. Simulation Results (模擬結果):

編譯並執行 test.v 即可得到模擬結果，結果如下：

```
VCD info: dumpfile pattern.vcd opened for output.
*****
*Game Start!*
*****
( 135) Scoreboard [501 : 501]
( 190) Scoreboard [481 : 501]
( 270) Scoreboard [461 : 501]
( 350) Scoreboard [401 : 501]
player 1 done.
( 430) Scoreboard [401 : 481]
( 510) Scoreboard [401 : 445]
( 590) Scoreboard [401 : 425]
player 2 done.
( 670) Scoreboard [341 : 425]
( 750) Scoreboard [291 : 425]
( 830) Scoreboard [271 : 425]
player 1 done.
( 910) Scoreboard [271 : 375]
( 990) Scoreboard [271 : 325]
(1070) Scoreboard [271 : 275]
player 2 done.
(1150) Scoreboard [211 : 275]
(1230) Scoreboard [151 : 275]
(1310) Scoreboard [131 : 275]
player 1 done.
(1390) Scoreboard [131 : 224]
(1470) Scoreboard [131 : 164]
(1550) Scoreboard [131 : 104]
player 2 done.
(1630) Scoreboard [ 91 : 104]
(1710) Scoreboard [ 51 : 104]
(1790) Scoreboard [ 34 : 104]
player 1 done.
(1870) Scoreboard [ 34 : 100]
(1950) Scoreboard [ 34 : 50]
(2030) Scoreboard [ 34 : 47]
player 2 done.
player 1 done.
(2190) Scoreboard [ 34 : 7]
player 2 done.
(2350) Scoreboard [ 0 : 7]
player 1 done.

*****
*Game set, player 1 won!*
*****
./pattern.v:64: $finish called at 2450000 (1ps)
```

可以觀察到：

兩位玩家的分數皆和測試流程時假設的相同(小括號內為模擬時間，中括號左側為 P1 分數，右側為 P2)

且在最後幾回合的分數更新皆正常，且觸發 BUST 時我們的主要電路模組功能正常，其中未顯示分數的原因為，玩家觸發 BUST 時分數不更新，因此未觸發 monitor 函數進行顯示。

I. Conclusions and Discussions (心得、感想、結論、及討論)

這次的專題將組員的興趣和課程所學進行結合，利用 Verilog-HDL，現實生活中的電子飛鏢機，從 0 開始到設計出正常運作的電路，非常的有成就感。雖然實作過程中遇到非常多的困難處，所幸最後皆一一克服。

而這次專題也運用到上課新學到的語法，在建立 LUT 時，運用 generate 語法，將數值利用迴圈 assign 給 LUT，非常方便也避免造成 code size 太大。

References (參考資料)：

- [1] [鏢靶上的分數 | 飛鏢新手指南 \(dartslive.com\)](#)
- [2] [01 遊戲規則說明 | 飛鏢新手指南 \(dartslive.com\)](#)