

邏輯設計期末報告

報告題目： 刷頻機器人

學生：林耕澤











學號：E94084032

授課教授姓名：賴瑾峰

專題構想：

研究動機：

我相信有在看實況的人都對直播平台圖奇(TWITCH)不陌生。在圖奇平台上，各式各樣的實況台及實況主，百家爭鳴，其中圖奇第一奶台史丹利(relaxing234)從中脫穎而出、一枝獨秀、引領風騷，每次開台平均都有一萬多個觀眾。不過其異常的聊天室，引來了需多質疑聲浪，也因此"史丹利買機器人充實況人氣"的話題在各大論壇討論得相當熱烈，雖然在後來圖奇狂刪百萬機器人之後還給了史丹利一個清白，不過此事件也啟發了我做刷頻機器人的想法。

	21	萌萌小蝶 (butterflyouo)	100,308 hours	32 hours	5,519	3,134	304,835	-3,092	302,491	Partnered	Mature	Chinese
	22	嬌兔 (zrush)	165,885 hours	44 hours	5,216	3,748	201,414	-3,068	284,211	Partnered	Mature	Chinese
	23	NeVeR_LosEs	120,338 hours	28 hours	8,784	4,297	222,601	-3,035	288,291	Partnered		Chinese
	24	二十五 (nomatterhow670)	107 hours	38 hours	7	2	2,622	-2,700	724			Chinese
	25	工商巨賈 (sixwong)	35,351 hours	19 hours	3,194	1,789	278,511	-2,578	104,703		Mature	Chinese
	26	花枝妙麗 (cooldog0725)	42,677 hours	65 hours	1,038	649	12,063	-2,360	8,130	Partnered	Mature	Chinese
	27	relaxing234	235,623 hours	26 hours	11,686	8,891	319,001	-2,310	356,046	Partnered		Chinese
	28	benjackergames	0 hours	0 hours	2	1	119	-1,978	15		Mature	Chinese
	29	噴馬很忙 (63598742)	2,603 hours	29 hours	123	88	2,889	-1,905	1,971		Mature	Chinese
	30	可愛小麥麥 (cute_mai)	284 hours	20 hours	31	13	4,263	-1,770	1,930		Mature	Chinese

證明史丹利為清白圖片支援(<https://www.ttshow.tw/article/79214>)

研究目的:

了解製作此專題報告所應用到的相關概念(此專題使用到編碼器、正反器)及在 Verilog 上如何實現。

預期目標：

希望能製作出一個當實況主有激烈反應(有好的表現等...)時，能自動刷頻，且可轉換刷頻內容的機器人。

實作方式：

第一部分:宣告

```
module Dff(clk,niceplay,rst,climax,count,contents);  
    output climax;  
    output reg [7:0] count;  
    output reg[2:0] contents;  
    input clk,niceplay,rst;  
    reg climax;  
    wire o;
```

Output:

count(刷頻次數)

contents(刷頻內容)

climax(高潮)

Input:

clk(時脈)

niceplay(當實況主打出漂亮操作時為 1)

rst(reset)

第二部分:程式碼

```
always@(posedge clk,posedge rst) begin
    if(rst) begin
        climax<=1'b0;
    end
    else begin
        climax<= niceplay;
    end
end
```

每當時脈正源來時或是 RST 正源來時，進入此 always block，如果 RST 是正的，則使 climax 為零(重製)，其他(時脈正源來時)把 niceplay 的值給 climax。(D 型正反器)

```

assign o=climax;
always@(posedge clk,posedge rst) begin
    if(rst) begin
        count<=8'b0;
    end
    else begin
        if(o==1) begin
            count<=count+1;|
        end
    end
end
end

```

把 climax 的值 assign 給 wire o(作為之後要用到的判斷依據)。

當時脈正源或 RST 正源時進入此 always block，當 RST 為零(重製)，count 歸零。其他當 o 等於一，則計數一次(刷頻次數)

```

always @(count) begin
    case(count)
        8'b00000001 : contents <= 3'b000;
        8'b00000010 : contents <= 3'b001;
        8'b00000100 : contents <= 3'b010;
        8'b00001000 : contents <= 3'b011;
        8'b00010000 : contents <= 3'b100;
        8'b00100000 : contents <= 3'b101;
        8'b01000000 : contents <= 3'b110;
        8'b10000000 : contents <= 3'b111;
        default:      contents <= 3'bxxx;
    endcase
end
endmodule

```

當 **count** 發生變化時，進入此 **always block**，當 **count** 次數增加時，刷頻內容會隨之改變，其餘為 **default**。(編碼器)

成果展示：

```

`timescale 1ns / 1ns

module tb_MUX();
reg niceplay;
reg clk;
reg rst;
wire climax;
wire [7:0] count;
wire [2:0] contents;

Dff dut(clk,niceplay,rst,climax,count,contents );

```

Tesbench 的宣告，宣告名稱與主程式相同。

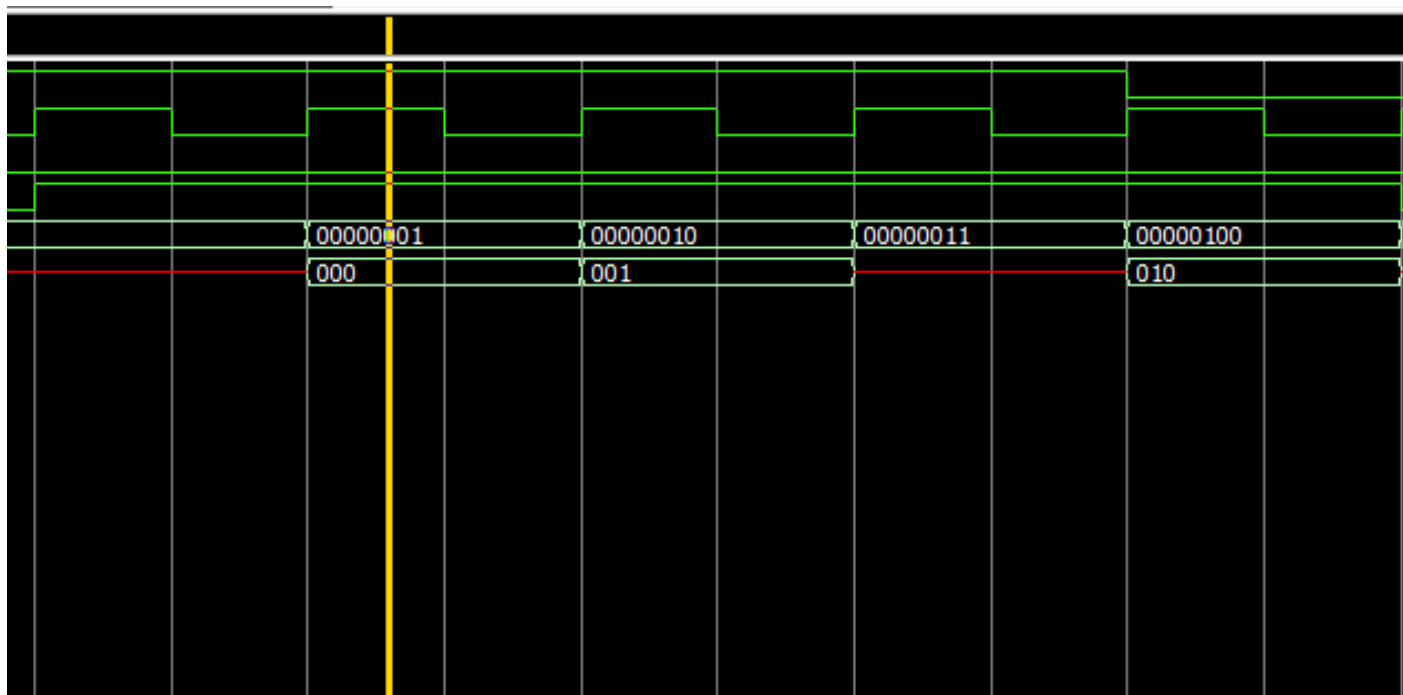
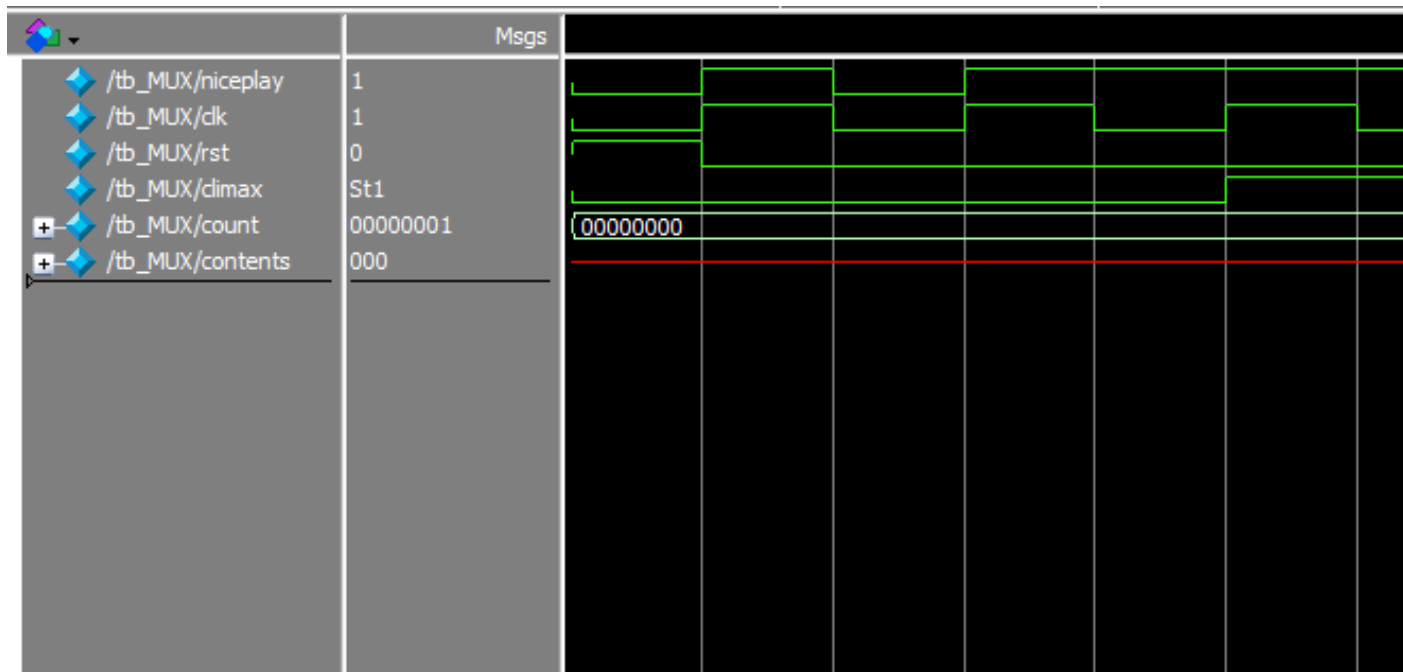
```
initial begin
    clk=0;
    forever #10 clk = ~clk;
end
```

第一個 initial begin 一開始 clk 為零，再來每十個單位時間 clk 會反轉。

```
initial begin
    rst=1;
    niceplay <= 0;
    #10;
    rst=0;
    niceplay <= 1;
    #10;
    niceplay <= 0;
    #10;
    niceplay <= 1;
    #100;
    niceplay <= 0;
end
endmodule
```

最後一個 initial begin。

波型圖:



未來展望：

希望未來能增加速度調整功能(利用類似除頻器的方式)、自動選擇頻內容功能、改善延遲(因為在同個模組使用兩個

always) ，以及能真正使用。

參考資料：

<https://www.youtube.com/watch?v=9rIjA1r5EB0>

<https://zh.wikipedia.org/zh-tw/%E8%A7%A6%E5%8F%91%E5%99%A8>

<https://kknews.cc/zh-tw/tech/3jgx253.html>