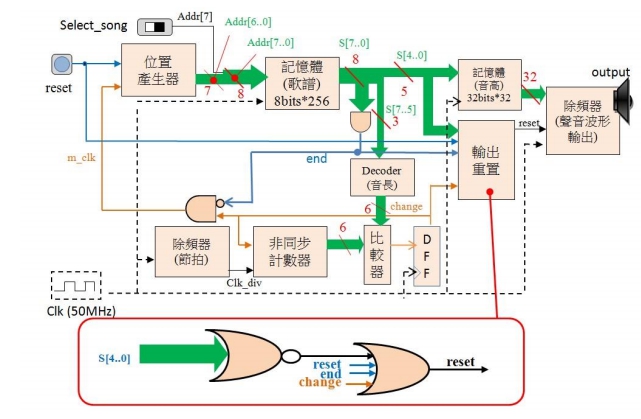
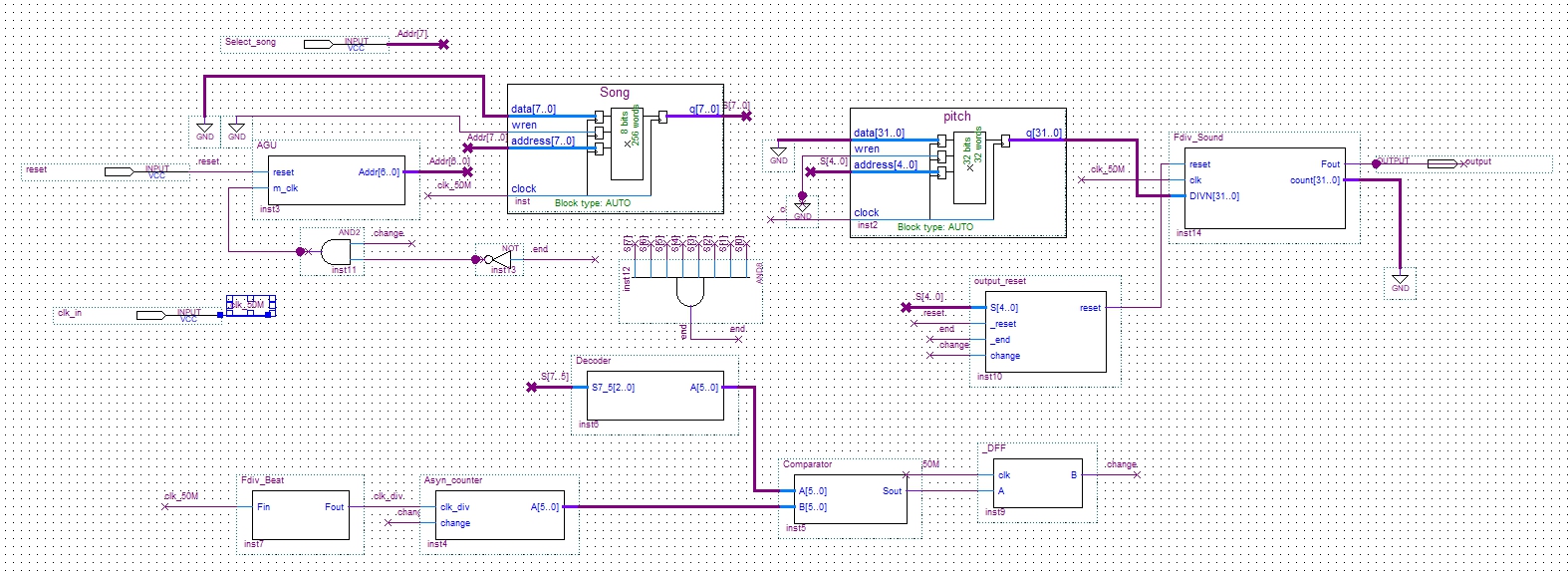
**數位實驗3：簡易音樂盒設計**

**第十四組**

**許銘森B063011058**

**整體架構圖**

****



**各模組的Verilog code**

1.位址產生器：

module AGU(clk,A);

input clk;

output [4:0]A;

reg [4:0]A;

always @(posedge clk)

A=A+1;

endmodule

**註解:**自動產生出記憶體位址。always @(posedge clk) 就是在clk為正緣時觸發A=A+1; 以達到自動產生記憶體位置的效果，其實這也只是一個基本counter記數器的程式碼。將所選曲目的起始位址送給 Memory (歌譜)，並當 clk 訊號正緣觸發時將位址加 1，使 Memory (歌譜)送出下一個音符的資料，reset 按下後位址歸零。

1. 兩個除頻器：

除頻器(節拍)：

module Fdiv\_Beat(Fin,Fout);

input Fin;

output Fout;

wire [31:0]DIVN=32'd3750000;

wire [31:0]\_DIVN;

reg [31:0]count;

reg Fout;

wire \_Fout;

assign \_DIVN={1'b0,DIVN[31:1]};

always @(posedge Fin)

if (count >= DIVN) count=32'd1;

else count<=count+1;

assign \_Fout=(count>\_DIVN)?1'b1:1'b0;

always @(negedge Fin)

Fout<=\_Fout;

Endmodule

**註解:**把板子上提供的 50M Hz 轉換成最短音符長度(32 分音符)。DIVN=32'd3750000，(50M除500K除6)=10/6Hz。

除頻器(輸出音)：

module Fdiv\_Sound(reset,clk,Fout,count,DIVN);

input clk,reset;

input [31:0]DIVN;

output Fout;

output [31:0]count;

wire [31:0]\_DIVN;

reg [31:0]count;

reg Fout;

wire \_Fout;

assign \_DIVN={1'b0,DIVN[31:1]};

always @(posedge clk)

begin

if (reset ==1'b1) count<=32'd1;

else if (count >= DIVN) count=32'd1;

else count<=count+1;

end

assign \_Fout=(count>\_DIVN)?1'b1:1'b0;

always @(negedge clk)

Fout<=\_Fout;

endmodule

**註解:**把板子上提供的 50M Hz 轉換成欲輸出音符的頻率。

1. 非同步計數器：

module Asyn\_counter(clk\_div,change,A);

input clk\_div,change;

output [5:0]A;

reg [5:0]A;

always@(posedge clk\_div or posedge change)

begin

if(change ==1'b1)

A<=6'd0;

else

A<= A+6'd1;

end

endmodule

**註解:**當 clk\_div 正緣觸發時加 1，change 訊號正緣時reset,

always@(posedge clk\_div or posedge change),此模組為非同步計數器(reset 與 clk 不同步)。

1. Decoder：

module Decoder(S7\_5,A);

input [2:0]S7\_5;

output [5:0]A;

reg [5:0]A;

always@(S7\_5)

begin

case(S7\_5)

3'd0:A<=6'b000001;

3'd1:A<=6'b000010;

3'd2:A<=6'b000100;

3'd3:A<=6'b001000;

3'd4:A<=6'b010000;

3'd5:A<=6'b011000;

3'd6:A<=6'b100000;

default:A<=6'd0;

endcase

end

endmodule

**註解:**將樂譜的前 3bits 解碼成 6bits 的音符長度供比較器做比對。

1. 比較器：

module Comparator(Sout,A,B);

input [5:0]A,B;

output Sout;

assign Sout=(A-B==6'd0)?1'b1:1'b0;

Endmodule

**註解:**比對是否已達欲輸出音符的長度，若已達長度則拉起 change

訊號。Sout=(A-B==6'd0)?1'b1:1'b0，A=B時，Sout為1，代表音符已達輸出長度，要輸出下一音符。

1. 輸出重置：

module output\_reset(S,\_reset,\_end,change,reset);

input [4:0]S;

input \_reset,\_end,change;

output reset;

wire \_S;

assign \_S=~(S[4]|S[3]|S[2]|S[1]|S[0]);

assign reset=(\_S|\_reset|\_end|change);

endmodule

**註解:**當 \_reset(從頭播放)、\_ end(歌曲結束)、change(換音)其中有訊號為“1”

時，reset為1。assign reset=(\_S|\_reset|\_end|change); OR Gate

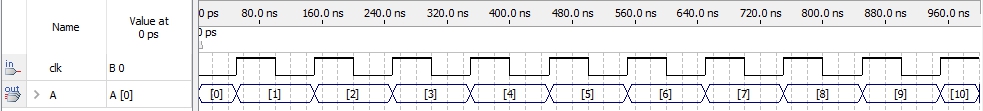
或

S[4..0]皆為 0 時(休止符)，reset為1。

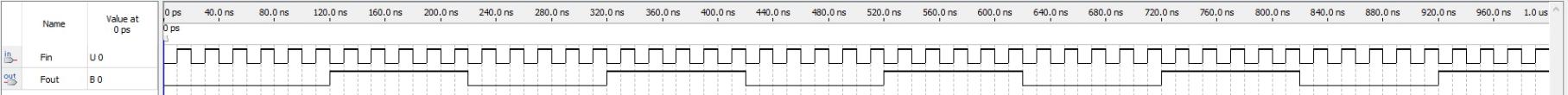
assign \_S=~(S[4]|S[3]|S[2]|S[1]|S[0]); NOR Gate

**各模組的波形模擬**

1. 位址產生器

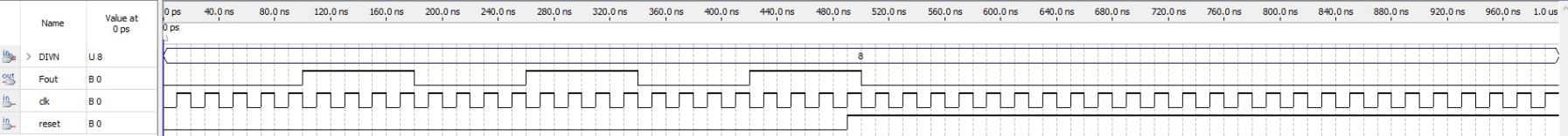
解釋：輸入為clk是頻率，輸出A在正緣時觸發了A=A+1，由上圖可見模組運作正常。

2.兩個除頻器：

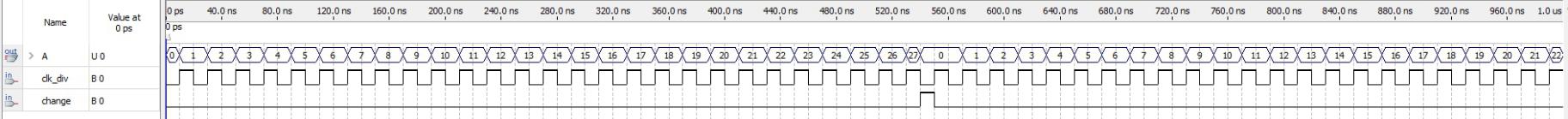
除頻器(節拍)：

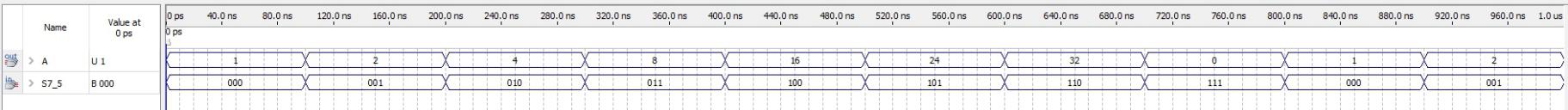
解釋：輸入為clk是頻率，DIVN=10,Fout為Fin十個波合成一個，由上圖可見模組運作正常。

除頻器(輸出音)：

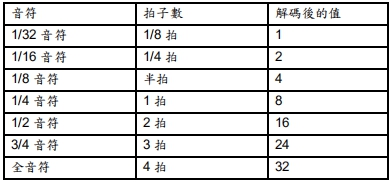
解釋：輸入為clk是頻率，DIVN是DIVN=8,Fout為Fin十個波合成一個，由上圖可見模組運作正常。

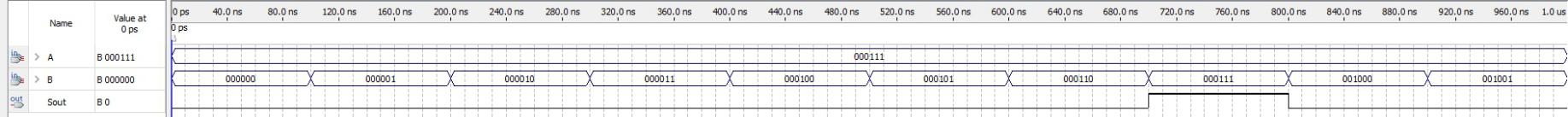
3.非同步計數器：

解釋：輸入為clk是頻率，A為拍子clk正緣時+1，change為1是代表拍子到了，A重新數，由上圖可見模組運作正常。

1. Decoder：

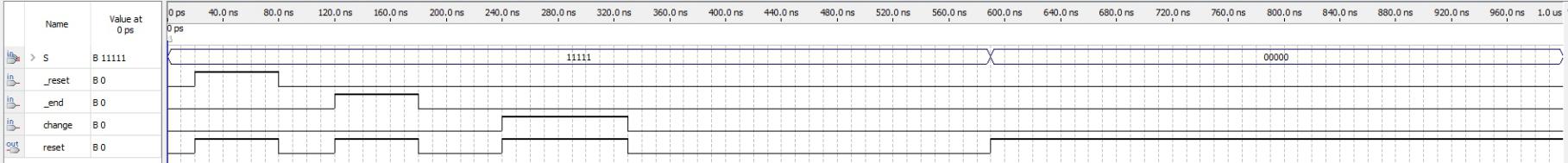
解釋：輸入為S7\_\_5是音符，A為拍子，由上圖可見模組運作正常。



1. 比較器：

解釋：輸入A為音長，B為拍子，A=B是Sout為1代表音符的拍子到了可以數下一個音符，由上圖可見模組運作正常。

6.輸出重置：

解釋：當 \_reset(從頭播放)、\_end(歌曲結束)、change(換音)其中有訊號為“1”

時，拉起 reset 訊號。assign reset=(\_S|\_reset|\_end|change); OR

S[4..0]皆為 0 時(休止符)拉起 reset 訊號。

assign \_S=~(S[4]|S[3]|S[2]|S[1]|S[0]); NOR

**創意介紹**

這次的音樂盒電路，我認爲可以改成不同的功能，例如把影像放在記憶體裏，透過位置產生器和熒幕輸出，就能達到電子相框的效果。Decoder可以用來解輸出時長，便可以做到自動換影像的功能。Select可以用來分要輸出的相冊。

**實驗心得**

這次實驗有利用到基本的模組，那就是除頻器，原理和上一次一樣，只是把除數改了。但是這次實驗比較難，因為code都要自己修改，但是還是要有一些模組要自己打，例如輸出重置。其實當天打code是很趕，應該在上課前先預習下次的內容，把code先打好，再用波形模擬確認無誤，上課時只需要花時間debug就好了。