**電工實驗（四）**

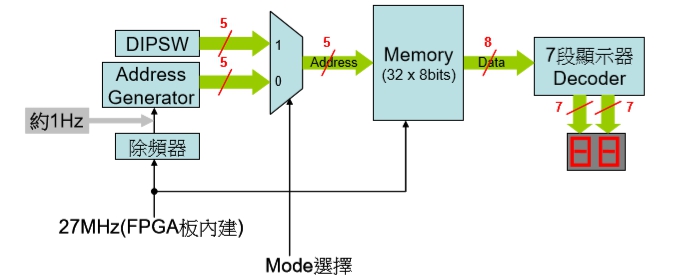
**微處理器實驗**

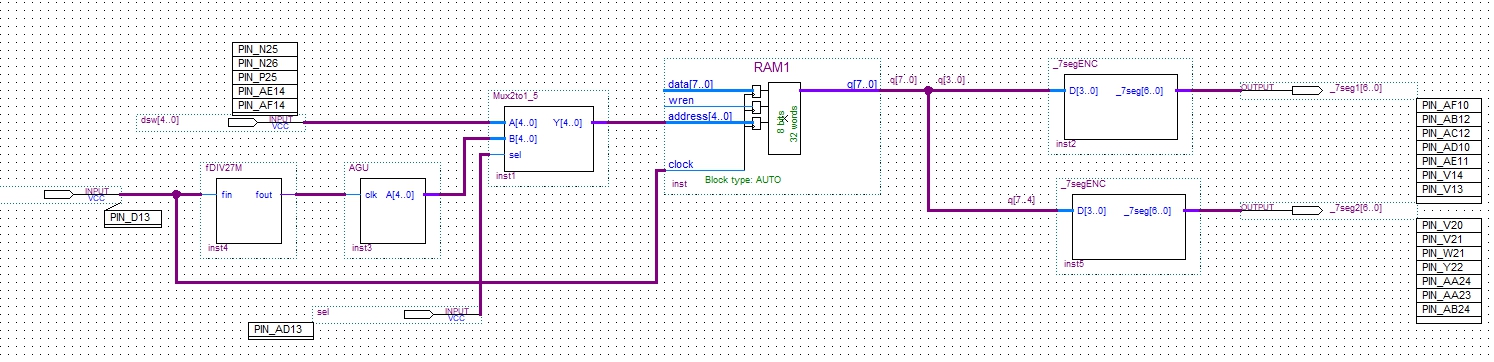
**實驗1：基礎I/O實驗(I)**

**第十四組**

**許銘森B063011058**

**整體架構圖**





**各模組的Verilog code**

1.二對一多工器

module Mux2to1\_5(A,B,sel,Y);

input[4:0]A,B;

input sel;

output[4:0]Y;

assign Y=(sel)?B:A;

Endmodule

**註解:**選擇記憶體位址的來源。模式 0 時，記憶體位址由位址產 生器決定；模式 1 時，記憶體位址由 Dip Switch(DIPSW)決定。此模組的工作原理，重點在於assign Y=(sel)?B:A; 這個語法，如果sel為1，就選擇B作輸出，為0則是A，令到我們可以選擇記憶體位置的產生方法。

2.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。

1. 除頻器：

module fDIV27M(fin,fout);

input fin;

output fout;

wire [31:0]DIVN,\_DIVN;

reg [31:0]count;

reg fout;

always @(posedge fin)

count=(count>=DIVN)?32'd0:count+1;

assign DIVN=32'd27000000;

assign \_DIVN=DIVN>>1;

always @(negedge fin)

if(count<\_DIVN)

fout<=1'b1;

else

fout<=1'b0;

endmodule

**註解:**把板子上提供的 27M Hz 轉換成 1 Hz。除頻器的基本原理就是透過count數輸入頻率正緣次數，然後利用assign \_DIVN=DIVN>>1; 去做一個移位達到除二的效果，如果 \_DIVN大於count就判定為高準位，小於則為低準位。這樣就能把許多個高頻波輸出成低頻波，以這次實驗做例子，就是把一秒27M個波輸出成一個波。

4.7段顯示器解碼器：

module \_7segENC(D,\_7seg);

input [3:0]D;

output [6:0]\_7seg;

reg [6:0]\_7seg;

always @(D)

begin

case(D)

4'd0:\_7seg=7'b1000000;

4'd1:\_7seg=7'b1111001;

4'd2:\_7seg=7'b0100100;

4'd3:\_7seg=7'b0110000;

4'd4:\_7seg=7'b0011001;

4'd5:\_7seg=7'b0010010;

4'd6:\_7seg=7'b0000010;

4'd7:\_7seg=7'b1111000;

4'd8:\_7seg=7'b0000000;

4'd9:\_7seg=7'b0011000;

4'd10:\_7seg=7'b0001000;

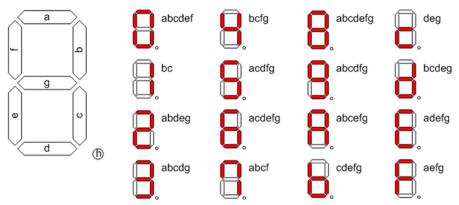
4'd11:\_7seg=7'b0000011;

4'd12:\_7seg=7'b1000110;

4'd13:\_7seg=7'b0100001;

4'd14:\_7seg=7'b0000110;

4'd15:\_7seg=7'b0001110;

 endcase

end

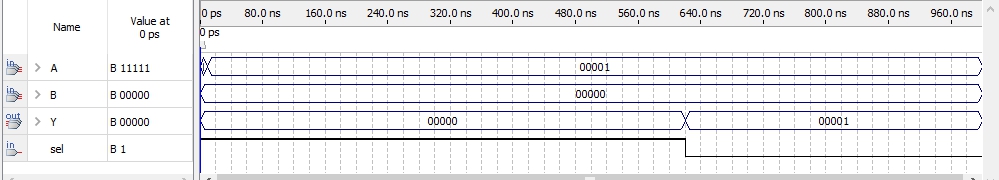
Endmodule

**註解:**把記憶體的內容值解碼顯示上 7 段顯示器上。原理是把7 段顯示器該亮的地方輸出成0，暗的則是1，以4'd8:\_7seg=7'b0000000;為例，當輸入為8時則全燈要亮，所以輸出為7個0。

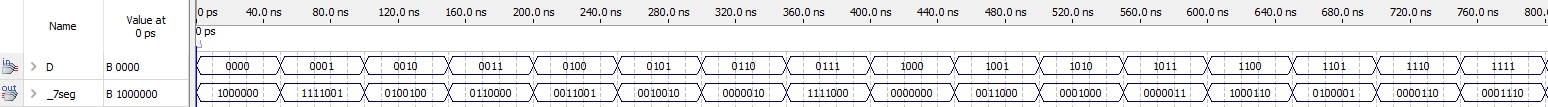
5.Memory：由 megafunction 產生出來的記憶體模組。

**各模組的波形模擬 (須解釋如何驗證功能正確)**

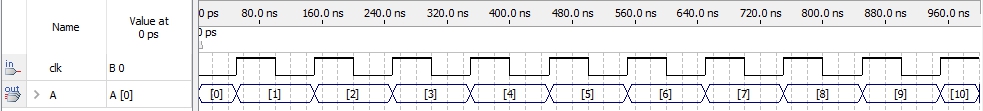
1.二對一多工器

解釋：輸入分別為AB代表直撥開關及位置產生器的地址，從上圖可見sel為1時，輸出y為B的位置，sel為0時，輸出y為A的位置，模組運作正常。

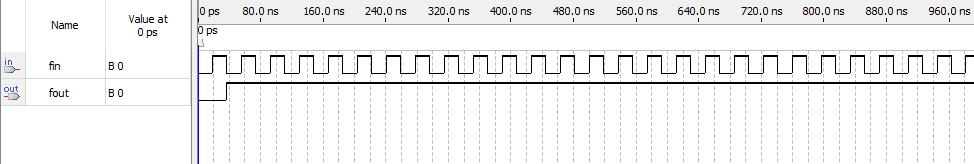
2.7段顯示器解碼器

解釋：輸入D在模擬中，我設定了輸入由0至15，輸出也在7段顯示器要亮的對應位置輸出為0。由上圖可見模組運作正常。

3.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。

1. 除頻器

解釋：fin為輸入頻率，我設了27M與實驗一樣，輸出fout也有正常除頻，但由於模擬時間不夠久，不能見證它下去的瞬間，但實驗時模組的確有正常運作。

**創意介紹**

這次的記憶體輸出控制電路，我認爲可以運用在日常生活中，例如我們可以把朋友的生日放進記憶體裏，然後當你想知道1號朋友的生日時就輸入1號地址去存取資料，但我是不會這樣做的。因爲我認爲有更加好的應用，就是把照片放在記憶體裏，透過位置產生器和熒幕輸出，就能達到電子相框的效果。

**實驗心得**

這次並不是我第一次用verilog寫程式，我修過硬體描述語言課程，當時成績一般，有些模組并不能真正瞭解其原理和運作，只是在考試前把程式碼背起來。這次實驗其中除頻器的運作就是我當初不太理解的，經過這次實驗后，我覺得我懂了，也因爲先前的經驗，實驗進行得十分順利。其中在硬體描述語言沒有學到的就是把程式碼化作symbol，我覺得這個功能好方便，以往各模組的串聯需要一個主程式才能運作，現在直接化作symbol就可以直接接綫，大大節省時間！