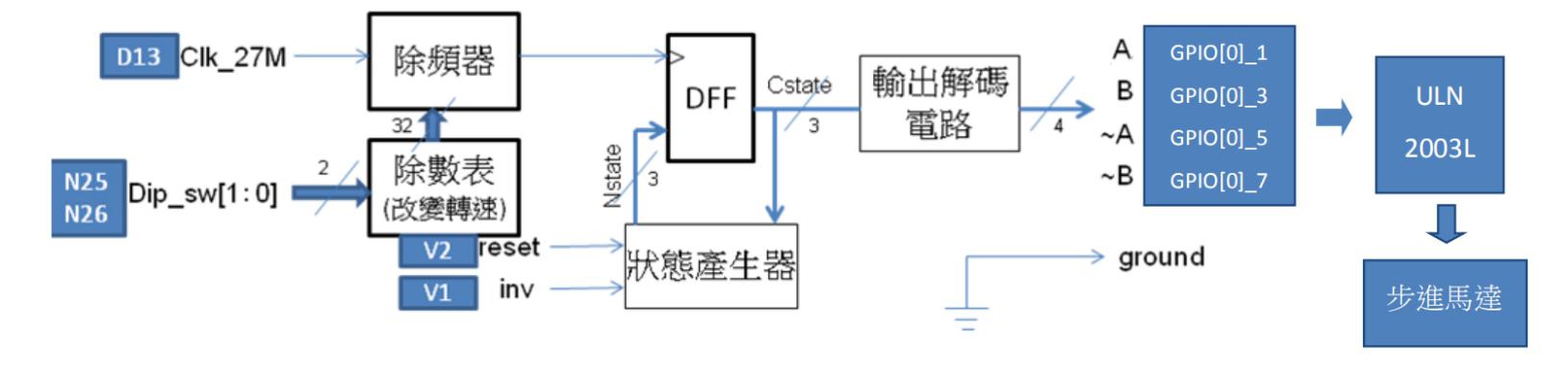
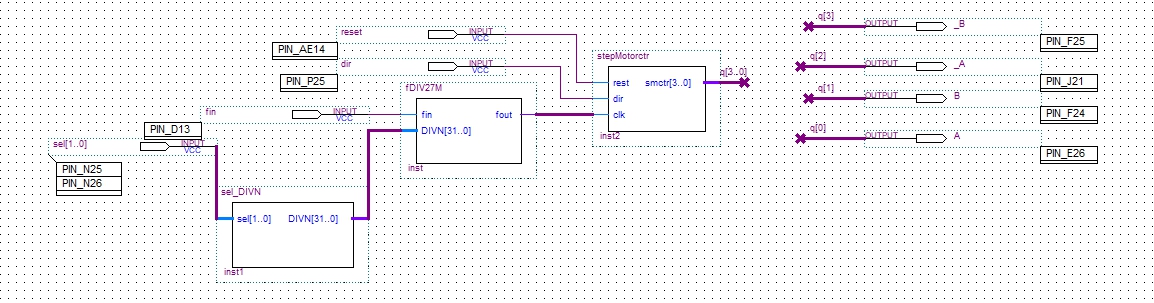
**實驗2：步進馬達控制電路設計**

**第十四組**

**許銘森B063011058**

**整體架構圖**

****

****

**各模組的Verilog code**

1.除頻器：

module fDIV27M(fin,DIVN,fout);

input fin;

input [31:0]DIVN;

output fout;

wire [31:0]\_DIVN;

reg [31:0]count;

reg fout;

always @(posedge fin)

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

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

always @(negedge fin)

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

endmodule

**註解:**把板子上提供的 27M Hz 轉換成低頻。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。 DIVN的值是透過sel\_DIVN去決定

2.除數表：

module sel\_DIVN(sel,DIVN);

input[1:0]sel;

output[31:0]DIVN;

reg[31:0]DIVN;

always@(sel)

case(sel)

2'd0:DIVN=32'd64000;

2'd1:DIVN=32'd128000;

2'd2:DIVN=32'd256000;

default:DIVN=32'd512000;

endcase

endmodule

**註解:**利用case語法去決定DIVN的值，這個值會是除頻器的DIVN值，從而達到調整速度的效果。

3.馬達控制器：

module stepMotorctr(rest,dir,clk,smctr);

input rest,dir,clk;

output[3:0]smctr;

reg [3:0]smctr;

reg[2:0]cs,ns;

always @(posedge clk)

cs<=ns;

always@(cs)

case(cs)

3'd0:smctr<=4'b0000;// \_B,\_A,B,A

3'd1:smctr<=4'b0011;

3'd2:smctr<=4'b0110;

3'd3:smctr<=4'b1100;

3'd4:smctr<=4'b1001;

default:smctr<=4'b0000;

endcase

always@(cs)

if(rest==1)

ns<=3'd0;

else

case(cs)

3'd0:ns=3'd1;

3'd1:ns=(dir)?3'd2:3'd4;//rotate direction decide state

3'd2:ns=(dir)?3'd3:3'd1;

3'd3:ns=(dir)?3'd4:3'd2;

3'd4:ns=(dir)?3'd1:3'd3;

default:ns=3'd0;

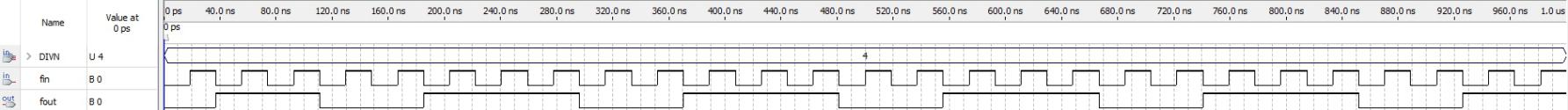
endcase

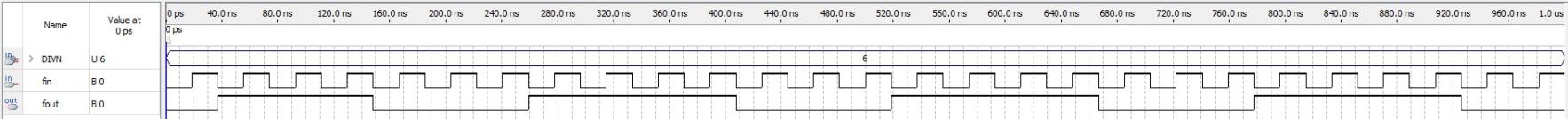
endmodule

**註解:**smctr是個四位元輸出，分別代表\_B\_ABA，也是馬達輸出的波形。dir是控制方向位元，3'd1:ns=(dir)?3'd2:3'd4; 其中？Operator可以根據dir決定下一個state是順向或逆向。 rest為1可以停止馬達的轉動。

**各模組的波形模擬 (須解釋如何驗證功能正確)**

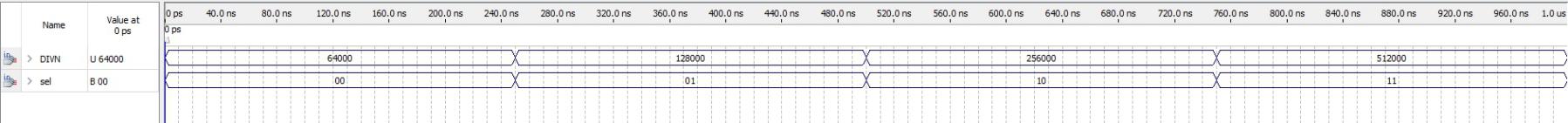
1.除頻器：

 1 2 3 4 5

 1 2 3 4 5 6 7

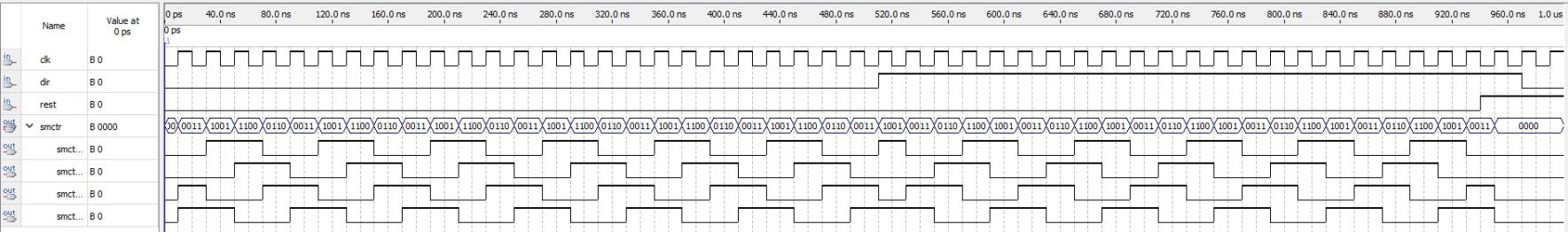
解釋：fin為輸入頻率，我設了27M與實驗一樣，我把DIVN分別設為4和6，fout也有正常除頻，4時fout把5個波合在一起，6時fout把7個波合在一起。模組有正常運作。

2.除數表：



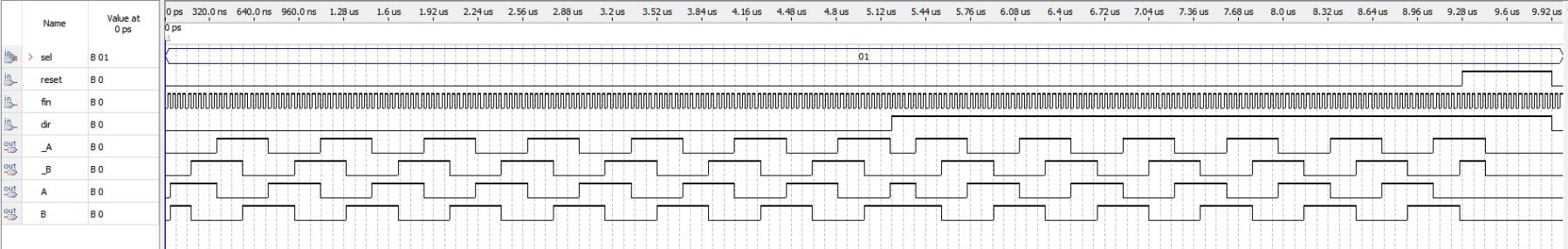
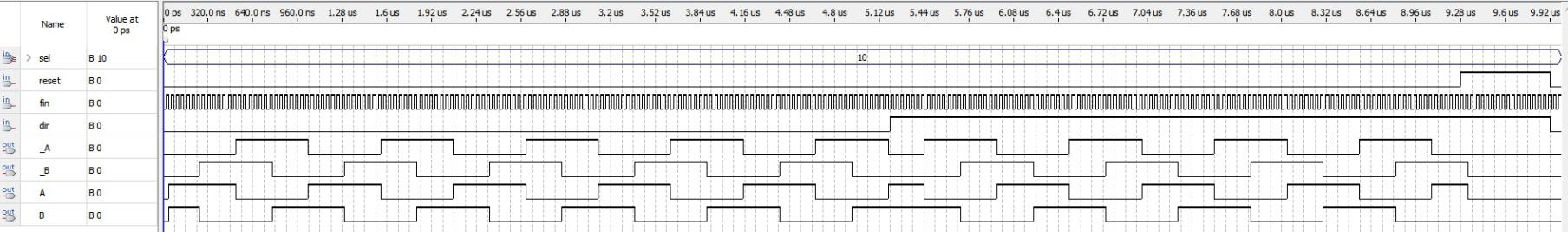
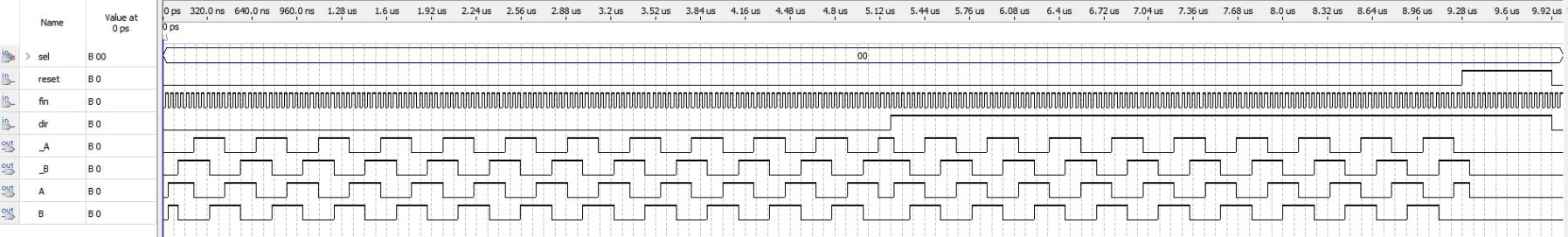
解釋：sel為不同值時，DIVN的確有輸出為我設定的值。sel為11是default，DIVN為512000是正常的。模組有正常運作。

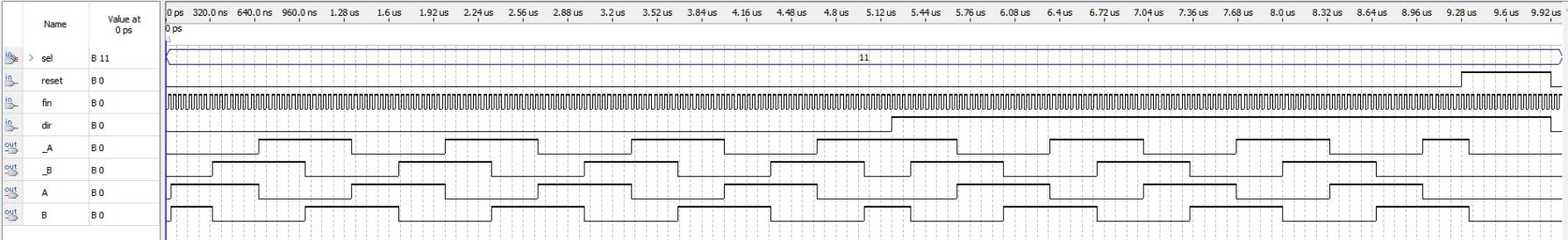
3.馬達控制器：

 dir=1 rest=1

解釋：dir是控制方向位元，為1是逆向，輸出波形也變了方向。rest為1時，波形停止輸出。模組有正常運作。

1. BDF：

****

**dir=1 rest=1**

解釋：fin為輸入頻率，我設了27M與實驗一樣。dir是控制方向位元，為1是逆向，輸出波形也變了方向。rest為1時，波形停止輸出。sel為除數表的除數，為了方便波形模擬把除數定為2，4，6，8。除數越大，輸出波越寬，馬達轉越慢。模組有正常運作。

**創意介紹**

這次的步進馬達控制電路，我認爲可以運用在日常生活中，例如這次的步進馬達控制電路，我認爲可以運用在日常生活中，例如可以在玩具車上運用因爲我們可以透過除頻器控制馬達的轉速，以使玩具車有不同的速度擋位。也可以運用在機器人，機器人身上有很多馬達其轉速可以令本體做出不同動作，但是需要額外加一個控制器去控制幾個馬達。

**實驗心得**

這次實驗有利用到上一次的模組，那就是除頻器啦，上一次並不能除不同的數字，這次新加入了除數表去選擇不同的數字,達到了調整馬達速度的效果。還有今次實驗新用到的是馬達控制模組，裏面的架構其實只是簡單的狀態轉換。這次實驗十分有趣，與之前描述語言課程學到的不同，是把verilog真正運用在一些日常電路中，令我明白verilog的强大之處，這些程式看似複雜，但其實藏著的概念很簡單，最重要的是我們如何運用。