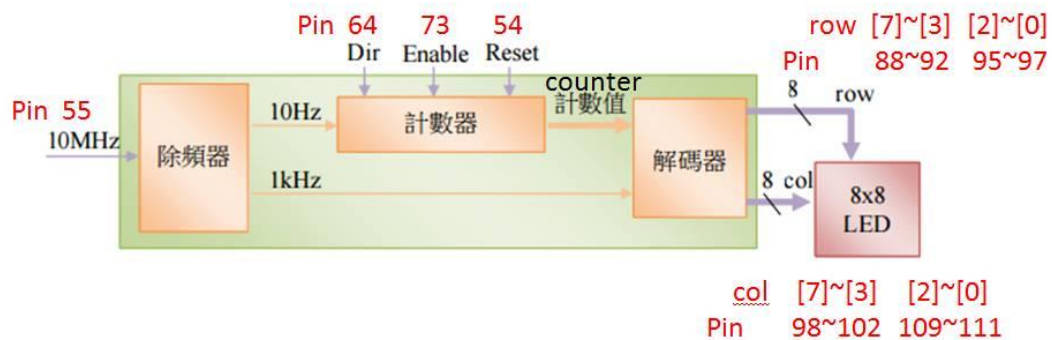


1. 組別：第 22 組
2. 組員：404415073 電機二 蔡孟勳 404415055 電機二 劉恩瑞
3. 題目名稱：實驗 6 跑馬燈
4. 功能說明：

這次的實驗為撰寫一個跑馬燈，輸入訊號為一個 10MHz 的 clock (當正緣觸發時可執行程式碼)、一個脈波 (使跑馬燈歸零並重新開始...即 Reset)、兩個 0 或 1 的訊號(分別控制計數器是否啟動與計數器遞增或遞減...即 Enable 和 Dir)；輸出訊號為 8*8 點矩陣 LED (用來顯示最後的圖形結果)。

整個電路的功能為：隨著時間變化改變圖形的跑馬燈。

5. 硬體架構圖：



電路設計的想法:

這次的實驗其實可以概略分為三個部分：除頻器、計數器、解碼器。由於實作板只能提供 10MHz 的脈波，因此我們先設計一個除頻器讓 10MHz 分別降至 1kHz 和 1Hz，當 1Hz 的 clock 正緣觸發且 Enable=1 時，啟動計數功能，其中 Dir=1 時，計數器遞增計數；Dir=0 時，計數器遞減計數，最後則是依照計數器的數字決定顯示的圖形(解碼器)，方法為：當 1kHz 正緣觸發時，另一個計數器遞增 1，0~7 分別決定讓矩陣 LED 的燈各亮一排，最後再設計 col 決定讓某一排的某幾個燈，雖然每一排的時間會差 1ms，但因為相差時間甚小，利用視覺暫留的原理則可顯示完整的圖形。這次的實驗有設計 Reset，當按下 Reset 時，整個計數器會歸零並重新開始，當然 Reset 也必須要消除跳現象，這樣就可以完成這次的實驗了。

6. 程式碼&註解：

```
1 module exp6(row,col,MHz,Dir,Enable,Reset); //跑馬燈模組宣告
2   input MHz,Dir,Enable,Reset; //輸入為MHz,Dir,Enable,Reset
3   output reg [7:0]row,col; //輸出為8-bits共16個數row[0]~[7],col[0]~[7]
4   wire kHz,HZ,out; //跑馬燈的邏輯電路圖中會用到3條線連接不同模組
5   reg [1:0]counter; //保留counter[0]~[1]的值到下一次指定新值
6   reg [2:0]count; //保留count[0]~[2]的值到下一次指定新值
7
8   div1 d1(kHz,MHz); //使用除頻器1的模組，輸出為kHz，輸入為MHz
9   div2 d2(Hz,MHz); //使用除頻器2的模組，輸出為Hz，輸入為MHz
10  |
11  debounce del(MHz,Reset,out); //使用消除彈跳的模組，輸出為out，輸入為MHz,Reset
12
13  always@(posedge HZ or negedge out) //當Hz正緣觸發或out負緣觸發時，底下的Behavioral Model的敘述會被執行
14  if(~out) //如果out反相為true，執行以下敘述
15    counter=0; //counter的值等於0
16  else //其他情況
17    if(~Enable) //如果Enable反相為true，執行以下敘述
18    if(~Dir) //如果Dir反相為true，執行以下敘述
19      counter<=counter+1; //counter的值+1
20    else //其他情況
21      counter<=counter-1; //counter的值-1
22
23  always@(posedge kHz) //當kHz正緣觸發時，底下的Behavioral Model的敘述會被執行
24  begin //Behavioral Model開始
25    count<=count+1; //counter的值+1
26    case(count) //counter的值為true時，執行以下敘述
27      2'd0: //2位元，十進制，數值為0
28        case(count) //count的值為true時，執行以下敘述
29          3'd0: //3位元，十進制，數值為0
30            begin //開始
31              row=8'b1000_0000; //row第1列為1，其餘為0
32              col=8'b1111_1110; //col第1,2,3,4,5,6,7行為1，其餘為0
33            end //結束
34          3'd1: //3位元，十進制，數值為1
35            begin //開始
36              row=8'b0001_0000; //row第4列為1，其餘為0
37              col=8'b0111_1100; //col第2,3,4,5,6行為1，其餘為0
38            end //結束
39          default: //其他
40            begin //開始
41              row=8'b1111_1111; //row第1,2,3,4,5,6,7,8行為1
42              col=8'b0100_0000; //col第2行為1，其餘為0
43            end //結束
44          endcase //結束case
45
46      2'd1: //2位元，十進制，數值為1
47        case(count) //count的值為true時，執行以下敘述
48          3'd0: //3位元，十進制，數值為0
49            begin //開始
50              row=8'b1000_0000; //row第1列為1，其餘為0
51              col=8'b0111_1110; //col第2,3,4,5,6,7行為1，其餘為0
52            end //結束
53          3'd1: //3位元，十進制，數值為1
54            begin //開始
55              row=8'b0001_0000; //row第4列為1，其餘為0
56              col=8'b0111_1110; //col第2,3,4,5,6,7行為1，其餘為0
57            end //結束
58          3'd2: //3位元，十進制，數值為2
59            begin //開始
60              row=8'b1111_0000; //row第1,2,3,4行為1，其餘為0
61              col=8'b0000_0010; //col第7行為1，其餘為0
62            end
63          default: //其他
64            begin //開始
65              row=8'b1111_1111; //row第1,2,3,4,5,6,7,8行為1
66              col=8'b0100_0000; //col第2行為1，其餘為0
67            end //結束
68          endcase //結束case
69
```

```

70 2'd2: //2位元,十進制,數值為2
71 case(count) //count的值為true時,執行以下敘述
72 3'd0: //3位元,十進制,數值為0
73 begin //開始
74 row=8'b1000_0000; //row第1列為1,其餘為0
75 col=8'b0111_1110; //col第2,3,4,5,6,7行為1,其餘為0
76 end //結束
77 3'd1: //3位元,十進制,數值為1
78 begin //開始
79 row=8'b0001_0000; //row第4列為1,其餘為0
80 col=8'b0001_1110; //col第4,5,6,7行為1,其餘為0
81 end //結束
82 3'd2: //3位元,十進制,數值為2
83 begin //開始
84 row=8'b0001_1111; //row第4,5,6,7,8列為1,其餘為0
85 col=8'b0000_0010; //col第7行為1,其餘為0
86 end //結束
87 3'd3: //3位元,十進制,數值為3
88 begin //開始
89 row=8'b0000_0001; //row第8列為1,其餘為0
90 col=8'b0111_1110; //col第2,3,4,5,6,7行為1,其餘為0
91 end //結束
92 default: //其他
93 begin //開始
94 row=8'b1111_1111; //row第1,2,3,4,5,6,7,8列為1
95 col=8'b0100_0000; //col第2行為1,其餘為0
96 end //結束
97 endcase //結束case
98
99 2'd3: //2位元,十進制,數值為3
100 case(count) //count的值為true時,執行以下敘述
101 3'd0: //3位元,十進制,數值為0
102 begin //開始
103 row=8'b1000_0000; //row第1列為1,其餘為0
104 col=8'b0111_1110; //col第2,3,4,5,6,7行為1,其餘為0
105 end //結束
106 3'd1: //3位元,十進制,數值為1
107 begin //開始
108 row=8'b0001_0000; //row第4列為1,其餘為0
109 col=8'b0111_1110; //col第2,3,4,5,6,7行為1,其餘為0
110 end //結束
111 3'd2: //3位元,十進制,數值為2
112 begin //開始
113 row=8'b1111_1111; //row第1,2,3,4,5,6,7,8列為1
114 col=8'b0000_0010; //col第7行為1,其餘為0
115 end //結束
116 default: //其他
117 begin //開始
118 row=8'b1111_1111; //row第1,2,3,4,5,6,7,8列為1
119 col=8'b0100_0000; //col第2行為1,其餘為0
120 end //結束
121 endcase //結束case
122 endcase //結束case
123 end //Behavioral Model結束
124 endmodule //跑馬燈模組結束
125
126 module debounce(kHz,in,out); //消除彈跳的模組宣告
127 input kHz,in; //輸入為kHz,in
128 output out; //輸出為out
129 reg [6:0]d; //保留d[0]~d[6]的值到下一次指定新值
130 always@(posedge kHz) //當kHz正緣觸發時,底下的Behavioral Model的敘述會被執行
131 begin //Behavioral Model開始
132 d[6]<=d[5]; //d[6]的值變為d[5]的值
133 d[5]<=d[4]; //d[5]的值變為d[4]的值
134 d[4]<=d[3]; //d[4]的值變為d[3]的值
135 d[3]<=d[2]; //d[3]的值變為d[2]的值
136 d[2]<=d[1]; //d[2]的值變為d[1]的值
137 d[1]<=d[0]; //d[1]的值變為d[0]的值
138 d[0]<=in; //d[0]的值變為in的值
139 end //Behavioral Model結束
140 and(out,d[6],d[5],d[4],d[3],d[2],d[1],d[0]); //邏輯閘AND,輸入為d[0]~d[6];輸出為out
141 endmodule //消除彈跳的模組結束
142

```

```

143 module div1(kHz,clock); //除頻器1的模組宣告
144 input clock; //輸入為clock
145 output reg kHz; //保留輸出kHz的值到下一次指定新值
146 reg [12:0]counter; //保留輸出counter[0]~[12]的值到下一次指定新值
147 always@(posedge clock) //當clock正緣觸發時，底下的Behavioral Model的敘述會被執行
148 if(counter==4999) //如果counter的值等於4999時，執行以下敘述
149 begin //開始
150 counter<=0; //counter的值變為0
151 kHz<=~kHz; //kHz反相
152 end //結束
153 else //其他情況
154 counter<=counter+1; //counter的值+1
155 endmodule //除頻器1的模組結束
156
157 module div2(Hz,clock); //除頻器2的模組宣告
158 input clock; //輸入為clock
159 output reg Hz; //保留輸出Hz的值到下一次指定新值
160 reg [22:0]counter; //保留輸出counter[0]~[22]的值到下一次指定新值
161 always@(posedge clock) //當clock正緣觸發時，底下的Behavioral Model的敘述會被執行
162 if(counter==4999999) //如果counter的值等於4999999時，執行以下敘述
163 begin //開始
164 counter<=0; //counter的值變為0
165 Hz<=~Hz; //Hz反相
166 end //結束
167 else//其他情況
168 counter<=counter+1; //counter的值+1
169 endmodule //除頻器2的模組結束

```

7. 心得：

404415073 蔡孟勳

這次的實驗其實沒有增加新的語法或觀念，完全是利用之前學過的內容就可以完成的，但我們卻沒有如想像中打的順利，前面的除頻器跟計數器都還 OK，但到最後的解碼器其時就有點亂掉了，希望下次可以盡量讓腦袋保持清晰，這樣在打程式上才能比較順利。