

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 3 無號數乘法器
4. 功能說明：

這次的實驗用到的無號數乘法器(4-bits * 4-bits)

兩個輸入訊號分別為被乘數 A[3]~A[0]和乘數 B[3]~B[0]；

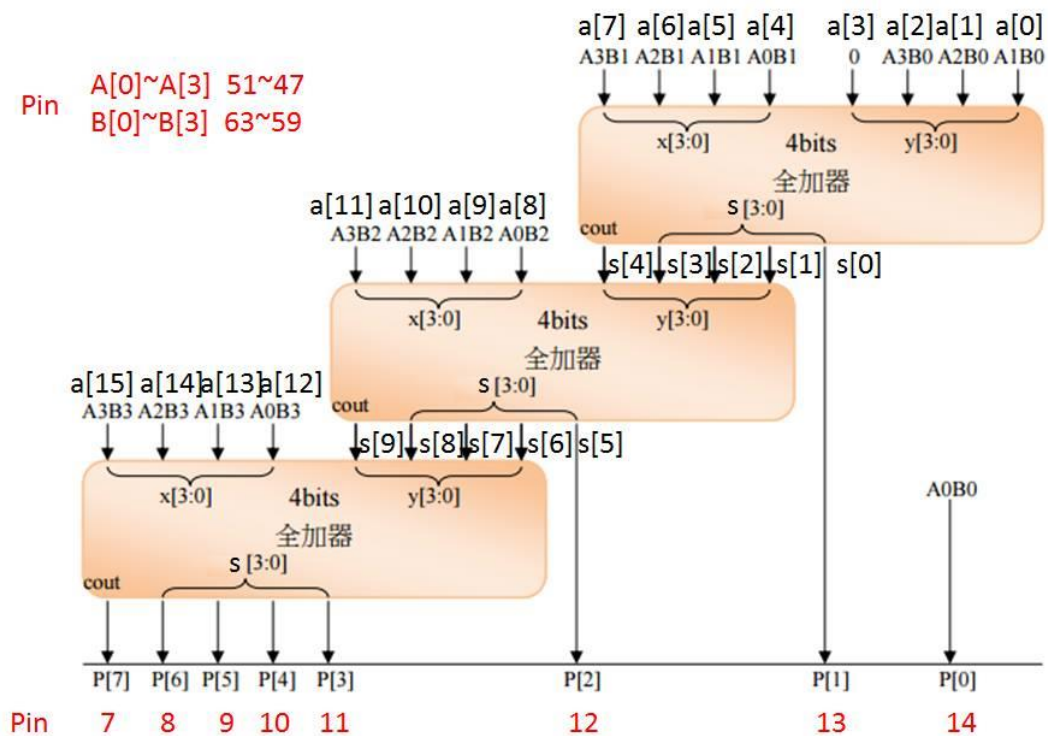
輸出訊號為 8-bits 乘積，其中 P[7]須全加器看是否有進位

整個電路的功能為：實行不考慮負號的 2 進位乘法。

				A3	A2	A1	A0	
				×	B3	B2	B1	B0
				<hr/>				
					A3B0	A2B0	A1B0	A0B0
			A3B1	A2B1	A1B1	A0B1		
		A3B2	A2B2	A1B2	A0B2			
+	A3B3	A2B3	A1B3	A0B3				
<hr/>								
P7	P6	P5	P4	P3	P2	P1	P0	__

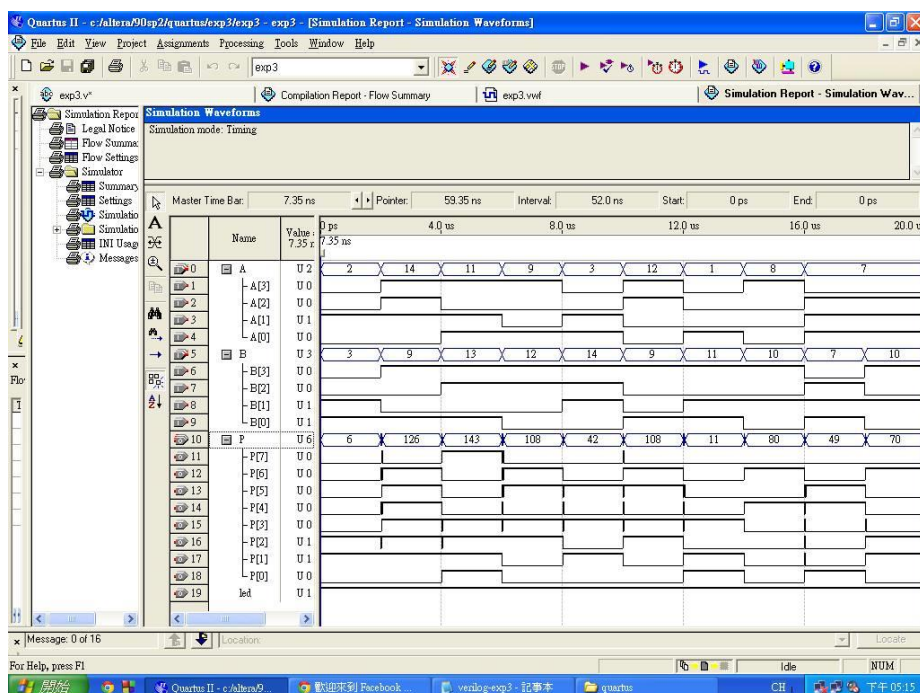
(無號數乘法器概念圖)

5. 硬體架構圖：



電路設計的想法：

其實乘法器就是全加器與 AND 閘的應用，因此我們先用 AND 邏輯閘將兩個 4-bits 輸入分別乘起來，然後使用 exp2 做的 4-bits 全加器並按照上方的硬體架構圖，將對應的輸入與輸出套入全加器，最後再 assign 輸出為 $P[7] \sim P[0]$ ，這樣無號數乘法器就完成了。



4-bits

全加器

波形模擬圖

6. 程式碼&註解：

```
1 module exp3(P,led,A,B); //無號數乘法器模組宣告
2 input [3:0]A,B; //輸入為4-bits共8個數A[0]~A[3],B[0]~B[3]
3 output [7:0]P; //輸出為8-bits共8個數P[0]~P[7]
4 output led; //輸出LED燈
5 wire [15:0]a; //無號數乘法器的邏輯電路圖中會用到16條線連接不同的模組
6 wire [9:0]s; //無號數乘法器的邏輯電路圖中會用到10條線連接不同的模組
7 and a1(P[0],A[0],B[0]); //邏輯閘AND · 輸入為A[0] B[0]；輸出為P[0]
8 and a2(a[0],A[1],B[0]); //邏輯閘AND · 輸入為A[1] B[0]；輸出為a[0]
9 and a3(a[1],A[2],B[0]); //邏輯閘AND · 輸入為A[2] B[0]；輸出為a[1]
10 and a4(a[2],A[3],B[0]); //邏輯閘AND · 輸入為A[3] B[0]；輸出為a[2]
11 assign a[3]=1'b0; //宣告a[3]為1位元,2進制,數值為0
12 and a5(a[4],A[0],B[1]); //邏輯閘AND · 輸入為A[0] B[1]；輸出為a[4]
13 and a6(a[5],A[1],B[1]); //邏輯閘AND · 輸入為A[1] B[1]；輸出為a[5]
14 and a7(a[6],A[2],B[1]); //邏輯閘AND · 輸入為A[2] B[1]；輸出為a[6]
15 and a8(a[7],A[3],B[1]); //邏輯閘AND · 輸入為A[3] B[1]；輸出為a[7]
16 and a9(a[8],A[0],B[2]); //邏輯閘AND · 輸入為A[0] B[2]；輸出為a[8]
17 and a10(a[9],A[1],B[2]); //邏輯閘AND · 輸入為A[1] B[2]；輸出為a[9]
18 and a11(a[10],A[2],B[2]); //邏輯閘AND · 輸入為A[2] B[2]；輸出為a[10]
19 and a12(a[11],A[3],B[2]); //邏輯閘AND · 輸入為A[3] B[2]；輸出為a[11]
20 and a13(a[12],A[0],B[3]); //邏輯閘AND · 輸入為A[0] B[3]；輸出為a[12]
21 and a14(a[13],A[1],B[3]); //邏輯閘AND · 輸入為A[1] B[3]；輸出為a[13]
22 and a15(a[14],A[2],B[3]); //邏輯閘AND · 輸入為A[2] B[3]；輸出為a[14]
23 and a16(a[15],A[3],B[3]); //邏輯閘AND · 輸入為A[3] B[3]；輸出為a[15]
24 exp2 z5(a[3:0],s[4],a[3:0],s[7:4],0); //使用4-bits全加器的模組exp2 s[0]~s[3],s[4]為輸出;a[0]~a[3],a[4]~a[7]為輸入,cin值為0
25 assign P[1]=s[0]; //宣告P[1]的值等於s[0]的值
26 exp2 z6(s[8:5],s[9],s[4:1],a[11:8],0); //再使用4-bits全加器的模組exp2 s[5]~s[8],s[9]為輸出;s[1]~s[4],a[8]~a[11]為輸入,cin值為0
27 assign P[2]=s[5]; //宣告P[2]的值等於s[5]的值
28 exp2 z7(P[6:3],P[7],s[9:6],a[15:12],0); //再使用4-bits全加器的模組exp2 P[3]~P[6],P[7]為輸出;s[6]~a[9],a[12]~a[15]為輸入,cin值為0
29 assign led=1'b1; //宣告LED燈為1位元,二進制,數值為1
30 endmodule //無號數乘法器模組結束

32 module exp2(sum,cout,x,y,cin); //4-bits全加器模組宣告
33 input [3:0]x,y; //輸入為4-bits共8個加數x[0]~x[3],y[0]~y[3]
34 input cin; //另一個輸入為進位值cin
35 output [3:0]sum; //輸出為4-bits本位和sum[0]~sum[3]
36 output cout; //另一個輸出為高位進位cout
37 wire k; //4-bits全加器的邏輯電路圖中會用到一條線連接不同的模組
38 F2 z3(sum[1:0],k,x[1:0],y[1:0],cin); //使用2-bits全加器的模組F2 sum[0]~sum[1],j為輸出;x[0]~x[1],y[0]~y[1],ci為輸入
39 F2 z4(sum[3:2],cout,x[3:2],y[3:2],k); //再使用一次2-bits全加器的模組F2 sum[2]~sum[3],j為輸出;x[2]~x[3],y[2]~y[3],ci為輸入
40 endmodule //4-bits全加器模組結束

42 module F2(s,co,a,b,ci); //2-bits全加器模組宣告
43 input [1:0]a,b; //輸入為2-bits共四個加數a[0],a[1],b[0],b[1]
44 input ci; //另一個輸入為進位值ci
45 output [1:0]s; //輸出為2-bits本位和s[0],[1]
46 output co; //另一個輸出為高位進位co
47 wire j; //2-bits全加器的邏輯電路圖中會用到一條線連接不同的模組
48 F1 z1(s[0],j,a[0],b[0],ci); //使用1-bit全加器的模組F1 s[0],j為輸出;a[0],b[0],ci為輸入
49 F1 z2(s[1],co,a[1],b[1],j); //再使用一次1-bit全加器的模組F1 s[1],co為輸出;a[1],b[1],j為輸入
50 endmodule //2-bits全加器模組結束

52 module F1(s,cout,a,b,cin); //1-bit全加器模組宣告
53 input a,b,cin; //輸入為兩個加數a b和進位值cin
54 output s,cout; //輸出為本位和s和高位進位cout
55 wire g,h,i; //1-bit全加器的邏輯電路圖中會用到三條線連接不同的邏輯閘
56 xor x1(g,a,b); //邏輯閘XOR · 輸入為a b；輸出為g
57 and a1(h,a,b); //邏輯閘AND · 輸入為a b；輸出為h
58 xor x2(s,g,cin); //邏輯閘XOR · 輸入為g cin；輸出為s
59 and a2(i,g,cin); //邏輯閘AND · 輸入為g cin；輸出為i
60 xor x3(cout,i,h); //邏輯閘XOR · 輸入為i h；輸出為cout
61 endmodule //1-bit全加器模組結束
```

7. 心得：

404415073 蔡孟勳

這次實驗是由我負責打程式，但由於當天我有報名電機通訊軟體競賽，因此我有在宿舍事先打了這次實驗要用的程式，然後到教室確定 compile 通過後，直接燒錄並模擬。幸好這次實驗還算簡單，而且可以使用到 exp2 完成的 4-bits 全加器，因此整體也變得好理解、也容易轉成程式語言，但也還滿對不起我隊友的，提早結束後就讓他自己一個人打註解…。