1.

	1’s complement	2’s complement
10100111	01011000	01011001
10101011	01010100	01010101
01110011	10001100	10001101
11100101	00011010	00011011
01010101	10101010	10101011

2.

(a)

0101 + 1110 = 0011 => $3_{(10)}$

-8 < 3 < 7, no overflow

(b)

0111110 + 1101011 = 0101001 => $41_{(10)}$

-64 < 41 < 63, no overflow

3.

(a)

0101 - 0110 => 0101 + 1010 = 1111 => $-1_{(10)}$

(b)

10110 – 1100 = 10110 - 01100 => 22 + (-12), 轉 2’s complement

= 10110 + 10100 = 01010 => $10_{(10)}$

(c)

1011110 – 1111110 => 94+(- 126), 轉 2’s complement

= 1011110 + 0000010 = 1100000 => $-32_{(10)}$

(d)

101010 – 101 = 101010 - 000101 => 42 + (-5), 轉 2’s complement

= 101010 + 111011 = 100101 => $37_{(10)}$

4.

(a)

0101 - 0110 => 0101 + 1010 = 1111 => $-1_{(10)}$,

no overflow(-8 < -1 < 7)

(b)

10110 – 1100 = 10110 - 11100 => -10 – (-4), 轉 2’s complement

=> 10110 + 00100 = 11010 => $-6_{(10)}$,

no overflow(-16 < -6 < 15)

(c)

1011110 – 1111110 => -30 – (+2), 轉 2's complement

=> 1011110 + 0000010 = 1100000 => $-32_{(10)}$,

no overflow(-64 < -32 < 63)

(d)

101010 – 101 = 101010 – 111101 => -22 – (-3), 轉 2's complement

=> 101010 + 000011 = 101101 => $-19_{(10)}$,

no overflow(-32 < -19 < 31)

5.

(a)

0101 - 0110 => 0101 + 1010 = 1111 => $-1_{(10)}$,

no overflow(-8 < -1 < 7)

(b)

10110 – 1100 => -6 – (-4), 轉 2's complement

=> 1010 + 0100 = 1110 = $-2_{(10)}$,

no overflow(-8 < -2 < 7)

(c)

1011110 – 1111110 => -30 – (-62), 轉 2's complement

=> 1100010 + 0111110 = 0100000 = $32_{(10)}$,

no overflow(-64 < -32 < 63)

(d)

101010 – 101 => -10 – (-1), 轉 2's complement

=> 10110 + 00001 = 10111 => $-9_{(10)}$,

no overflow(-16 < -9 < 15)

6.

(+27) – (50) => 0011011 – 0110010 => 0011011 + 1001110 = 1101001 => $-23_{(10)}$

(-42) – (30) => 11010110 – 00011110 => 11010110 + 11100010 = 10111000 => $-72_{(10)}$

7.

Sol 1

先畫出真值表

z3	z2	z1	z0	Z3	Z2	Z1	Z0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	x	x	x	x
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

由真值表畫出 K-amp，並求出 logic function。

Z3 = 0

Z2 = (z3'z2 +z3z2' + z2z1'z0') or (z3'z2 +z3z2' + z3z1'z0')

z1z0\z3z2	00	01	11	10
00		1	1	x
01		1		1
11		1		1
10		1		1

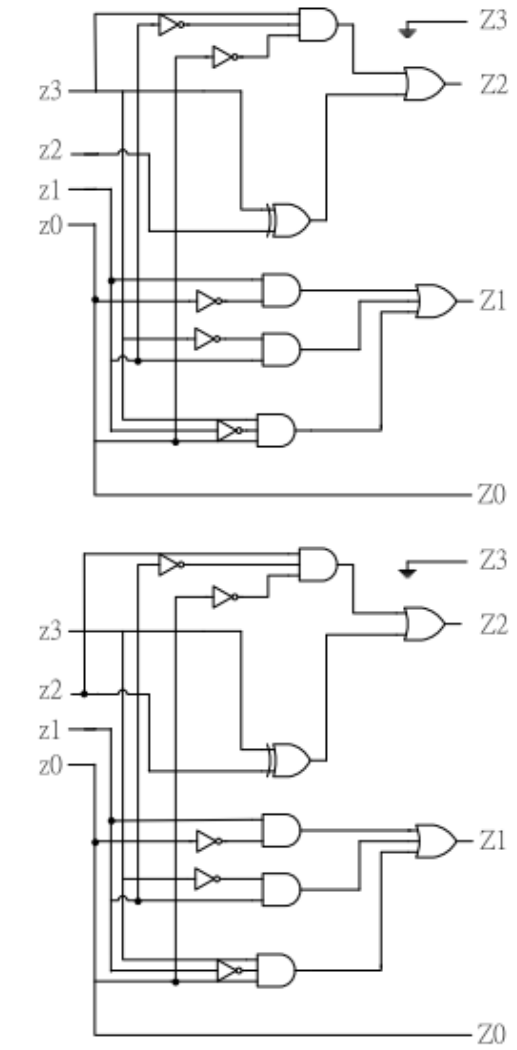
Z1 = z1z0' +z3'z1 +z3z1'z0

z1z0\z3z2	00	01	11	10
00				x
01			1	1
11	1	1		
10	1	1	1	1

Z0 = z0

z1z0\z3z2	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

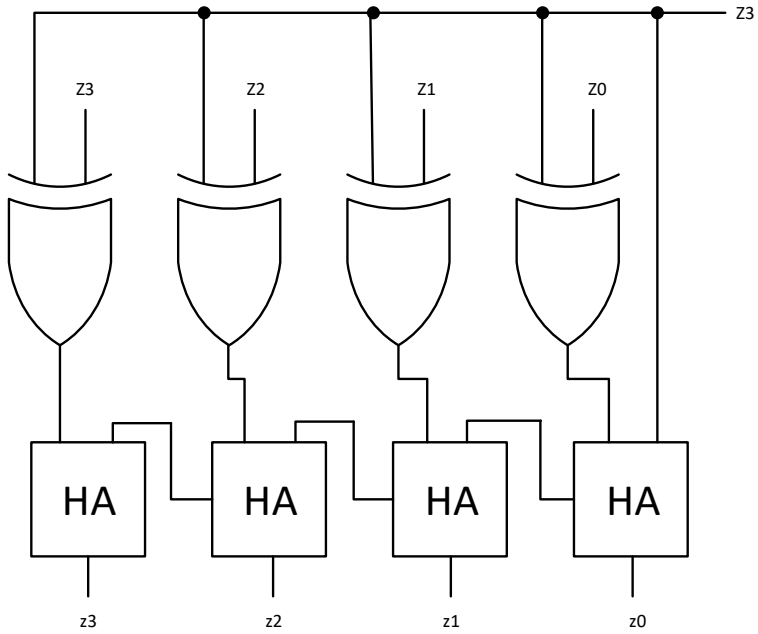
畫出 logic diagram



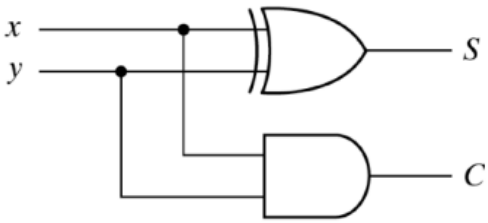
ps. 1000 為錯誤的輸入，若輸入 1000 結果會是 error

sol2.

Z=|z|



Half adder

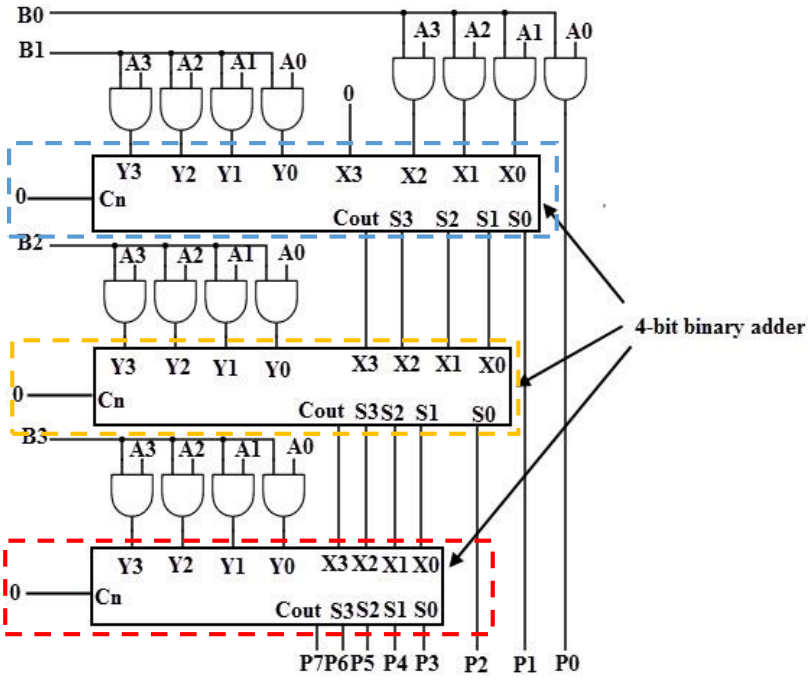


ps. 1000 為錯誤的輸入，若輸入 1000 結果會是 error

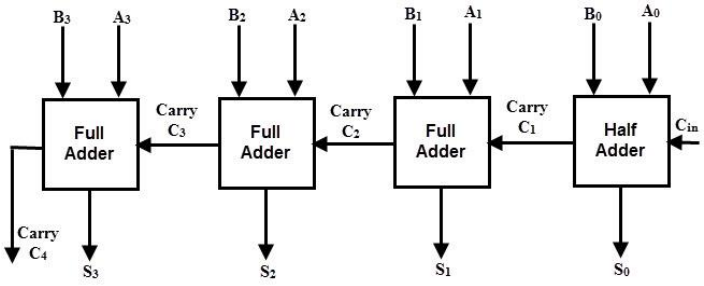
8.

4-bit x 4-bit multiplier (unsigned)

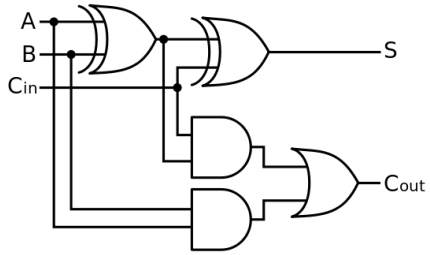
$$
\begin{array}{rcccccccc}
 & & & & A3 & A2 & A1 & A0 \\
\times & & & & B3 & B2 & B1 & B0 \\
\hline
 & & & & A3B0 & A2B0 & A1B0 & A0B0 \\
 & & & A3B1 & A2B1 & A1B1 & A0B1 & \\
 & & A3B2 & A2B2 & A1B2 & A0B2 & & \\
+ & A3B3 & A2B3 & A1B3 & A0B3 & & & \\
\hline
P7 & P6 & P5 & P4 & P3 & P2 & P1 & P0
\end{array}
$$



4-bit binary adder

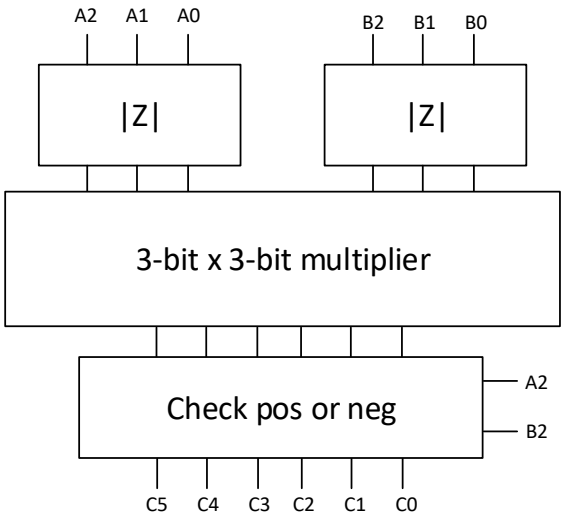


Full adder



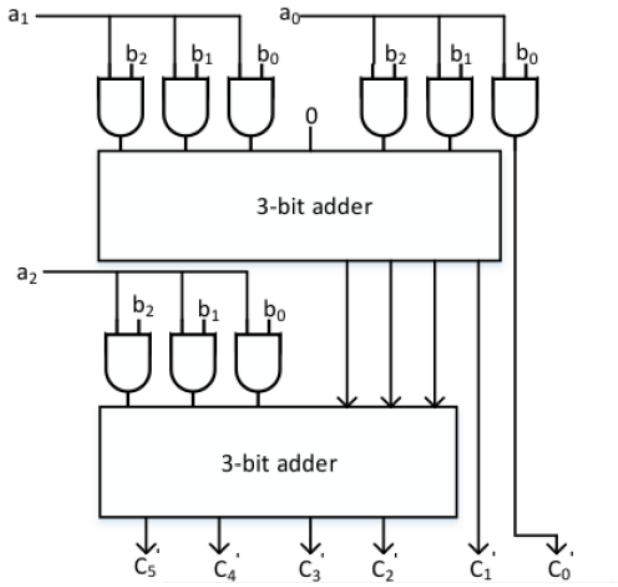
9.

Block diagram

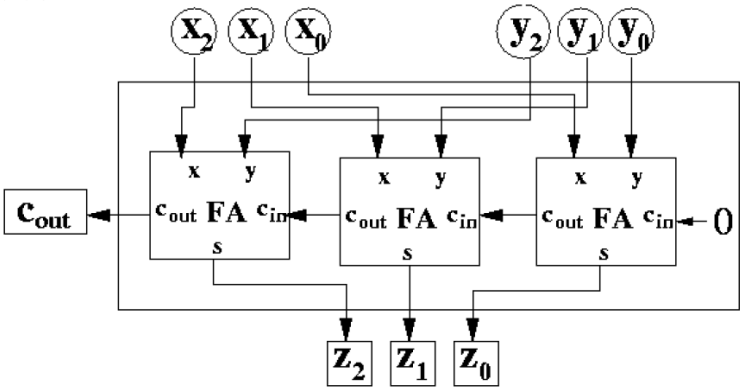


|Z|(取絕對值)的 logic function 可參考第 7 題

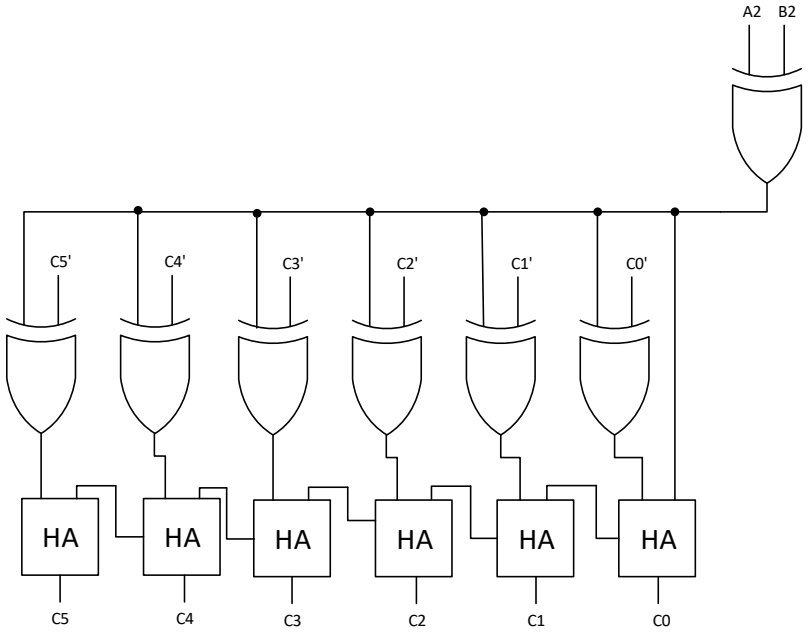
3-bit x 3bit multiplier



其中 3-bit adder



Check pos or neg



利用 A2 和 B2 判斷乘完的結果是否要變號

10.

Design

```
module absolute_value_calculator(Z,z);
        input [3:0]Z;
        output [3:0]z;

        assign z = (Z[3])?(~Z+Z[3]):(Z);
endmodule
```

Testbench

```
module test;
reg    [3:0]Z;
wire [3:0]z;

absolute_value_calculator avc(.Z(Z),.z(z));
initial begin
Z=4'b0000;
#5 Z=4'b0001;
#5 Z=4'b0010;
#5 Z=4'b0011;
#5 Z=4'b0100;
#5 Z=4'b0101;
#5 Z=4'b0110;
#5 Z=4'b0111;
#5 Z=4'b1001;
#5 Z=4'b1010;
#5 Z=4'b1011;
#5 Z=4'b1100;
#5 Z=4'b1101;
#5 Z=4'b1110;
#5 Z=4'b1111;
end
endmodule
```