

(1) Emulate a full adder:

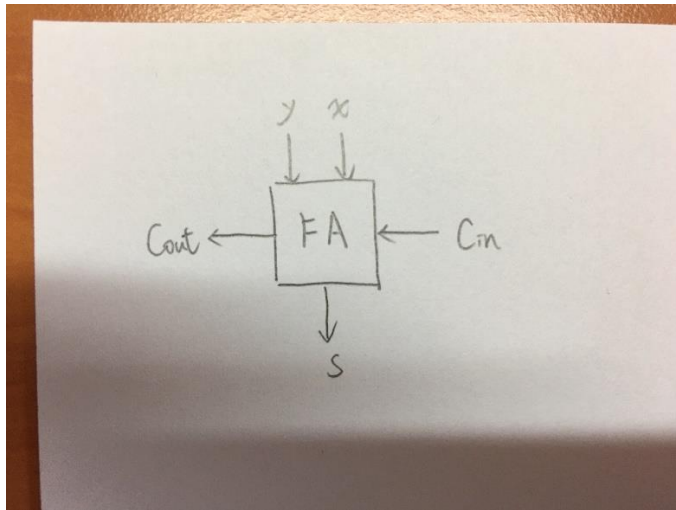
IO:

For a one bit full adder:

Input: x, y, cin

Output: s, cout

Block diagram:

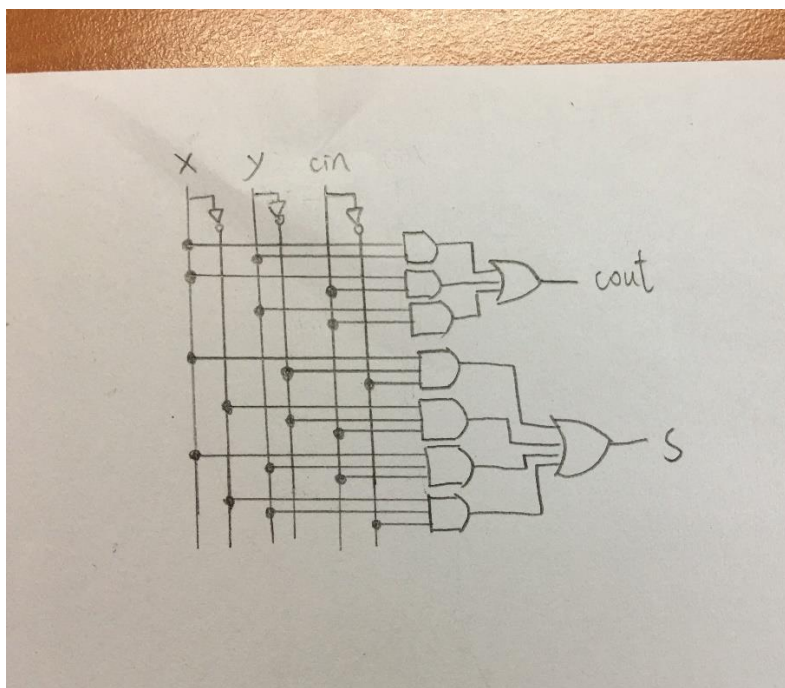


Logic function:

$$\text{cout} = x y + x \text{cin} + y \text{cin}$$

$$s = x y' \text{cin}' + x' y' \text{cin} + x y \text{cin} + x' y \text{cin}'$$

Logic diagram:



I/O pin assignment:

x-V17

y-V16

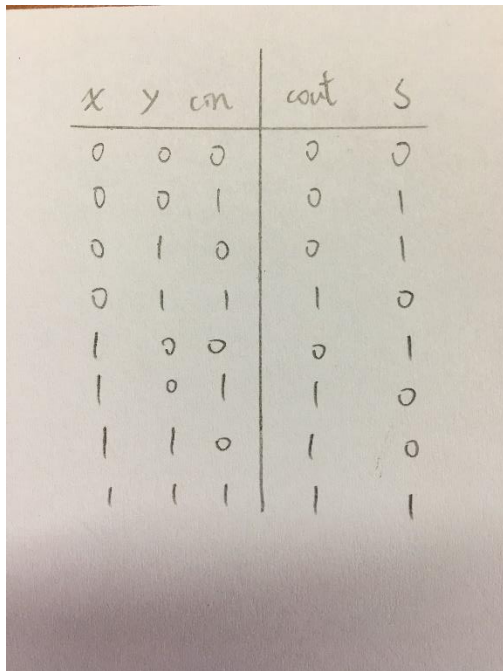
cin-W16

s-U16

cout-E19

Discussion:

這個實驗是要建構一個包含進位(carry in、carry out)的加法器。



x	y	cin	cout	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

透過 truth table 可以得知輸入(x、y、cin)與輸出(cout、s)之間的關係；也就是說，當輸入為奇數個 1 時會產生 s，而當輸入超過兩個 1 時，會產生 cout。在模擬圖中，將輸入值的所有可能性跑過一次，可以驗證加法器產生的結果，最高為 3，最低為 0。

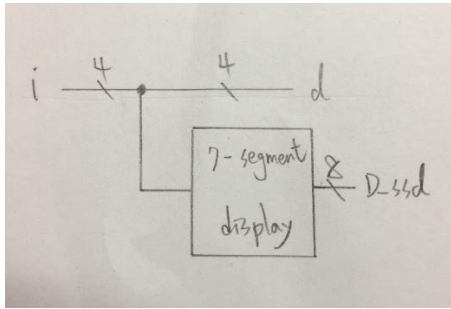
(2) BCD to 7-segment display decoder:

IO:

Input: [3:0]i

Output: [7:0]D_ssd, [3:0]d

Block diagram:



Logic function:

$$d[3] = i[3]$$

$$d[2] = i[2]$$

$$d[1] = i[1]$$

$$d[0] = i[0]$$

$$D_ssd[7] = \Sigma (1,4) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2] i[1]' i[0]'$$

$$D_ssd[6] = \Sigma (5,6,10 \sim 15) = i[3]' i[2] i[1]' i[0] + i[3]' i[2] i[1] i[0]' + i[3] i[2]' i[1] i[0]' + i[3] i[2]' i[1] i[0] + i[3] i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0] + i[3] i[2] i[1] i[0]$$

$$D_ssd[5] = \Sigma (2,10 \sim 15) = i[3]' i[2]' i[1] i[0]' + i[3] i[2]' i[1] i[0]' + i[3] i[2]' i[1] i[0] + i[3] i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0] + i[3] i[2] i[1] i[0]$$

$$D_ssd[4] = \Sigma (1,4,7,10 \sim 15) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2] i[1]' i[0]' + i[3]' i[2] i[1] i[0] + i[3] i[2]' i[1] i[0]' + i[3] i[2]' i[1] i[0] + i[3] i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0] + i[3] i[2] i[1] i[0]$$

$$D_ssd[3] = \Sigma (1,3,4,5,7,9) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2]' i[1] i[0] + i[3]' i[2] i[1]' i[0]' + i[3]' i[2] i[1] i[0] + i[3] i[2]' i[1]' i[0]$$

$$D_ssd[2] = \Sigma (1,2,3,7) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2]' i[1] i[0]' + i[3]' i[2]' i[1] i[0] + i[3]' i[2] i[1]' i[0]$$

$$D_ssd[1] = \Sigma (0,1,7) = i[3]' i[2]' i[1]' i[0]' + i[3]' i[2]' i[1]' i[0] + i[3]' i[2] i[1] i[0]$$

$$D_ssd[0] = \Sigma (0,1,2,3,4,5,6,7,8,9,10 \sim 15) = \text{logic-1}$$

I/O pin assignment:

$$i[0] \rightarrow V17$$

$$i[1] \rightarrow V16$$

$$i[2] \rightarrow W16$$

$$i[3] \rightarrow W17$$

$$D_ssd[0] \rightarrow V7$$

$$D_ssd[1] \rightarrow U7$$

$$D_ssd[2] \rightarrow V5$$

D_ssd[3]→U5
D_ssd[4]→V8
D_ssd[5]→U8
D_ssd[6]→W6
D_ssd[7]→W7
d[0]→U16
d[1]→E19
d[2]→U19
d[3]→V19

Discussion:

本次實驗要將 4bit 的 BCD 輸入值(i)以七段顯示器顯示 0 到 9 的數字出來，七段顯示器的接腳是規定好的，所以只要按照講義上的接法就可以順利接好，而要特別小心的地方是七段顯示器是 low active。另外，輸出(d)的作用是在顯示出輸入(i)的值，這樣可以讓進行實測時更加清楚當下的輸入情形。D_ssd 是 8bit 的輸出，代表的是七段顯示器上的每一小段以及一個小數點，透過 truth table 可以得知當 i 在 0 到 9 之間變化時七段顯示器應該要如何編碼；此外，當 $i \geq 10$ 時，七段顯示器要顯示 F(0111_0001)。

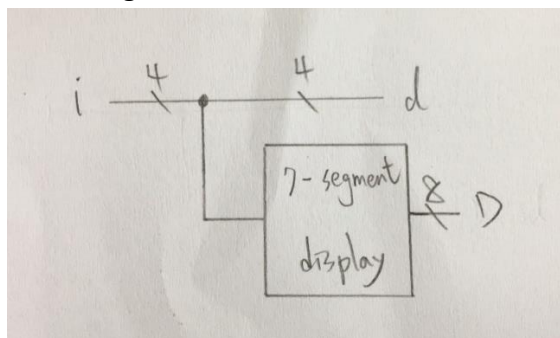
(3) binary to 7-segment display decoder:

IO:

input [3:0]i

output [7:0]D, [3:0]d

Block diagram:



Logic function:

$d[3] = i[3]$

$d[2] = i[2]$

$d[1] = i[1]$

$d[0] = i[0]$

$$\begin{aligned}
D[7] &= \Sigma(1,4,11,13) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2] i[1]' i[0]' + i[3] i[2]' i[1] i[0] \\
&\quad + i[3] i[2] i[1]' i[0] \\
D[6] &= \Sigma(5,6,11,12,14,15) = i[3]' i[2] i[1]' i[0] + i[3]' i[2] i[1] i[0]' + i[3] i[2]' i[1] i[0] \\
&\quad + i[3] i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0]' + i[3] i[2] i[1] i[0] \\
D[5] &= \Sigma(2,12,14,15) = i[3]' i[2]' i[1] i[0]' + i[3] i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0]' \\
&\quad + i[3] i[2] i[1] i[0] \\
D[4] &= \Sigma(1,4,7,10,15) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2] i[1]' i[0]' + i[3] i[2] i[1] i[0] \\
&\quad + i[3] i[2]' i[1] i[0]' + i[3] i[2] i[1] i[0] \\
D[3] &= \Sigma(1,3,4,5,7,9) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2]' i[1] i[0]' + i[3] i[2] i[1]' i[0]' \\
&\quad + i[3] i[2] i[1]' i[0] + i[3] i[2] i[1] i[0]' + i[3] i[2]' i[1]' i[0] \\
D[2] &= \Sigma(1,2,3,7,13) = i[3]' i[2]' i[1]' i[0] + i[3]' i[2]' i[1] i[0]' + i[3] i[2]' i[1] i[0] \\
&\quad + i[3] i[2] i[1] i[0]' + i[3] i[2] i[1]' i[0] \\
D[1] &= \Sigma(0,1,7,12) = i[3]' i[2]' i[1]' i[0]' + i[3]' i[2]' i[1]' i[0] + i[3] i[2] i[1] i[0] \\
&\quad + i[3] i[2] i[1]' i[0]' \\
D[0] &= \Sigma(0,1,2,3,4,5,6,7,8,9,10 \sim 15) = \text{logic-1}
\end{aligned}$$

I/O pin assignment:

$i[0] \rightarrow V17$

$i[1] \rightarrow V16$

$i[2] \rightarrow W16$

$i[3] \rightarrow W17$

$D[0] \rightarrow V7$

$D[1] \rightarrow U7$

$D[2] \rightarrow V5$

$D[3] \rightarrow U5$

$D[4] \rightarrow V8$

$D[5] \rightarrow U8$

$D[6] \rightarrow W6$

$D[7] \rightarrow W7$

$d[0] \rightarrow U16$

$d[1] \rightarrow E19$

$d[2] \rightarrow U19$

$d[3] \rightarrow V19$

Discussion:

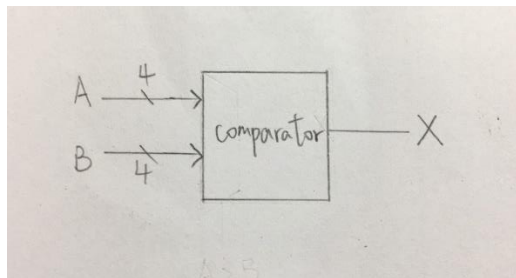
這個實驗跟上個小題的實驗相當類似，唯一不同的地方是當輸入值 $i \geq 10$ 時，七段顯示器要用 16 進位的方式顯示出數值(分別為 A,b,C,d,E,以及 F)，其他像是輸入值(i)以及 LED 輸出(d)都跟上一個小題完全相同。

(4) Design a combinational circuit that compares two 4-bit unsigned numbers A and B to see whether A is greater than B. The circuit has one output X such that $X = 0$ if $A \leq B$ and $X = 1$ if $A > B$.

IO:

input [3:0]A, [3:0]B,
output [3:0]Ain, [3:0]Bin, X

Block diagram:



Logic function:

$$X = A[3]B[3]' + (A[3] \oplus B[3])'A[2]B[2]' + (A[3] \oplus B[3])'(A[2] \oplus B[2])'A[1]B[1]' + (A[3] \oplus B[3])'(A[2] \oplus B[2])'(A[1] \oplus B[1])'A[0]B[0]'$$

I/O pin assignment:

A[3] → W13
A[2] → W14
A[1] → V15
A[0] → W15
Ain[3] → V14
Ain[2] → U14
Ain[1] → U15
Ain[0] → W18
B[3] → W17
B[2] → W16
B[1] → V16
B[0] → V17
Bin[3] → V19
Bin[2] → U19
Bin[1] → E19
Bin[0] → U16
X → L1

Discussion:

本次實驗目的在於將兩個 4bit 的 unsigned number 輸入值(A,B)比大小。當 $A > B$ 時，將輸出(X)指定為 1，當 $A \leq B$ 時，則將輸出(X)指定為 0。兩個 4bit 的輸出值 A_{in} 以及 B_{in} 則可以將輸入的值用 LED 顯示出來。將輸入和輸出都接上接腳後，即可驗證電路板上的顯示結果。

此外，上述 logic function 的邏輯為先比較最高位數字，若相同，則繼續往下一位數比，以此類推，當比到有一方較另一方大時，便可得知兩數的大小關係。

Conclusion:

這次實驗與以前最不同的地方就是接上了電路板，雖然跑電路板需要消耗一段時間，但是相比之下卻變得不像波形圖一樣那麼枯燥乏味，也有了實際操作的真實感。這次的題目並不複雜，是熟悉 FPGA 的良好機會，也終於看到上學期的理論基礎慢慢的實體化，做起來相當有成就感。