Lab1

107060017 廖凰汝

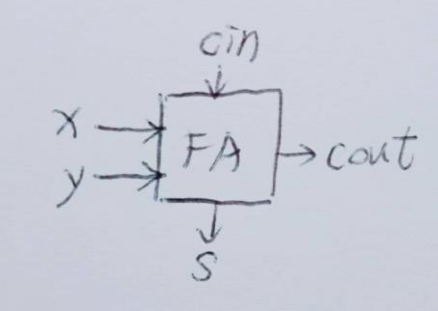
1.
Design Specification

✓ **Input / Output**

For a full adder(s+cout=x+y+cin):

Input: x, y, cin.
Output: s, cout.

✓ **Block Diagram**

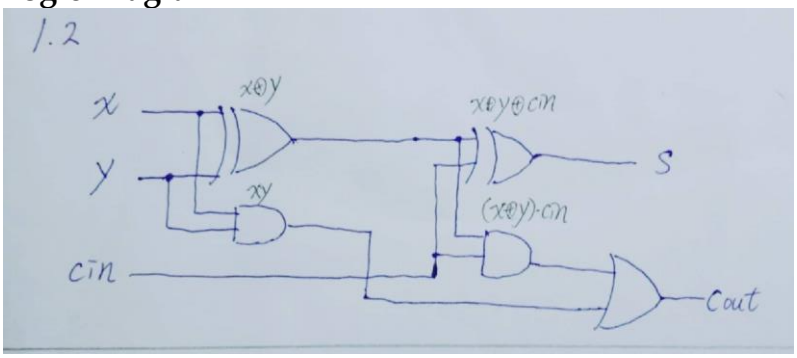


Design Implementation

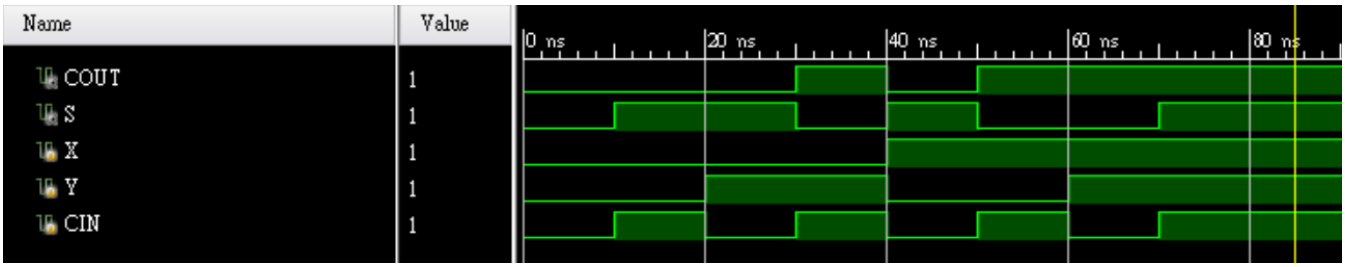
✓ **Logic Functions**

s = x ^ y ^cin
cout = (x ^ y) & cin | x & y

✓ **Logic Diagram**



✓ **Simulation**



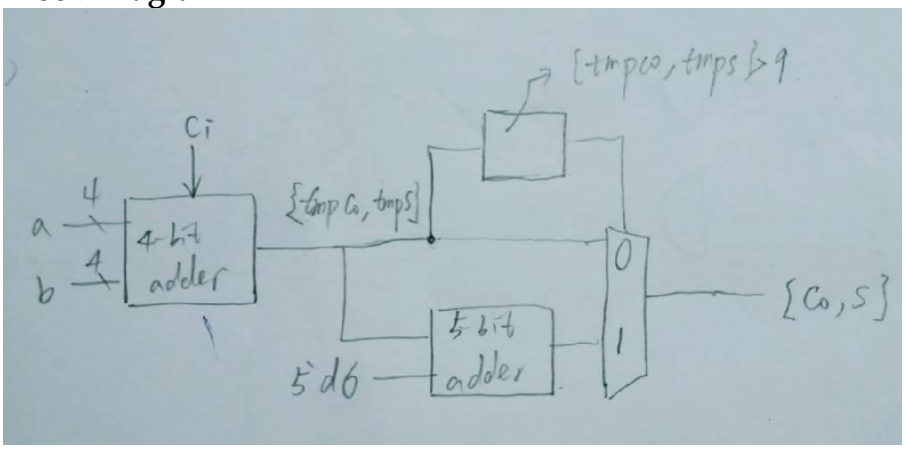
2.
Design Specification

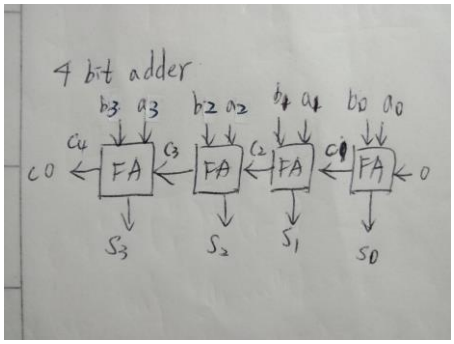
✓ **Input / Output**

For a single digit decimal adder:

Input: [3:0]a, [3:0]b, ci
Output: [3:0]s, co

✓ **Block Diagram**

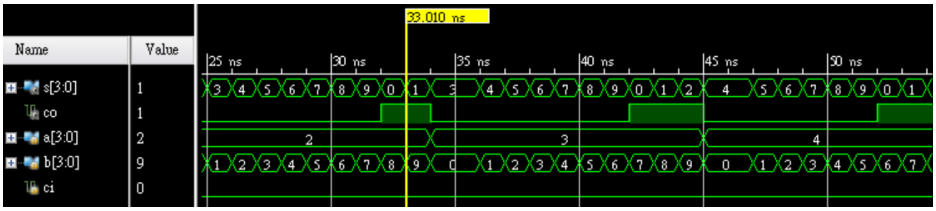


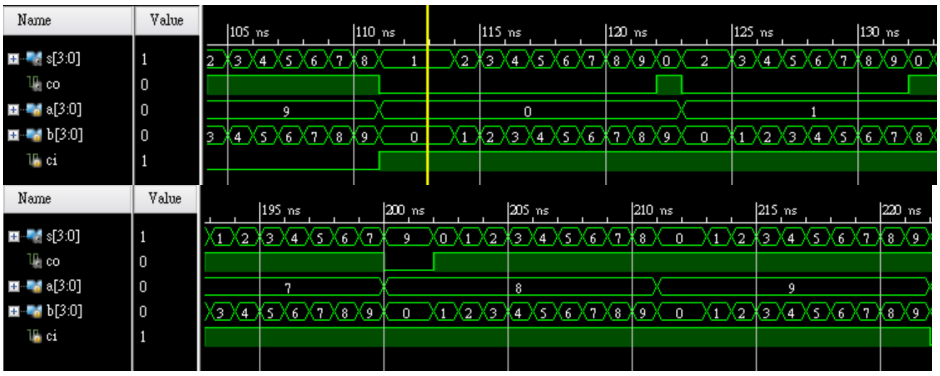


Design Implementation

✓ **Logic Functions**
if (a + b + ci)>9: co = 1, s = (a + b + ci) + 6
else: co = 0, s = (a + b + ci)

✓ **Simulation**





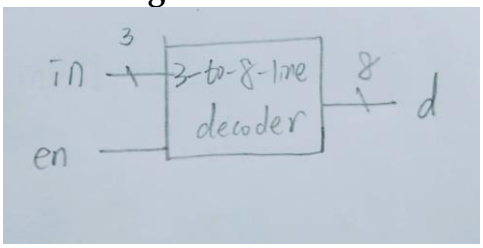
3. Design Specification

✓ **Input / Output**

For a 3-to-8-line decoder:

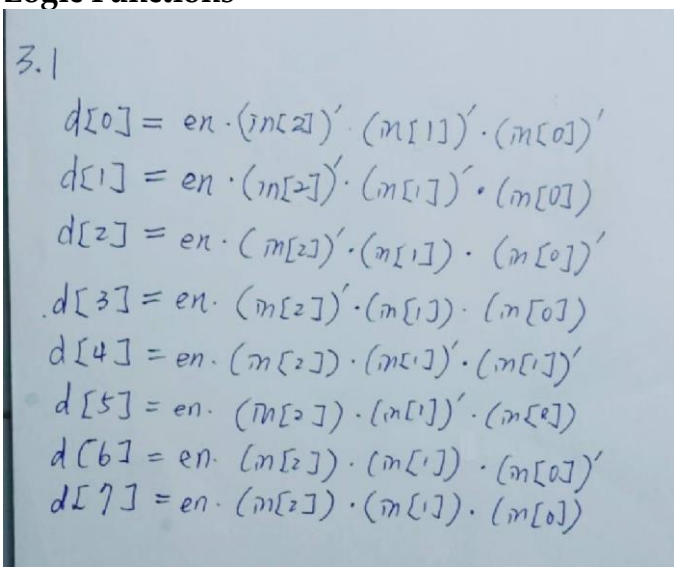
Input: [2:0]in, en
Output: [7:0]d

✓ **Block Diagram**

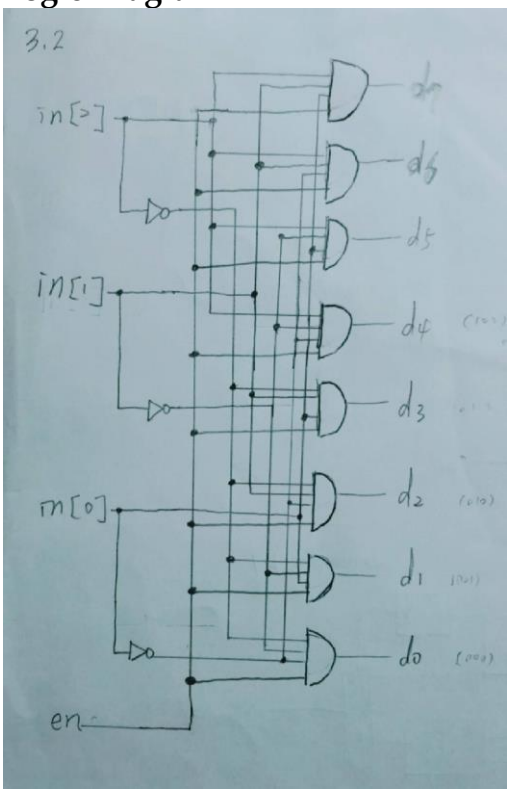


Design Implementation

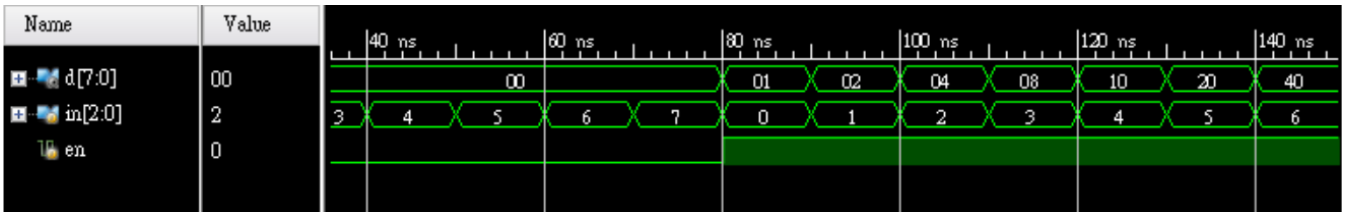
✓ **Logic Functions**



✓ **Logic Diagram**



✓ **Simulation**



Discussion

✓ Record the thoughts you have throughout this experiment.

這次的實驗做了Full Adder, MUX, 4 and 5 bit adder和decoder, 馬上強迫我們喚回上學期的記憶! 在實驗的過程，我又翻了以前的講義，重新理解這些功能做出來的邏輯原理，寫成程式模擬出結果，感覺滿踏實的! 更棒的是，多bit加法器，多功器這類寫成logic function 都很麻煩，但用verilog 只要一個"+", "if" 就完成了!

✓ Explains why the results came out as you expected, or the things went wrong if they **did** go wrong.

利用block diagram 再寫出正確的logic function，照理來說出來的結果就會符合我們的預期。但我這次出錯在第二小題的testbench，後來發現是我的測資間的間隔太大了，調小以後就能看到所有的測資了!

- ✓ Analyze and explain the experiment results, including any thoughts, comments, or suggestions related to this experiment.
  第一小題的Full adder與第三小題的decoder都像是一個大module裡可以用到的功能，或許以後要用到時可以直接將它加進其中的module中。例如第一題可以作成多bit adder，第二題可以作成mux。而這兩個功能也恰巧出現在第二題實作出來了

Conclusion

在這個實驗中，我學會了如何利用verilog將上學期教的Full Adder, MUX,多bit adder和decoder實作模擬出來，並用testbench將所有可能測資試過一遍，檢查自己的對這個功能的理解與實作有沒有錯。

另外，我體會到verilog的好用之處，例如多bit adder :一個” +”就省去了上學期還要開兩三個module的麻煩。