

# Lab01 Verilog HDL

107061112 王昊文

1. Design and implement a full adder. ( $s + \text{cout} = x + y + \text{cin}$ )

1.1. Write the logic equation.

I/O

Input: x, y, cin

Output: s, cout

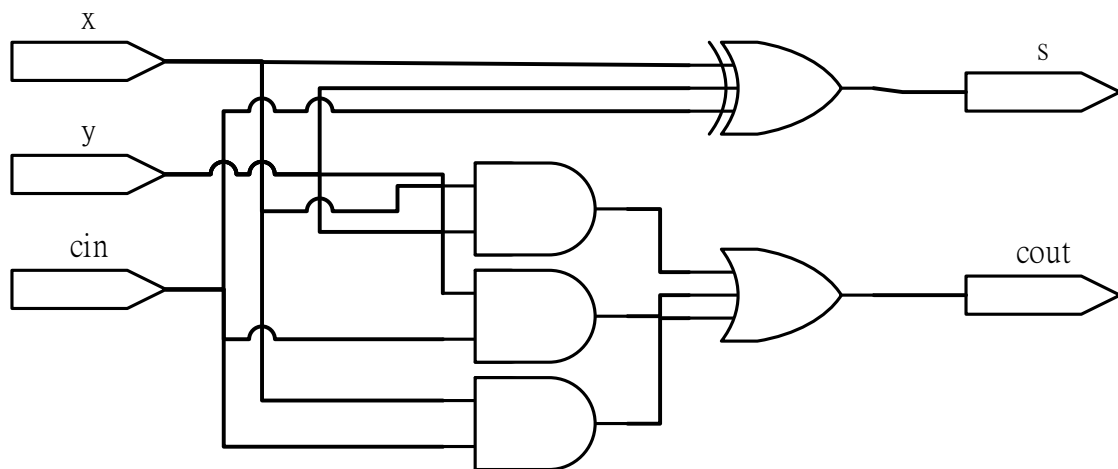
Obtain the truth table

Input			Output	
x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

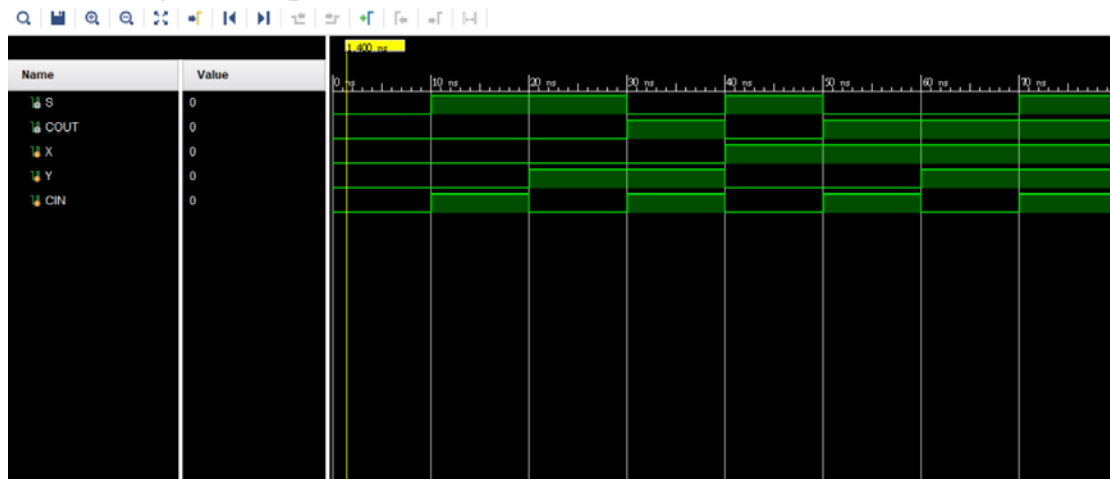
$$S = x \oplus y \oplus \text{cin}$$

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

1.2 Draw the related logic diagram.



### 1.3 Verilog RTL representation with verification.



#### Discussion:

這項實驗中，主要學習最基礎 Verilog 的語法，基本上只要完全按照語法打出邏輯基本上就沒有問題。

雖然是簡單的題目，但是在做的時候我忘記檢查，邏輯打錯沒有發現，在 Demo 的時候出包。經過這次，我學習到即使是簡單的題目也要好好檢查。

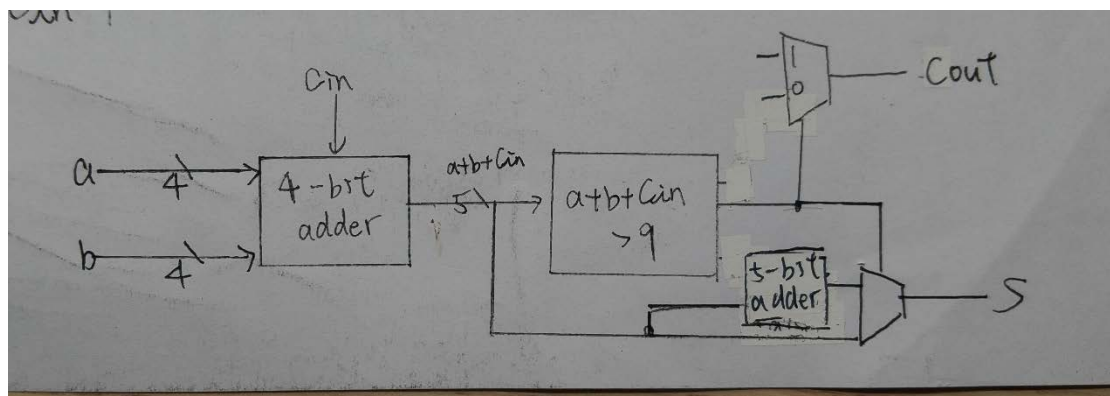
2 Design a single digit decimal adder with input  $A(a_3a_2a_1a_0)$ ,  $B(b_3b_2b_1b_0)$ ,  $C_{in}(ci)$ , and output  $S(s_3s_2s_1s_0)$  and  $C_{out}(co)$ .

I/O

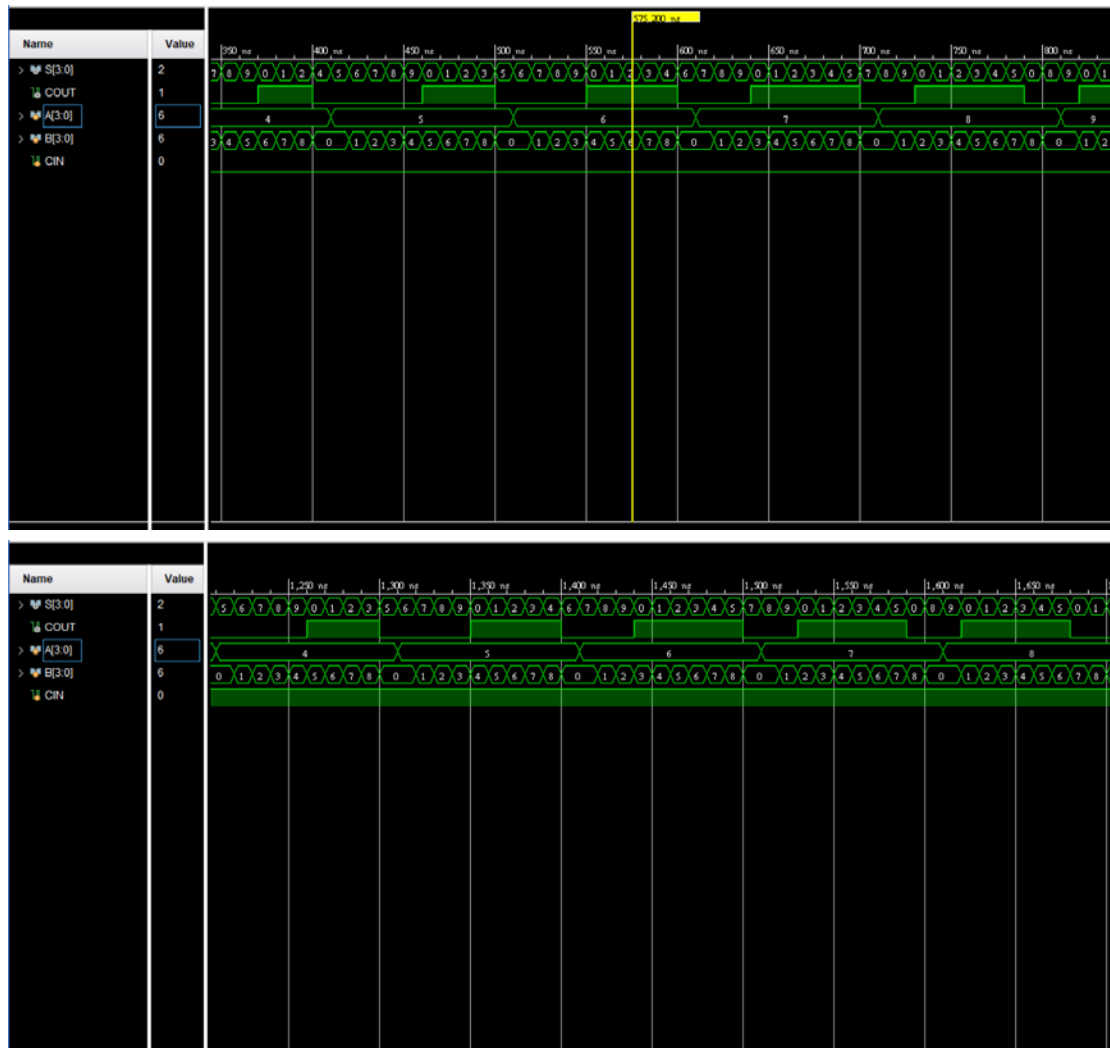
Output:  $s[3:0]$ ,  $cout$

Input:  $a[3:0]$ ,  $b[3:0]$ ,  $cin$

#### Logic Schematic



Waveform:



### Discussion:

基本上這一題的想法並不難，只是必須考慮 Verilog 是硬體描述語言，在寫 Code 的時候必須考慮硬體上的限制。必須先畫出 Logic Block 再來寫 Code，會比較符合現實。

3 (Bonus) Design a 3-to-8-line decoder with enable (input in[2:0], enable en and output d[7:0]).

I/O

Input: in[2:0], en

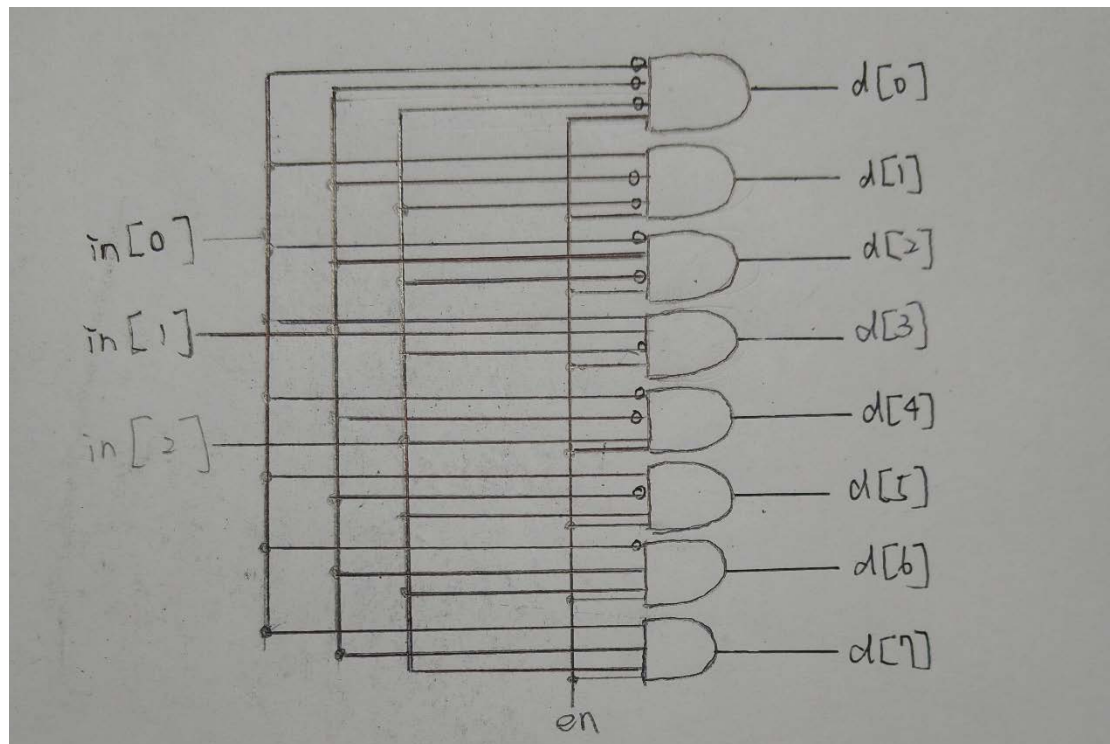
Output: d[7:0]

3.1 Logic equation,

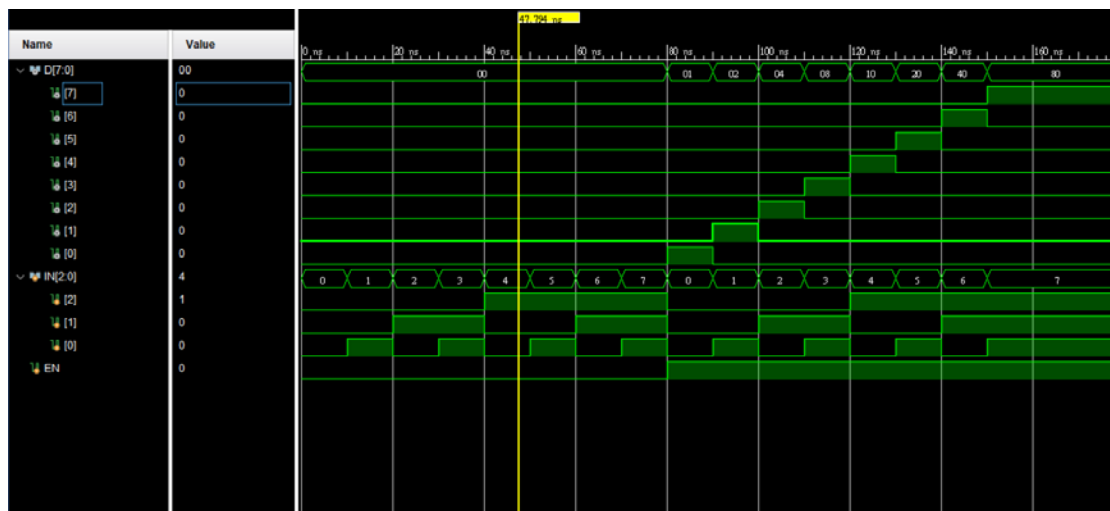
$$d[0] = \text{in}[2]' \& \text{in}[1]' \& \text{in}[0]' \& \text{en}$$

$d[1] = in[2]' \& in[1]' \& in[0] \& en$   
 $d[2] = in[2]' \& in[1] \& in[0]' \& en$   
 $d[3] = in[2]' \& in[1] \& in[0] \& en$   
 $d[4] = in[2] \& in[1]' \& in[0]' \& en$   
 $d[5] = in[2] \& in[1]' \& in[0] \& en$   
 $d[6] = in[2] \& in[1] \& in[0]' \& en$   
 $d[7] = in[2] \& in[1] \& in[0] \& en$

### 3.2 Logic schematic



### 3.3 Verilog RTL representation with verification.



Discussion:

這一題基本上十分簡單，將邏輯全部條列出來即可。但是我很好奇如果是一個很大的 Decoder 是否一樣需要全部列出？還是有更快的寫法。

## Conclusion:

第一次的 Lab 的難度並不高，但是上學期的 Verilog 結構並沒有好好學習。這學期算是重頭學起，第一個 Lab 給我一個練習的好機會。透過這次 lab，我把基本結構，Test bench 等等觀念都搞懂了，期待下次更有挑戰性的 Lab。