105061254 林士平 邏輯設計實驗報告 Lab1

1.

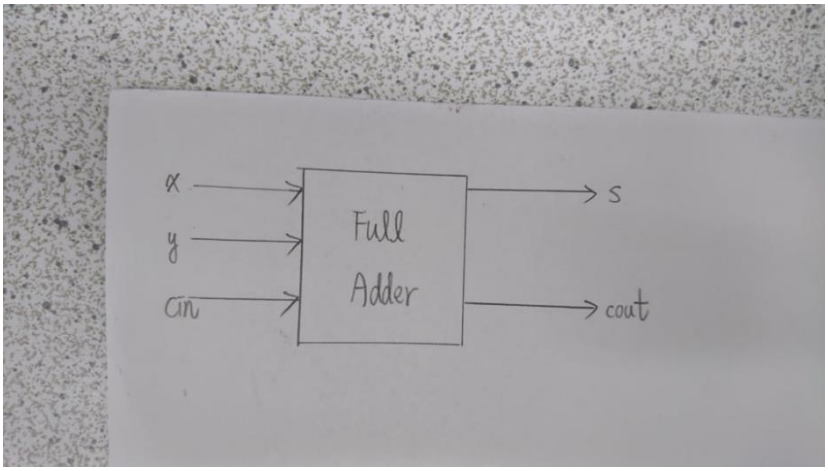
(1) Design specification :

A. Inputs and outputs(表一)：

Inputs	x, y, cin
Outputs	s, cout

↑表一：Inputs and outputs of 1.

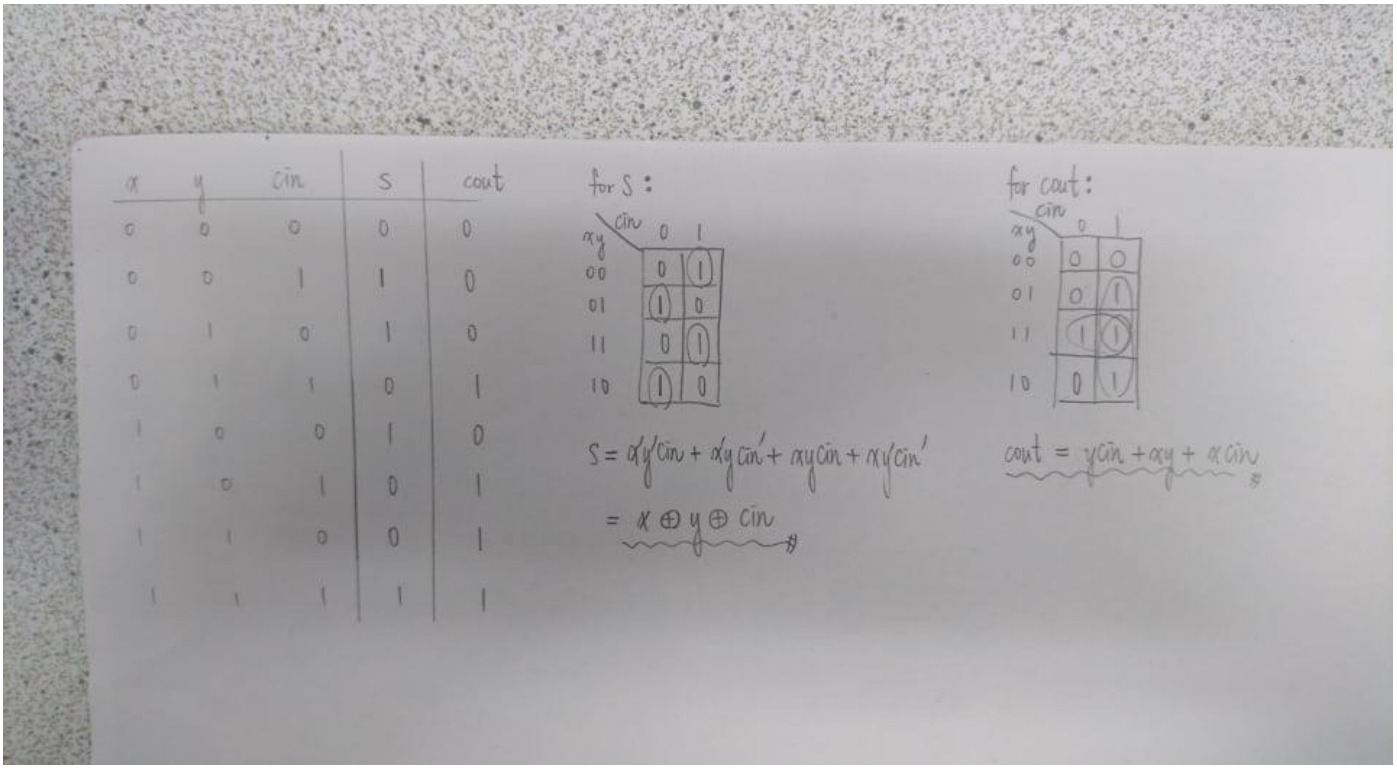
B. Block diagram(圖一)：



↑圖一：The block diagram of 1.

(2) Design implementation :

A. Logic function(圖二)：



↑圖二：logic funtion 計算過程

由(圖二)可知 logic function 為：

$$s = \mathrm{x} \oplus \mathrm{y} \oplus \mathrm{cin}$$

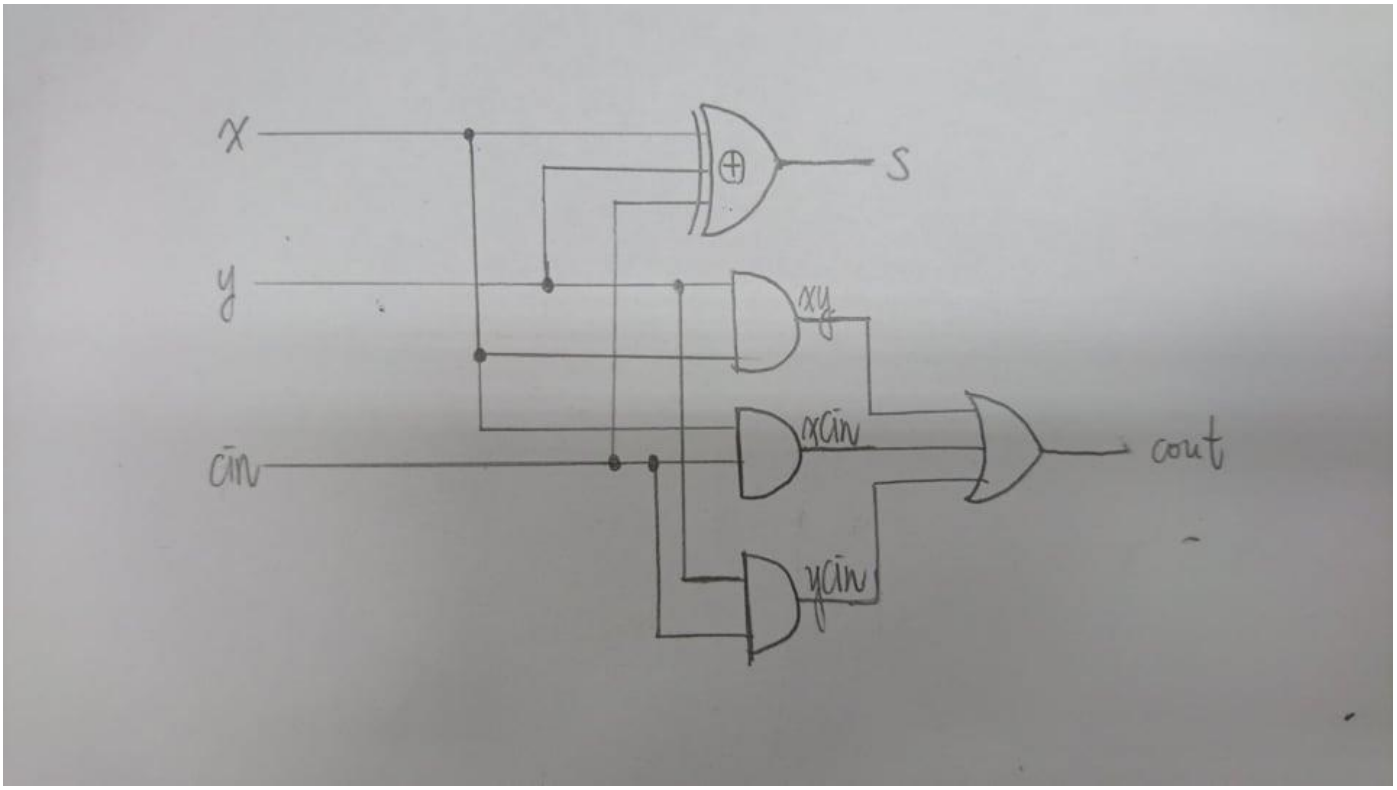
$$\mathrm{cout} = \mathrm{ycin} + \mathrm{xy} + \mathrm{xcin}$$

寫成 verilog 的形式如下：

$$s = x \ \hat{} \ y \ \hat{} \ cin$$

$$cout = (x \ \& \ y) \mid (x \ \& \ cin) \mid (y \ \& \ cin)$$

B. Logic diagram(圖三)：



↑圖三：logic diagram of 1.

2.

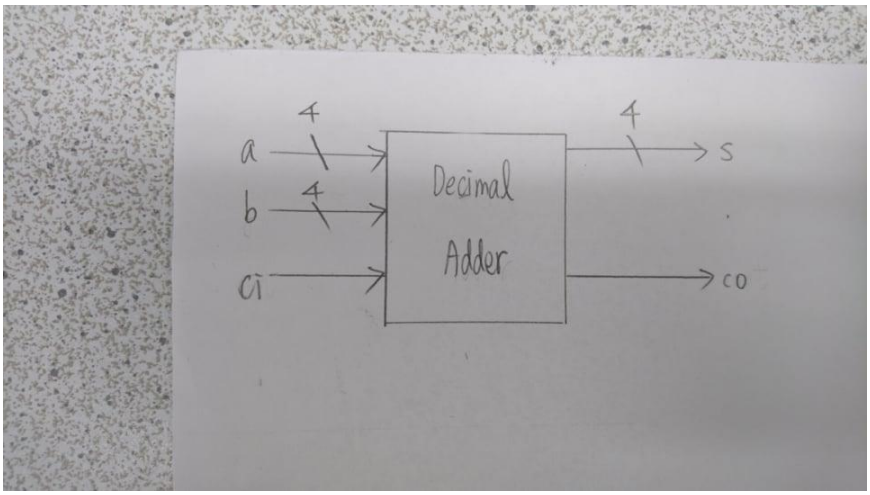
(1) Design specification :

A. Inputs and outputs(表二)：

Inputs	a[3:0], b[3:0], ci
Outputs	S[3:0], co

↑表二：Inputs and outputs of 2.

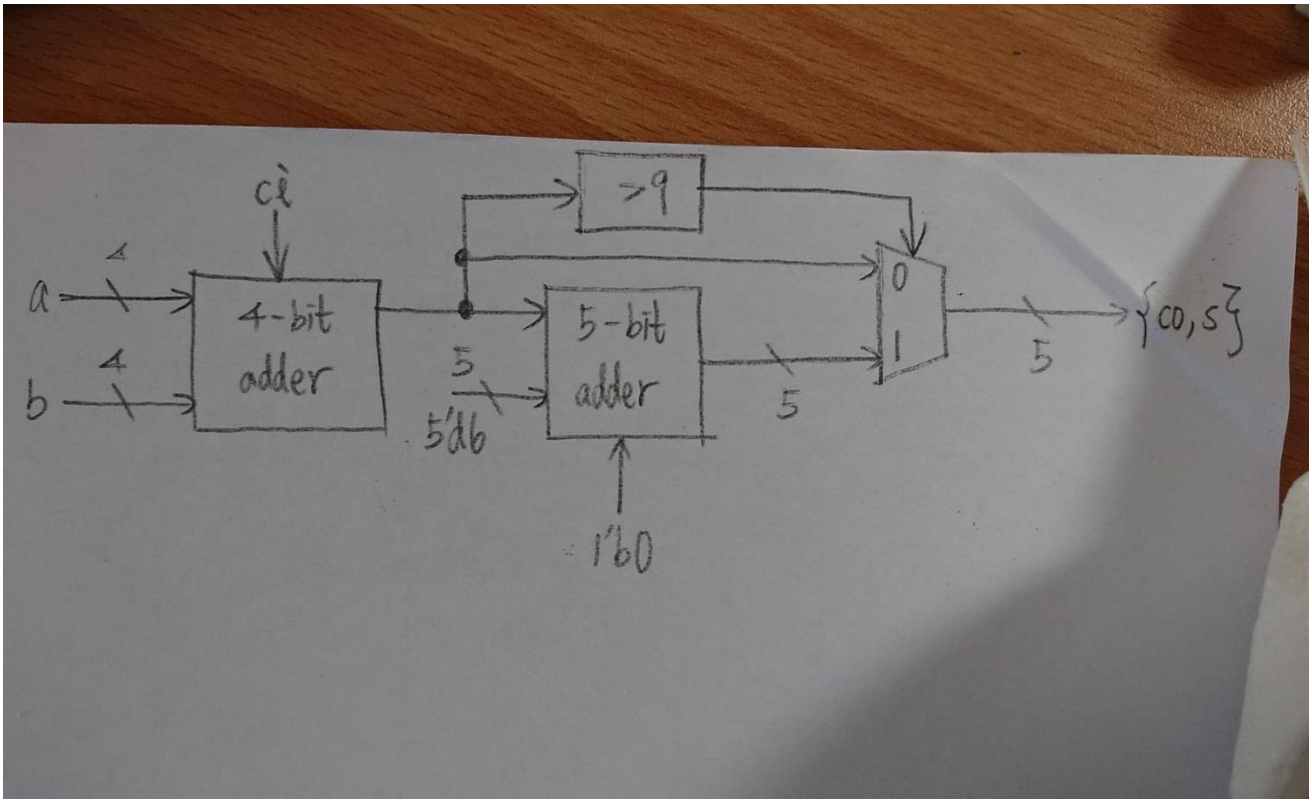
B. Block diagram(圖四)：



↑圖四：The block diagram of 2.

(2) Design implementation :

A. Logic diagram(圖五)：



↑圖五：logic diagram of 2.

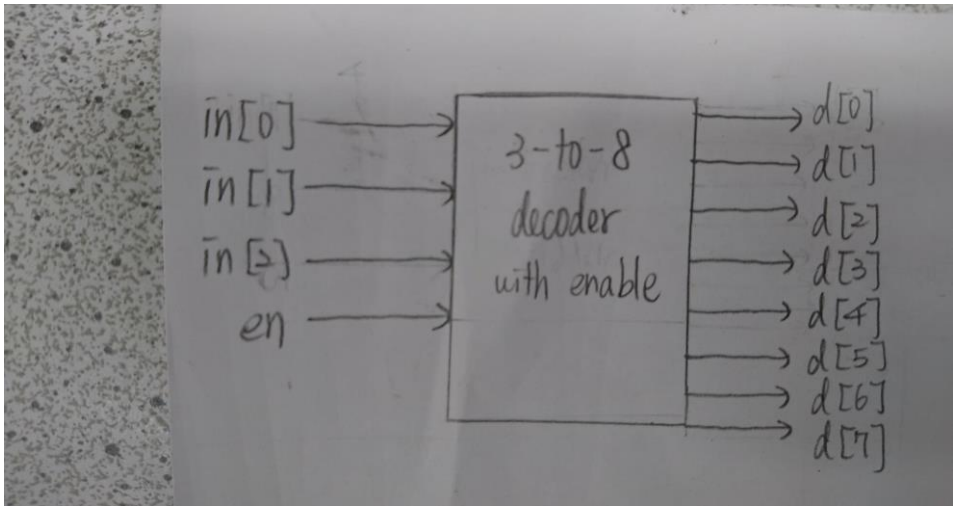
3.

(1) Design specification :

A. Inputs and outputs(表三)：

Inputs	In [2:0], en
Outputs	D [7:0]
↑表三：Inputs and outputs of 3.	

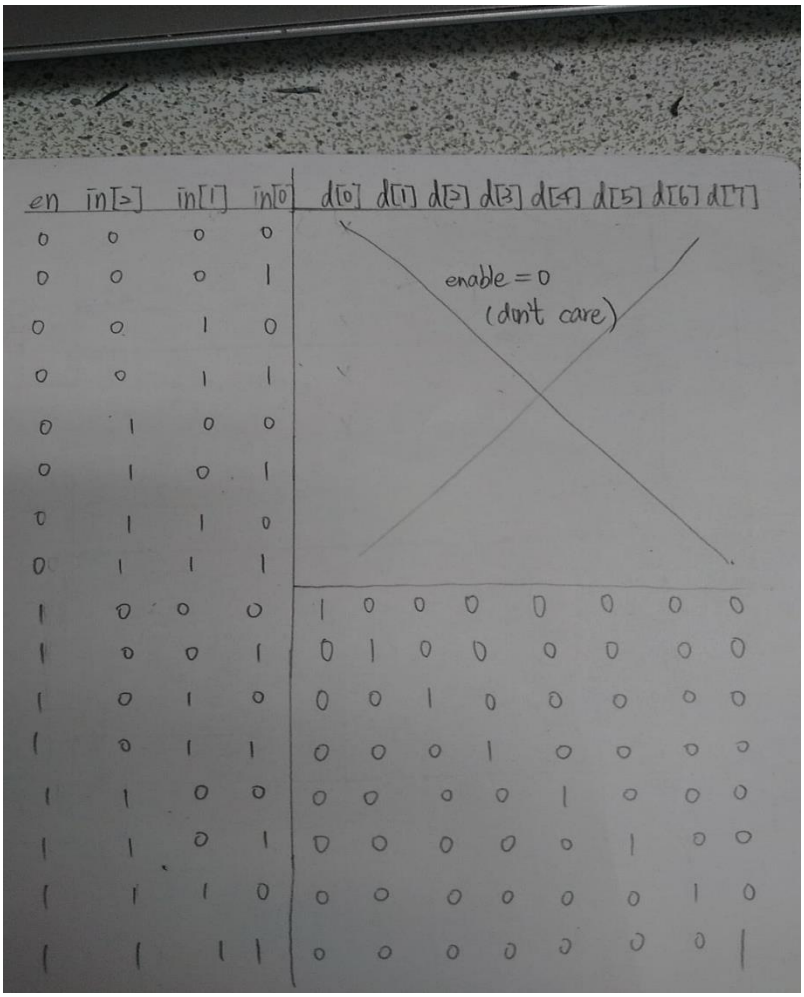
B. Block diagram(圖六)：



↑圖六：The block diagram of 3.

(2) Design implementation :

A. Logic function(圖七)：



↑圖七：truth table

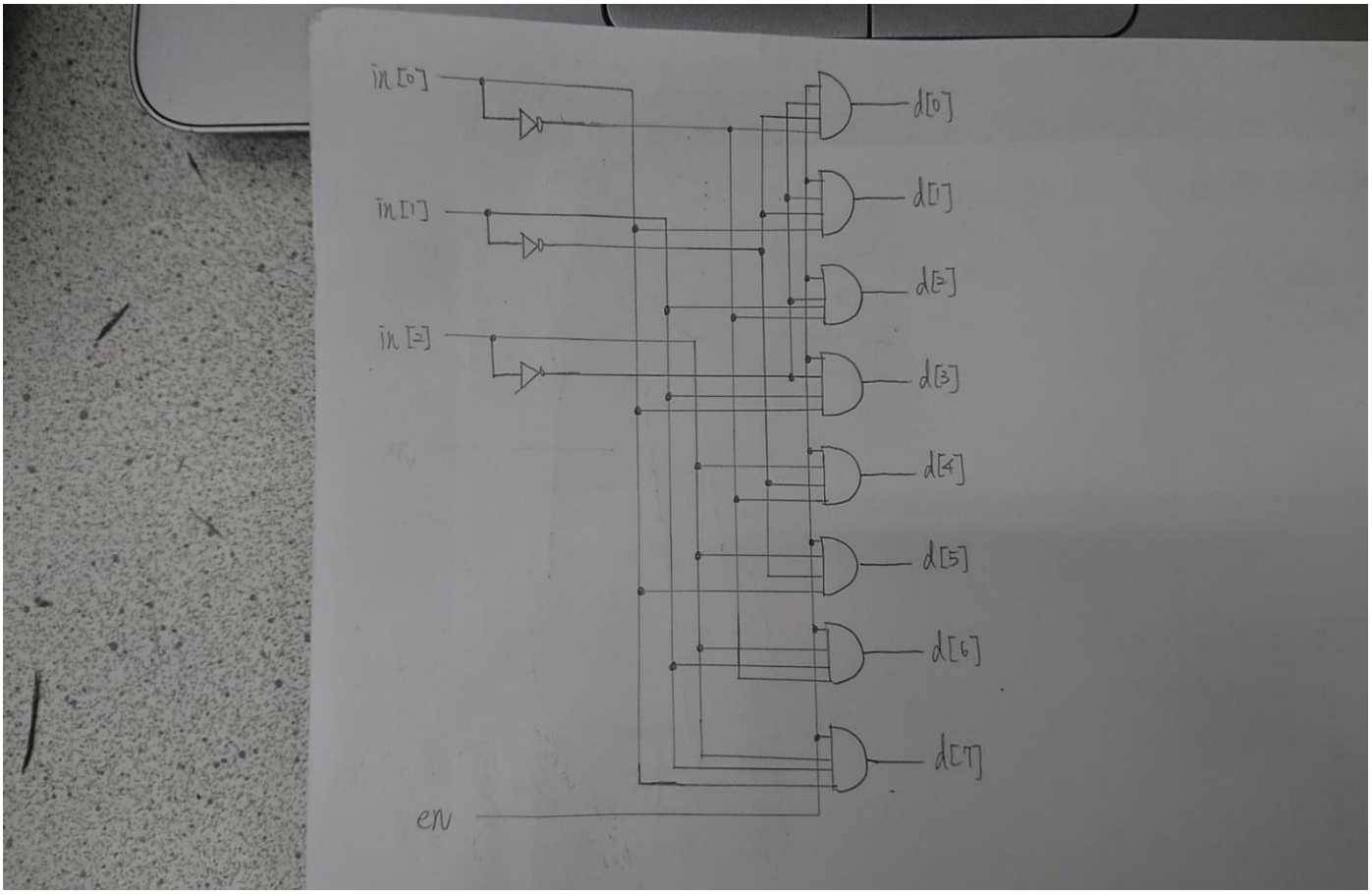
由(圖七)可知 logic function 為：

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

寫成 verilog 的形式如下：

$$d[0] = (\sim in[0])\ \&\ (\sim in[1])\ \&\ (\sim in[2])\ \&\ en$$
$$d[1] = in[0]\ \&\ (\sim in[1])\ \&\ (\sim in[2])\ \&\ en$$
$$\mathrm{d[2]} = (\sim \mathrm{in[0]})\ \&\ \mathrm{in[1]}\ \&\ (\sim \mathrm{in[2]})\ \&\ \mathrm{en}$$
$$\mathrm{d[3]} = \mathrm{in[0]}\ \&\ \mathrm{in[1]}\ \&\ (\sim \mathrm{in[2]})\ \&\ \mathrm{en}$$
$$\mathrm{d[4]} = (\sim \mathrm{in[0]})\ \&\ (\sim \mathrm{in[1]})\ \&\ \mathrm{in[2]}\ \&\ \mathrm{en}$$
$$d[5] = in[0]\ \&\ (\sim in[1])\ \&\ in[2]\ \&\ en$$
$$\mathrm{d[6]} = (\sim \mathrm{in[0]})\ \&\ \mathrm{in[1]}\ \&\ \mathrm{in[2]}\ \&\ \mathrm{en}$$
$$\mathrm{d[7]} = \mathrm{in[0]}\ \&\ \mathrm{in[1]}\ \&\ \mathrm{in[2]}\ \&\ \mathrm{en}$$

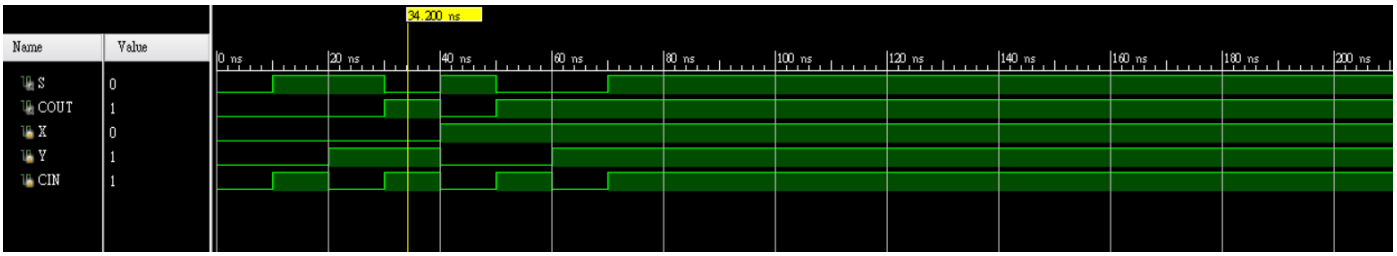
B. Logic diagram(圖八)：



↑圖八：logic diagram of 3.

4. Discussion

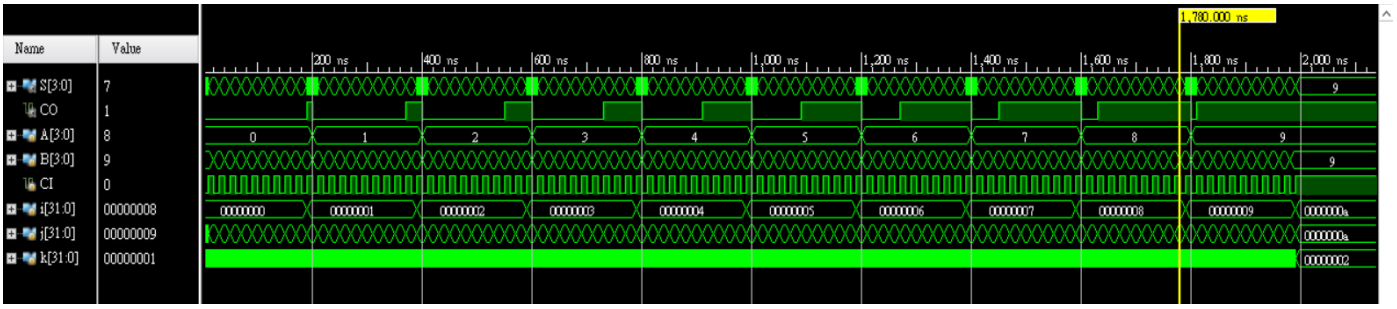
第一題是一個 full adder，基本上只要把 truth table 寫出來，用 k-map 化簡找出 logic equation，就能夠把程式寫出來，下圖（圖九）是第一題模擬波型圖：



↑圖九：1.的模擬結果

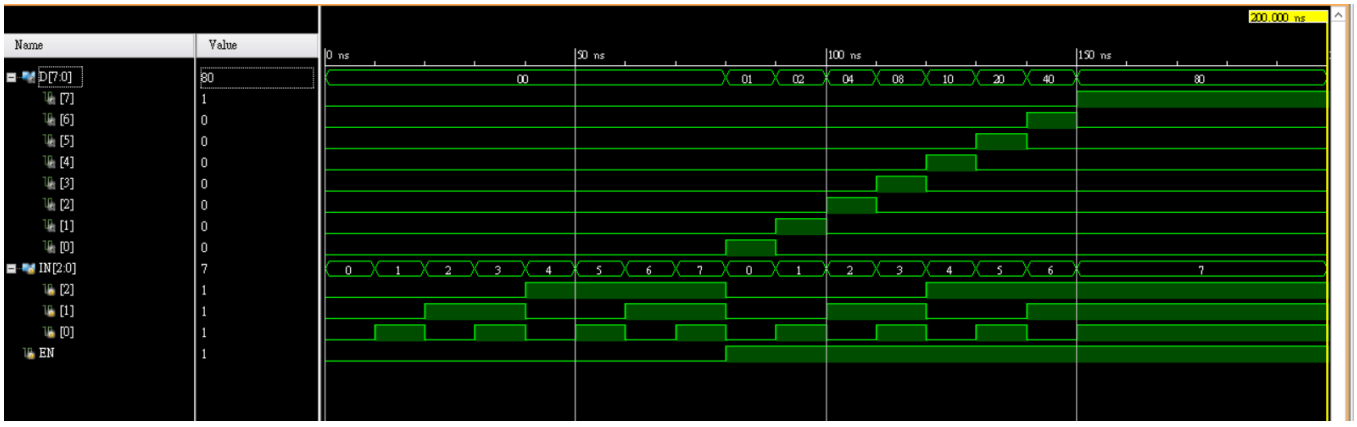
第一題讓我回憶 verilog 模擬的整個流程，也建立了使用 verilog 的一些習慣，例如：檔名不能有關鍵字（中文、空白……等）、如何 debug 我的程式，對我後面的實驗有很大的幫助。

第二題是 single digit “decimal” adder，簡單來說這一題的做法是：如果 4-bit full adder 加出來的結果大於 10，那就要把結果加 6 才是正確的答案。這題花了我比較多時間，主要是因為對 verilog 的寫法不熟悉，尤其是 if…else…敘述，我常常會把寫 C 時的想法帶入 verilog 中，但這是不對的，verilog 是硬體描述語言，C 是驅動軟體的語言，兩者有根本上的差異，後來經過一些時間習慣之後才找到 bug；再來寫 testbench 的時候也用了不少時間，因為 input 總共有上百個，所以為了找到 verilog 迴圈的使用方法花了我很大的力氣。下圖是（圖十）是第二題模擬波型圖：



↑圖十：2.的模擬結果

第三題較第二題容易，只要把 3-to-8 line decoder with enable 的 truth table 寫出來，再由 truth table 寫出 logic equation 即可，最後得到的波型圖如下（圖十一）：



↑圖十：2.的模擬結果

5. Conclusion

透過老師的上課內容和 Lab1 讓我一步一步了解 verilog 模擬的過程，大原則是：先寫出 logic equation(logic diagram)，再寫 code。其中我遇到最大的困難是對 verilog 的語法不熟悉，得要多花時間熟悉，而且 verilog 和 C 是不同的，不能用寫 C 的想法來寫 verilog，會出很多問題。

大一上邏輯設計課偶爾會有 verilog 作業，但我寫得非常糟糕，完全不知道 verilog 在做甚麼，間接讓我不想在大一下的時候修邏輯設計實驗課。後來發現 verilog 可能會在嵌入式實驗或計算機結構課用到，而且看了一下各個企業的實習工作內容，會 verilog 等於多了好幾個機會，更重要的是，幾乎每個大學的電機系硬體設計實驗都是必修，沒有修過就比別人矮了一截，基於上面幾個原因，我才決定一定要修邏輯設計實驗課。感謝助教願意解答我的疑惑。