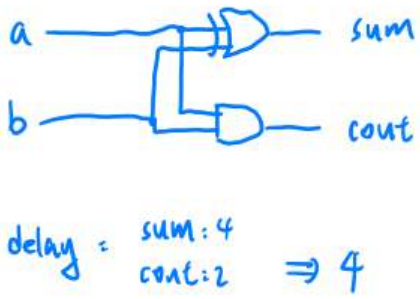
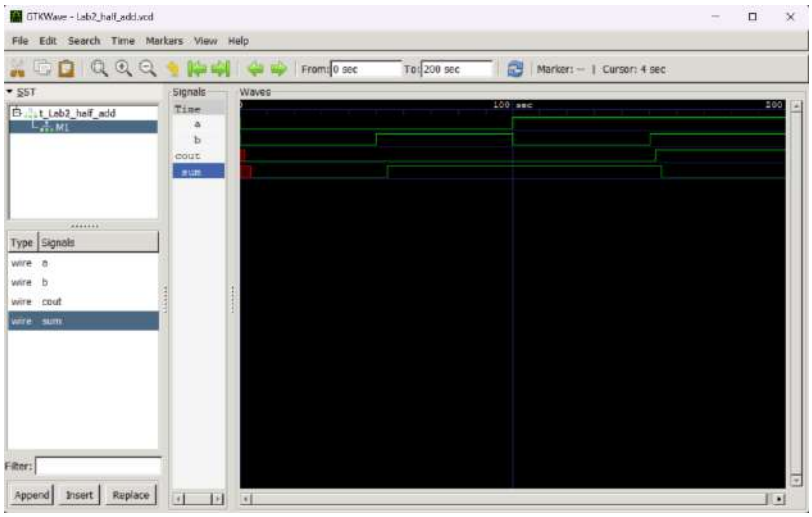
1.



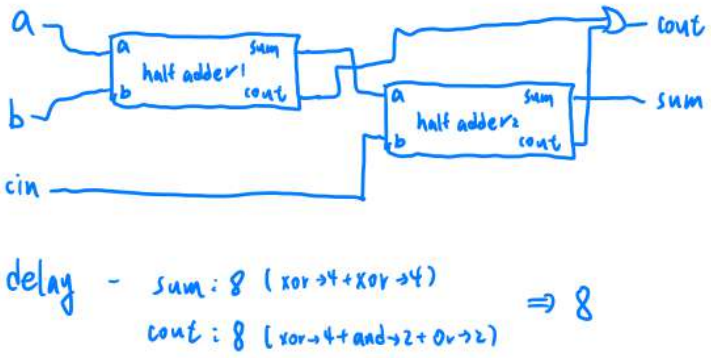
電路圖 & 延遲時間



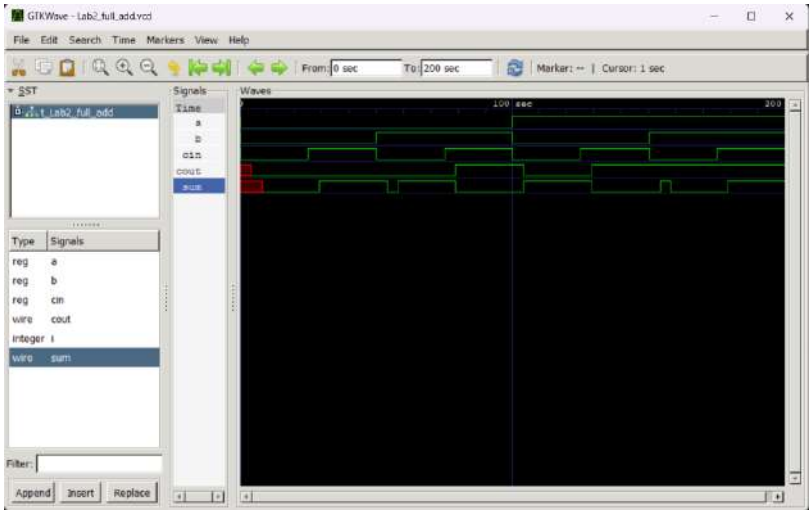
波形圖

所有 a, b 在 delay 結束後都能得到正確的 cout, sum，所以波型圖正確

2.



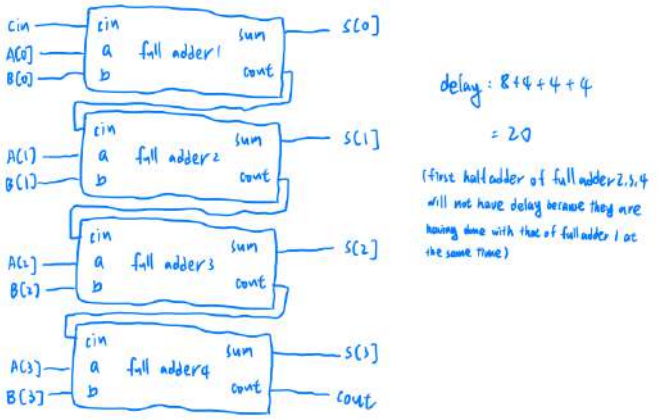
電路圖 & 延遲時間



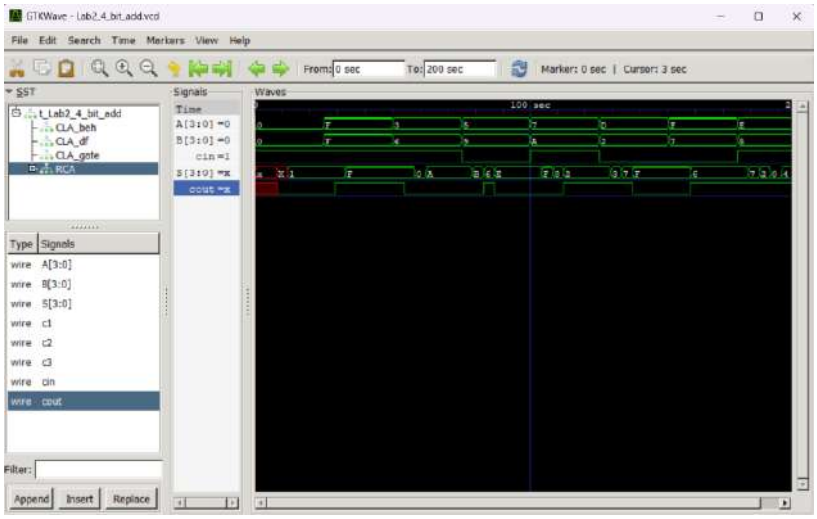
波形圖

所有 a, b, cin 在 delay 結束後都能得到正確的 cout, sum，所以波型圖正確

3.



電路圖 & 延遲時間



模擬結果

我把每個 A,B,cin 手算後，與 S, cout 相符，所以波型圖正確

4.

$C_0 = C_{in}$

$P_i = A_i \oplus B_i$, for $i = 3:0$

$G_i = A_i B_i$, for $i = 3:0$

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

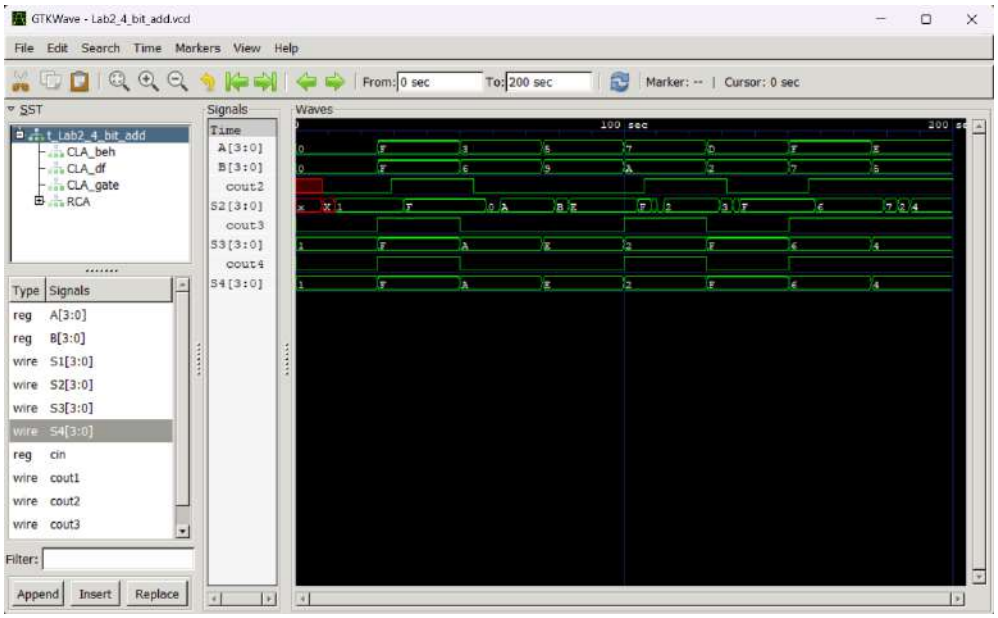
$C_4 = G_3 + P_3 G_2 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

$cout = C_4$

$S_i = P_i \oplus C_i$, for $i = 3:0$

time for gate level modeling = 4 + 4 + 4 = 12
(xor)(and+or)(xor)

布林代數式 & 延遲時間



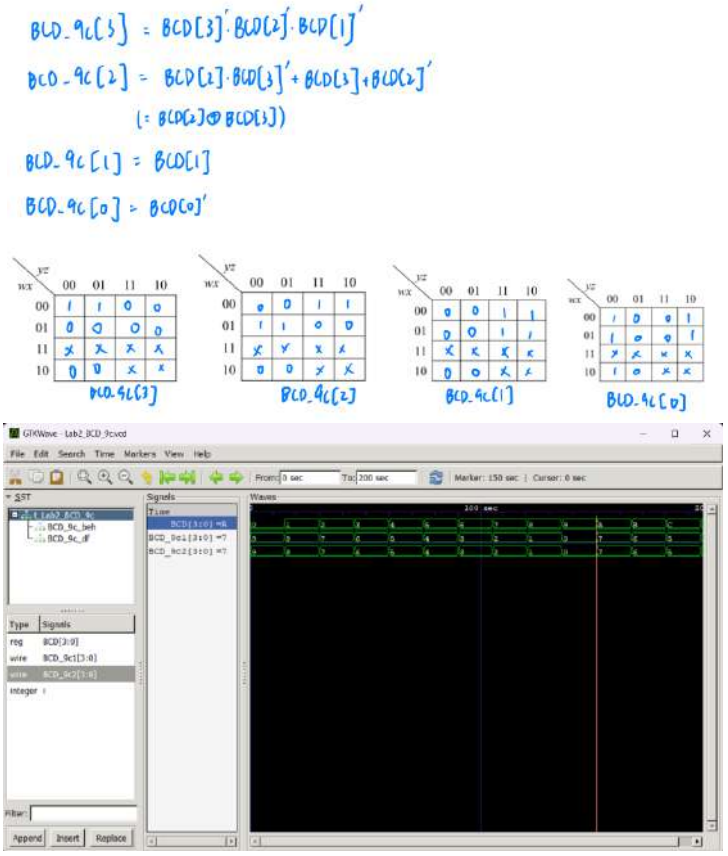
模擬結果

CLA_gate(S2, cout2), CLA_df(S3,cout3), CLA_beh(S4, cout4)

測資與結果都和 3. 一樣，所以都正確

5.

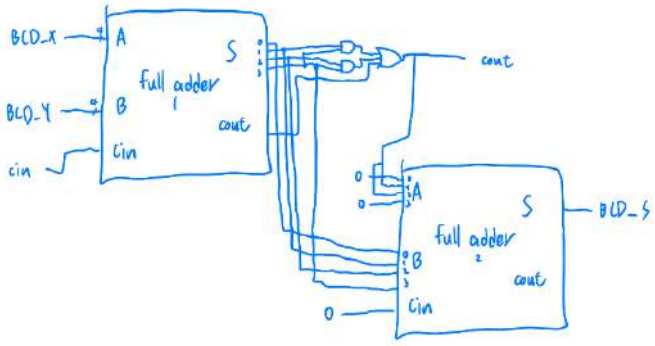
將 BCD_9c 的每個 bit 都寫成函式，並找出最精簡的 sum of products，推導過程如下



模擬結果

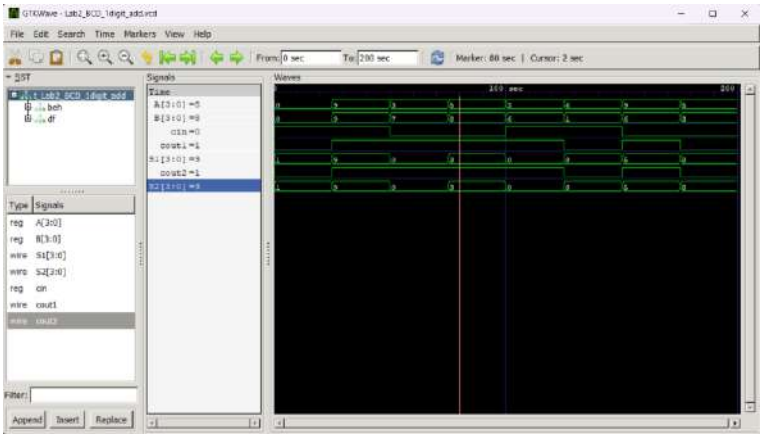
紅線右邊為 don't care condition，前 10 項 BCD + BCD_9c 都等於 9，所以正確

6.



電路圖

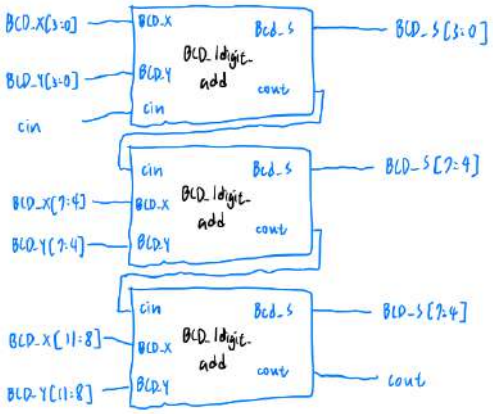
Full adder 1 把兩個 BCD 加起來，用 and 和 or 判斷是否超過 9，如果超過 9 的話 cout = 1，然後原本的 S + 0110 接到 BCD_S，跟手算 BCD 運算的過程相同。



模擬結果

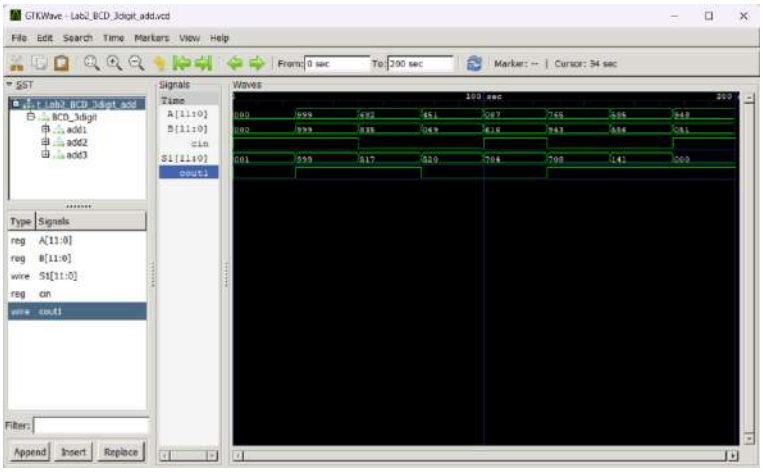
加法的結果都正確，所以正確

7.



電路圖

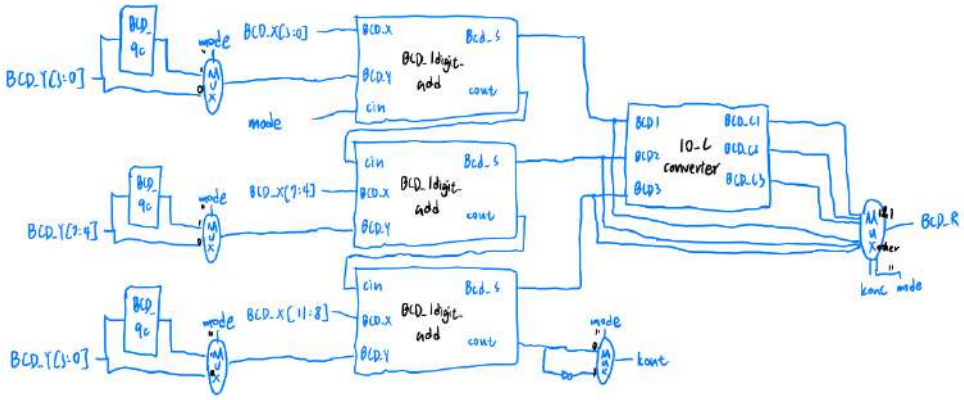
和 4 bit adder 設計原理相同，都是將前一個 adder 的 cout 丟到下一個 adder 的 cin，差別是處理的單位從 1bit 變成 4bit，加法則是用 BCD 的加法。



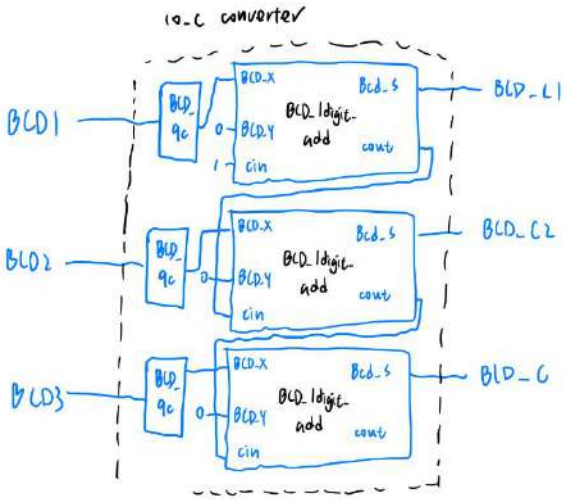
模擬結果

A + B + cin 的結果都等於 cout * 1000 + S1，所以模擬結果正確

8.



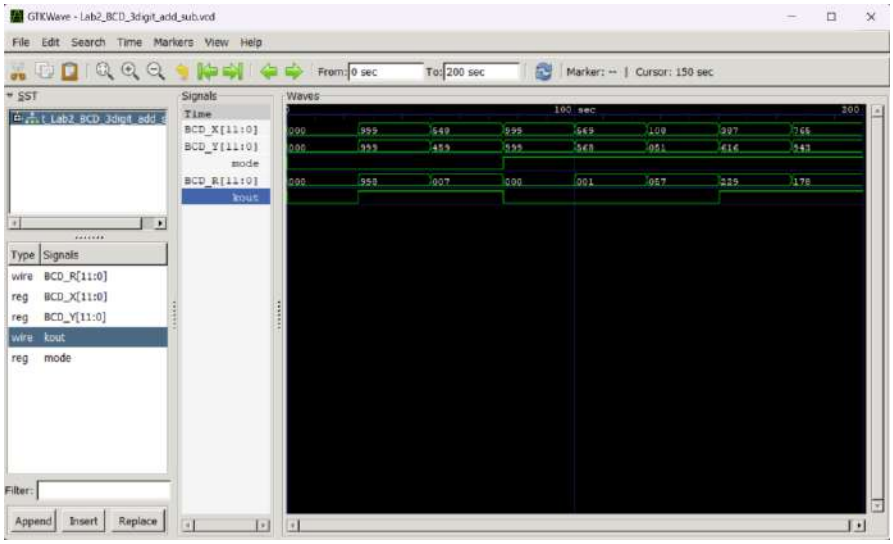
電路圖 A



10_complement 的電路圖 B

我實作的時候沒有把他們分開，這樣畫只是為了方便理解而已。

電路圖 A 的最左邊是用 MUX 來決定是否要把 BCD_Y 換成對應的 9 補數，mode = 0 (加法)不用，mode = 1 (減法)則需要。加減法的部分則是像只有加法時一樣把他串起來，如果是減法的話第一個 cin 需要 +1，而加法則是 0，這部分跟 mode 的定義一樣，所以直接用 mode 作為第一個 adder 的 cin。加減法法做完後，加法的 cout = kout，而減法如果 cout = 1 的話 BCD_X >= BCD_Y，反之則 BCD_X < BCD_Y，所以 cout = ~kout。減法如果沒有 carryout 的話則要再做一次 10 complemen。10 complement 的做法是把每四個 bit 做 9 complement，再加一。



模擬結果

驗算後，所有 BCD_R 和 kout 都正確，所以模擬正確

9.

問題與困難 ：

我覺得我花了很多時間在 debug，其中花最久的部分是 behavioral mideling 寫法(要用 reg，怎麼接寫好的 module 等等)，但寫完第一份後面就快了很多。

心得感想 ：

我花了不少時間設計 BCD 的 add 和 sub，這也是我第一次在驗算模擬結果的過程中發現錯誤，所以寫 testbench 來測試 module 實在非常重要。

我覺得 verilog 和其他我接觸到最不一樣的地方是會有很多種不同的 modeling。用 dataflow modeling 能精簡很多 gatelevel 需要 define 的 wire，寫起來也比較方便，而 behavioral modeling 比較像 functional programming，條件符合時才會執行某一段電路。