

(1) 依照 Carry Lookahead Generator 原理推導出布林代數式如下：

$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

$$C_{i+1} = G_i + P_i \cdot C_i$$

展開得到：

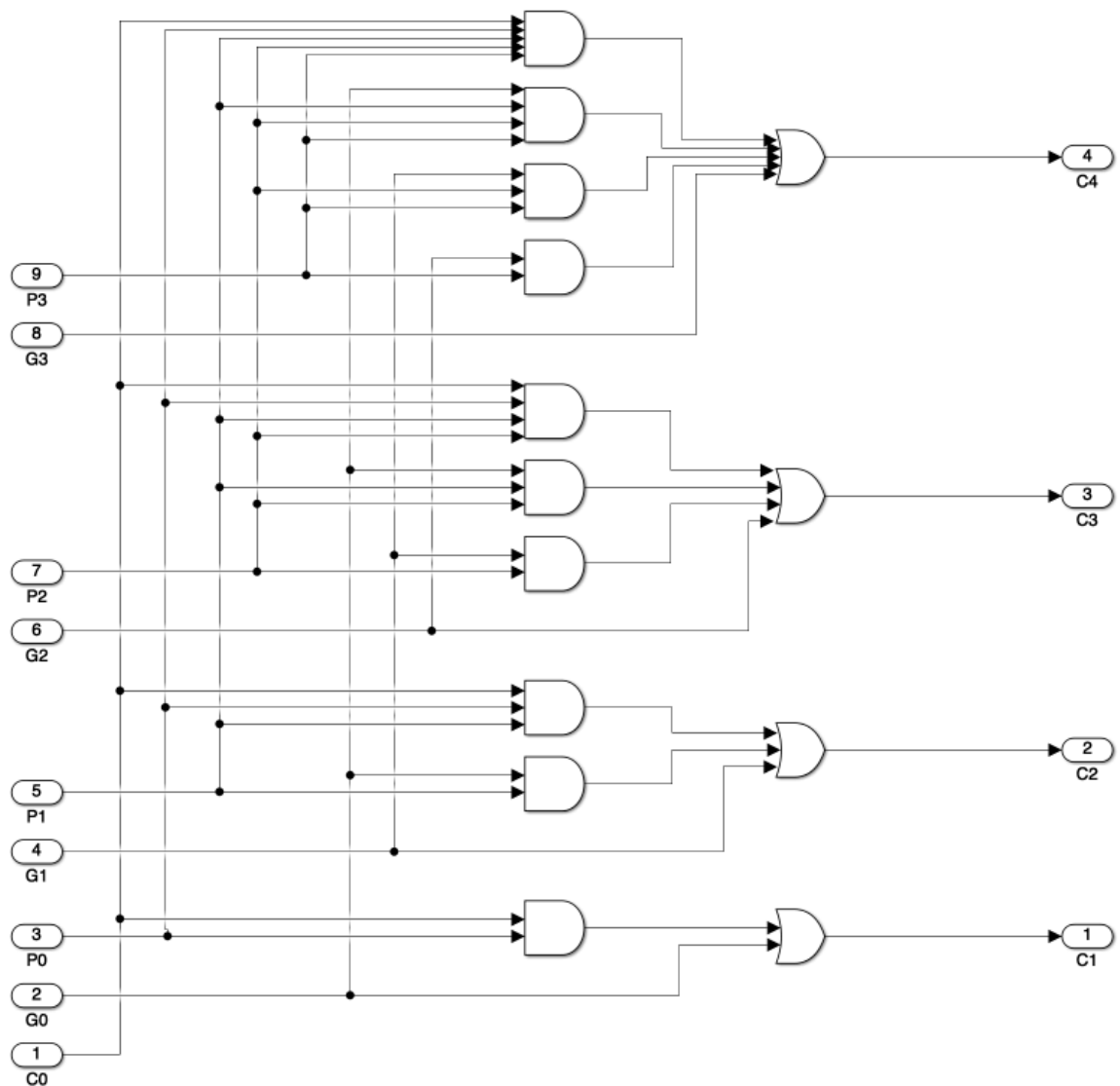
$$C_1 = G_0 + C_0 P_0$$

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

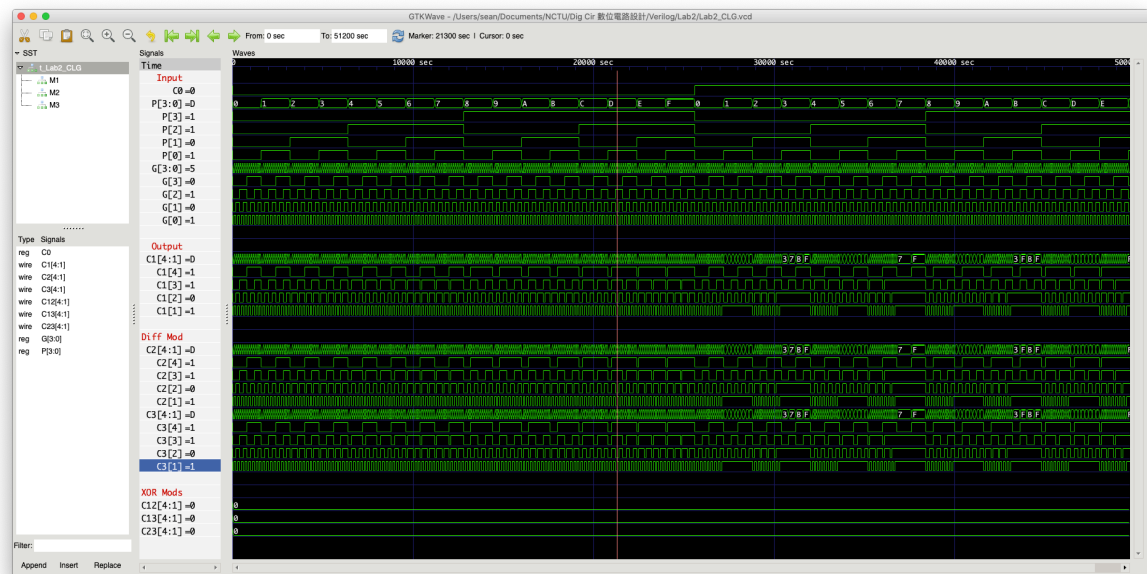
$$C_3 = G_2 + G_1 P_2 + G_0 P_2 P_1 + C_0 P_0 P_1 P_2$$

$$C_4 = G_3 + G_2 P_3 + G_1 P_3 P_2 + G_0 P_3 P_2 P_1 + C_0 P_0 P_1 P_2 P_3$$

以下為 Simulink 繪製之邏輯電路圖：



下圖為三種模擬結果之波型圖：



其中 C_1 為 gate-level modeling、

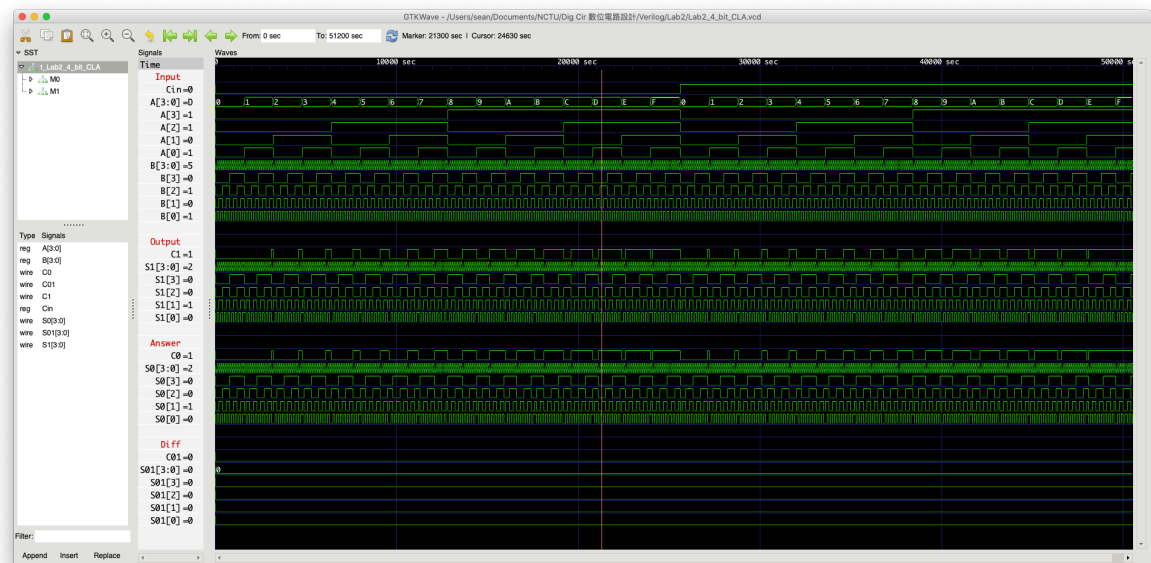
C_2 為 dataflow-level modeling、

C_3 為 behavior-level modeling

為確認三者相同，使用 $C_{i,j} = C_i \oplus C_j$ 代表兩兩之間差異。

經比對六組測資之輸出皆與題目相符。

(2) 下圖為 4-bit Carry Lookahead Adder 模擬波形圖：



其中 C_1, S_1 是以 Carry Lookahead Generator 為基礎實作出的結果、
 C_0, S_0 是以 Full Adder 做成的 Ripple Carry Adder 計算結果，
 為驗證正確性，使用 $C_{01} = C_0 \oplus C_1$ 、 $S_{01} = S_0 \oplus S_1$ 代表兩者之間差異。
 經比對八組測資之輸出皆與題目相符。

(3) 仿照 Carry Lookahead Adder 邏輯，以下為 Borrow Lookahead Subtractor 之布林代數式：

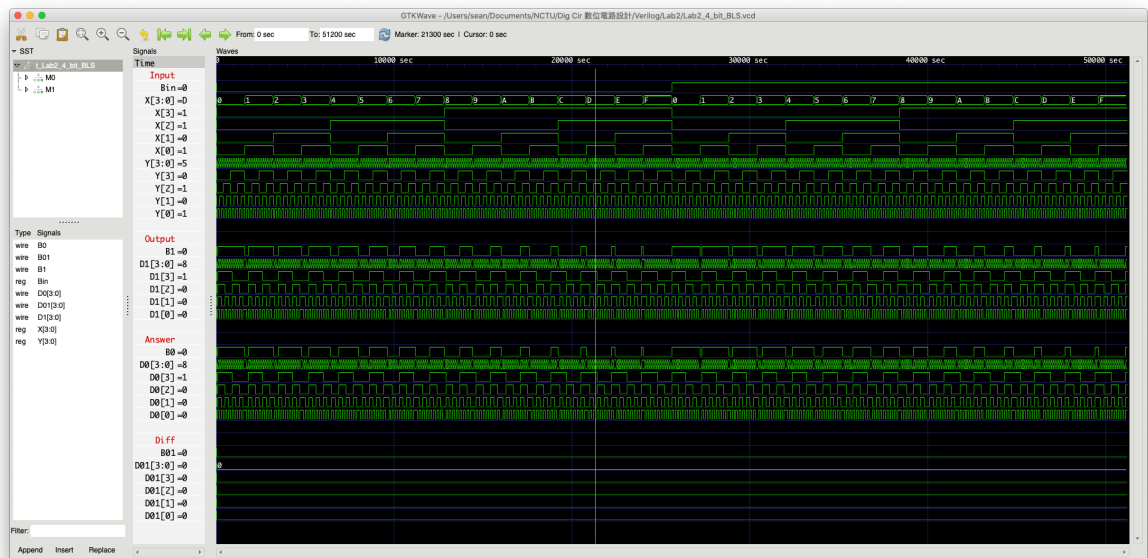
$$P_i = \overline{X_i} + Y_i$$

$$G_i = \overline{X_i} \cdot Y_i$$

$$B_i = G_i + P_i \cdot B_{i-1}$$

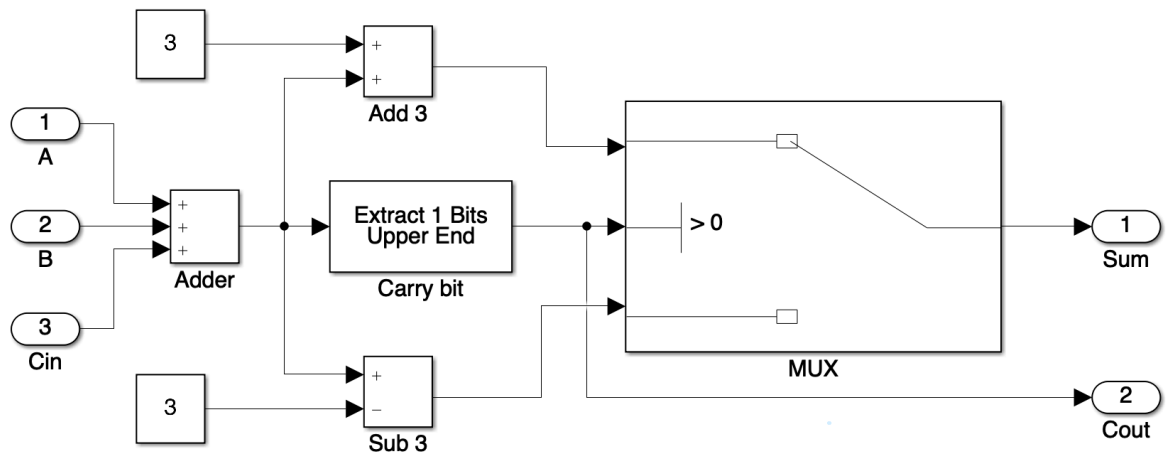
$$D_i = X_i \oplus Y_i \oplus B_i$$

電路模擬波型圖如下：



其中 B_1, D_1 是以 Carry Lookahead Generator 為基礎實作出的結果、
 B_0, D_0 是以 Full Subtractor 做成的 Ripple Borrow Subtractor 計算結果，
 為驗證正確性，使用 $B_{01} = B_0 \oplus B_1$ 、 $D_{01} = D_0 \oplus D_1$ 代表兩者之間差異。
 經比對八組測資之輸出皆與題目相符。

- (4) 使用 Excess-3 特性，以 Carry bit 判斷如何修正結果，如遇到 $C = 0$ 則 $S = A + B - 3$ ， $C = 1$ 則為 $S = A + B + 3$ 。以下為電路方塊圖

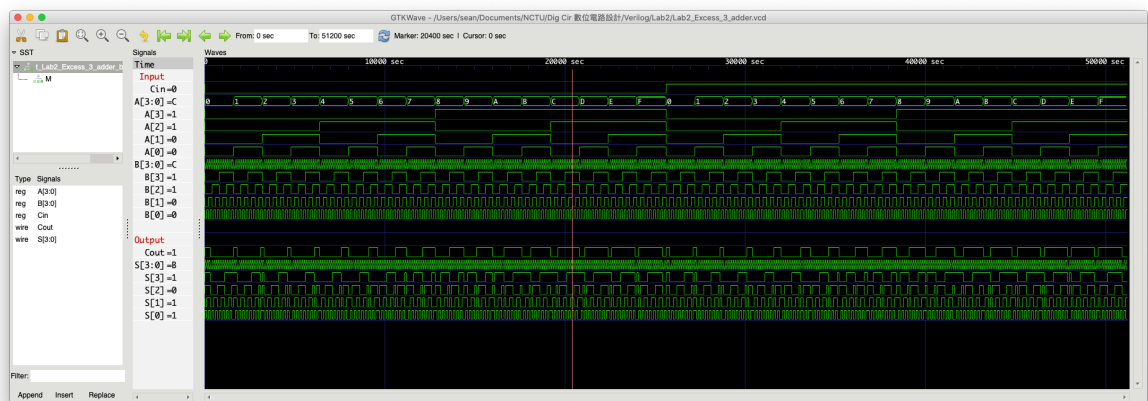


由於工具限制，無法準確表達相同電路方塊圖

圖中 Adder 為 $A + B$ 及 Carry-in = C_{in} ，使用額外方塊取 Carry bit，

後方 MUX 以中間輸入作為 Control Signal，當 $C_{out} = 0$ 時取下方值、 $C_{out} = 1$ 則取上方值

下圖為模擬波型圖：



經比對六組測資之輸出皆與題目相符。

- (5) 心得：這次學到如何使用 always block + repeat 產生全覆蓋測資，也幫助我發現過程中打錯 wire name 造成 unknown state 的問題。

此規模使用 LaTeX 繪製電路圖較不切實際，這次嘗試使用 Simulink 工具繪製，但方塊風格、功能仍比不上教材圖片。

由於不熟悉工具操作，撰寫此報告花費比想像中還要久的時間，但也逐漸建立起相關流程了，希望能愈來愈順手。