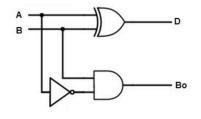
數位電路設計 Lab2 期中報告

詳述半減器之電路流程。附上2A(a)ii (半減器) 之模擬結果波形圖,並說明波形圖是否正確及所需之延遲時間。

Α	В	A⊕B (Diff)	Α'	A'·B (Bout)	
0	0	0	1	0	
0	1	1	1	1	
1	0	1	0	0	
1	1	0	0	0	

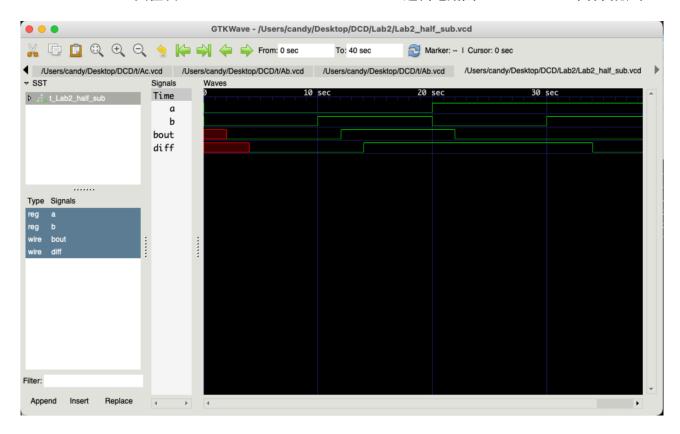


Diff = A ⊕ B Bout = A'B

I. 真值表

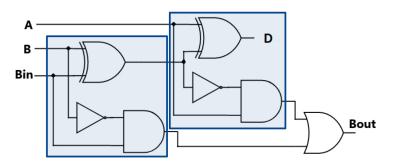
Ⅲ. 邏輯電路圖

III. 布林代數式

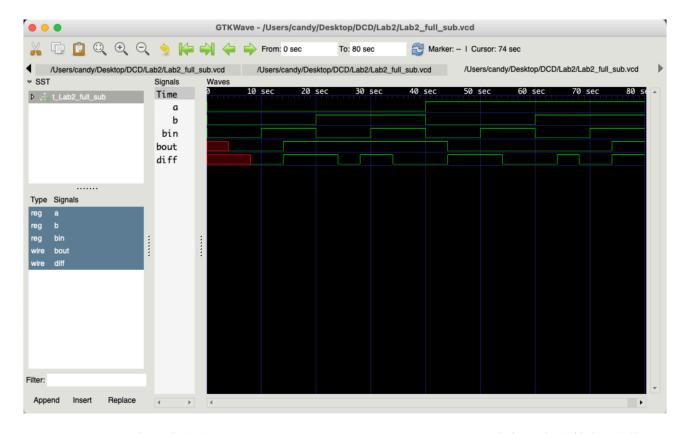


建立半減器時將 XOR gate delay 設為4秒 AND gate delay 設為2秒,因此 output 中的 bout 會延遲2秒才產生,而 diff 則是延遲4秒後產生,所以在輸出波形時才會有一開始 的紅色區塊。在扣除延遲後對應輸出波形與真值表,發現兩者結果相同,由此可以肯定輸出 結果正確。

2. 詳述如何以半減器建構全減器,畫出電路方塊圖。附上 2A(b)ii (全減器) 之模擬結果波形圖,並說明波形圖是否正確及所需之延遲時間。

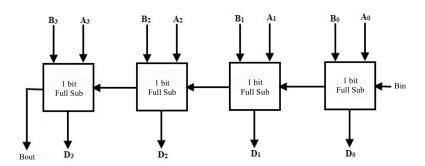


全減器可以視為將兩個數字相 減後再將所得結果減去還沒被 減到的數字,也因此可以由一 個 OR gate 再加上兩個半減器 組合而成。

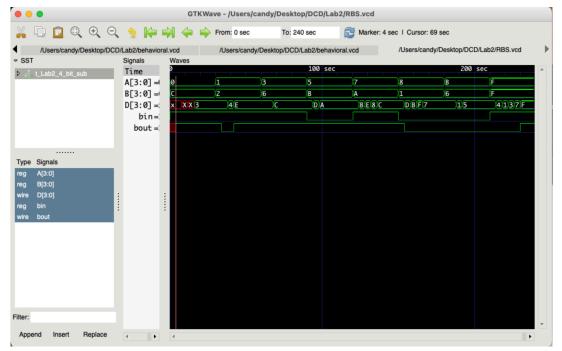


而 Bout 在輸出前經過一個 AND gate 及一個 OR gate,因此在輸出時共有4秒的延遲時間,Diff 則經過兩個 XOR gate,因此延遲時間為8秒。

3. 詳述如何以全減器建構四位元漣波借位減法器,畫出電路方塊圖。附上 2A(c)ii (4-bit RBS) 之模擬結果波形圖,並說明波形圖是否正確及所需 之延遲時間。



輸入為 4 bit 的 A 跟 B,輸出為 4 bit 的 D。將前一位元的 Bout 接到下一位元的 Bin,即可得到 4-bit RBS。



Bout 最多經過三個 XOR gate 和一個 OR gate,所以最長會有14單位的延遲,而 Diff 最多經過四個 XOR gate 和一個 OR gate,所以最長有18單位的延遲。

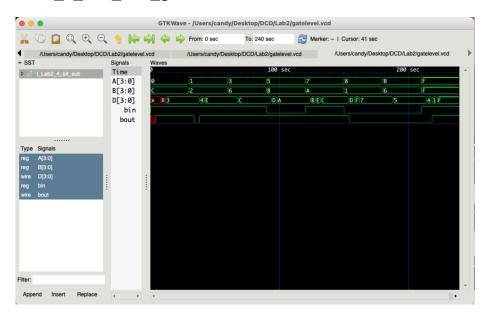
A[3:0]	B[3:0]	Bin	Bout	D[3:0]
0000	1100	1	1	0011
0001	0010	1	1	1110
0011	0110	1	1	1100
0101	1011	0	1	1010
0111	1010	1	1	1100
1000	0001	0	0	0111
1011	0110	0	0	0101
1111	1111	1	1	1111

4. 詳述四位元前看借位減法器之電路設計流程,如:列出相關布林代數式 (如:Pi、Gi、Bi、Di等)。附上 2A(d)iv (4-bit BLS) 之模擬結果波形圖, 說明三個不同電路模組之波型圖是否正確,以及 gate-level modeling 電路模組所需的延遲時間。

布林代數式

```
\begin{array}{lll} Gi &=& Ai'Bi \\ Pi &=& (Ai \oplus Bi)' \\ Di &=& (Pi \oplus Ci)' \\ C[0] &=& bin \\ 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] \\ bout &=& G[3] + P[3]G[2] + P[3]P[2]G[1] + P[3]P[2]P[1]G[0] + P[3]P[2]P[1]P[0]C[0] \end{array}
```

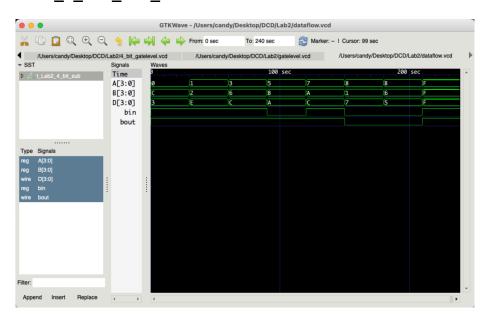
Lab2_4_bit_BLS_gatelevel



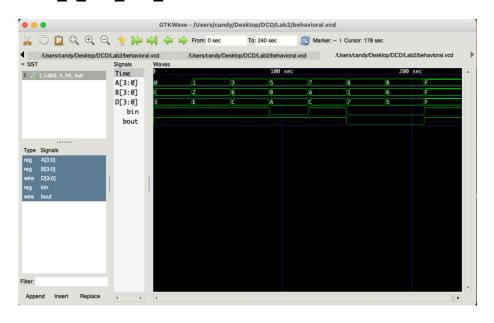
D在生成時會經過兩個xnor gate,因此會有8秒的延遲時間。

Bout在生成的時候 會經過and gate 及 or gate 各一個,因此會有 4秒的延遲時間。

Lab2_4_bit_BLS_dataflow



Lab2 4 bit BLS behavioral



Gatelevel 需要考慮延遲時間,再與計算結果相互對照,因此圖形與另外兩個不同。
Dataflow 及 behavioral 兩者電路圖並沒有增加 delay 的限制條件,因此可以直接對照計算結果且兩者圖形相同。

5. 詳述加三碼轉二進位數字轉換器之電路設計,推導出各輸出變數的最簡 sum-of-products 布林代數式,並畫出電路方塊圖。附上 2B(d) 之模擬 結果,並說明是否正確。

I. 布林代數式

```
V = E[3] \oplus E[2] + E[3]E[1]'E[0]' + E[3]'E[1]E[0]

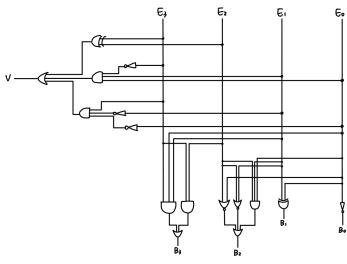
B[3] = E[3]E[1]E[0] + E[3]E[2]

B[2] = E[2]'E[1]' + E[2]'E[0]' + E[2]E[1]E[0]

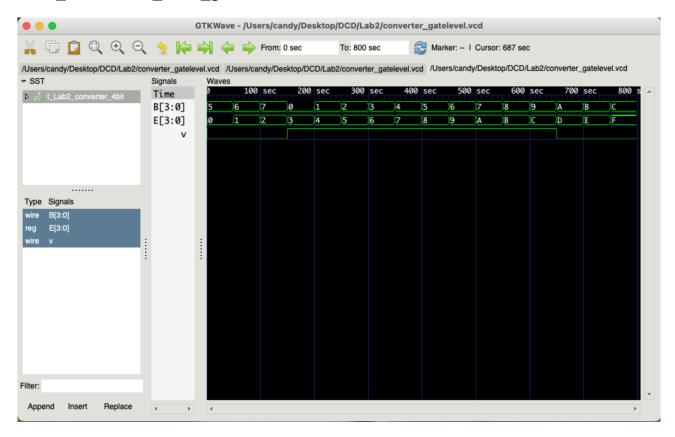
B[1] = E[1] \oplus E[0]

B[0] = E[0]'
```

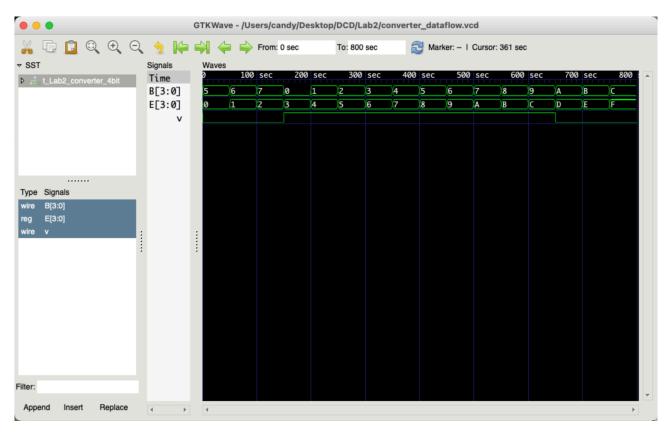
Ⅱ. 電路方塊圖



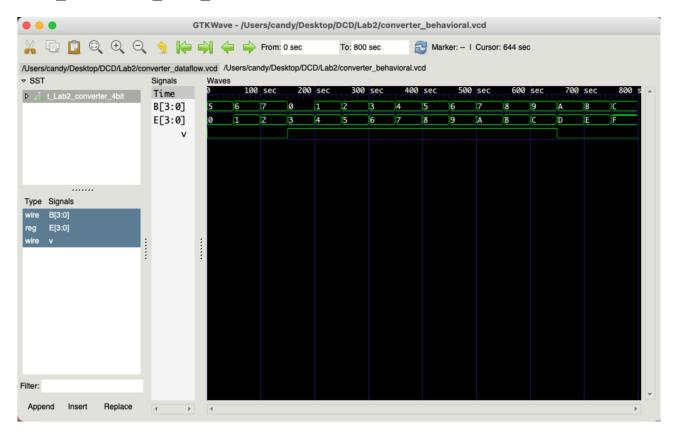
Lab2_converter_4bit_gatelevel



Lab2_converter_4bit_dataflow



Lab2_converter_4bit_behavioral



利用各輸出變數的最簡 sum-of-products 結果畫出電路圖,並對應三者圖形以及 Binary、Excess-3-code對應表,四者結果均相同,故結果正確。

6. 心得與感想、及遭遇到的問題或困難。

與平時在寫的C++不同,每次要將.v 檔圖形化便要重新生成.vvp及.vcd,因此在修改程式錯誤時特別麻煩,如此一來便會在debug及編寫電路時格外小心,避免錯誤的再次產生。然而在實作電路模組的時候使我更加地了解到半減器、全減器等基本數位電路的設計過程及原理,對與硬體描述語言也有了進一步的認知,更有助於課堂的理解,是一次很棒的經驗。