

- 詳述半減器之電路設計流程，包括：真值表、布林代數式、邏輯電路圖。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。

(a) 真值表

首先根據所有輸入組合的分別狀況推出兩 bit 相減的 borrow out、difference，如下表：

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Fig. 1 半減器 Truth Table

(b) 布林代數式

根據真值表，可以得出 borrow out、difference 的 sum of midterm： $B = \sum m(1)$ ； $D = \sum m(1,2)$ ，再嘗試以各種 gate 表示，得出最簡為 $B = x'y$ ； $D = x \oplus y$ 。

(c) 邏輯電路圖

根據其布林代數式繪製電路圖。

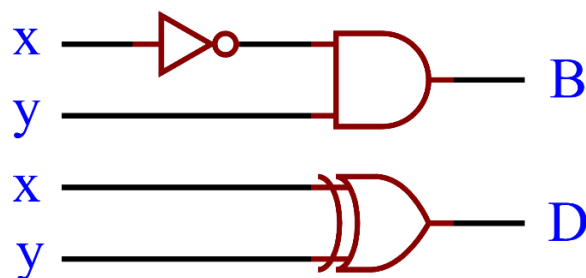


Fig. 2 半減器 邏輯電路圖

(d) 2A(a)波形圖

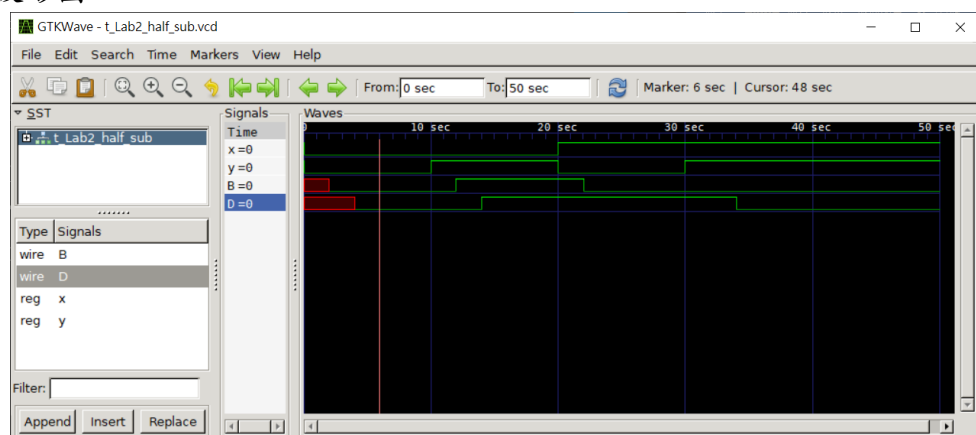


Fig. 3 半減器 波形圖

根據電路圖、gate 的 delay，可得出產出 B 的 delay 為 2s；產出 D 的 delay 為 4s，與波形圖 B, D 確定狀態時間相同；而我們將(x, y)從(0, 0)~(1, 1)的組合，依照

每隔 10s 切換，根據真值表，B 應於切換為(0, 1)時為 1，其餘為 0，然後再考慮 propagation delay(2s)，D 應於切換為(0, 1), (1, 0)時為 1，其餘為 0，然後再考慮 propagation delay(4s)，固可得知其波形正確。

- 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。

全減器即為被減數(x)減掉減數(y)之後再去減掉 borrow in(z)，則這兩部分即為兩個半減器， $x - y$ ：B1 = $x'y$, $D1 = x \oplus y$ ； $(x - y) - z$ ：B2 = $D1'z$, $D2 = D1 \oplus z$ ；則 D 即為 D2，B 即為 B1+B2。據此關係可以繪製去下圖電路，使用了兩個半減器及一個 or gate。D 所經過的路徑、gate 為 xor(D1)、xor(D2)，delay 為 8s；B 所經過的路徑、gate 為 xor(D1)、and(B2)、or(B)，delay 為 8s；and(B1)、or(B)，delay 為 4s；最長為 8s。

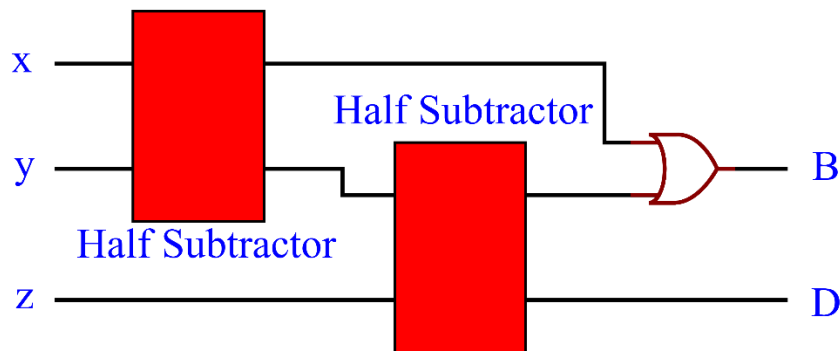


Fig. 4 全減器 電路圖

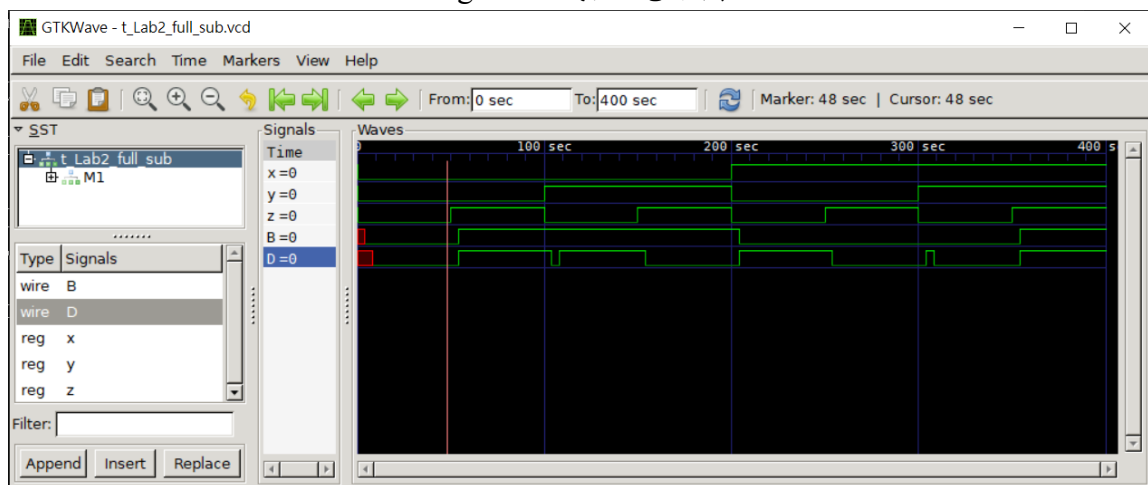


Fig. 5 全減器 波形圖

首先推得出全減器的真值表，如 Fig. 6 所示。而我們將(x, y, z)從(0, 0, 0)~(1, 1, 1)的組合，依照每隔 50s 切換，據 Fig. 6，B 應於切換為(0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 1, 1)時為 1，其餘為 0，然後再考慮 propagation delay(8s)，D 應於切換為(0, 0, 1), (0, 1, 0), (1, 0, 0), (1, 1, 1)時為 1，其餘為 0，然後再考慮 propagation delay(8s)，固可得知其波形正確。

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig. 6 全減器 Truth Table

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。

全減器為有 borrow in 的一個 bit 的減法，將其 borrow out 當作下一個 bit 減法的 borrow in 即可組成 n 位元漣波借位減法器；四位元漣波借位減法器及如下圖所示，每個 bit 各自丟進全減器作運算，而 borrow in 為上一個 bit 的 borrow out。最後一個 borrow out 即為 Bout，每個全減器的 D 即為 Diff[i]。而這個電路最長的 propagation delay 為漣波借位的路線：xor、and、or (B[1])、and、or (B[2])、and、or (B[3])、and、or (Bout)。即第一個 bit 由 xor (半減器)產生的 difference 與 borrow in 的 and、or (半減器)產生 borrow out—做為下一個 bit 的 borrow in 與下一個 bit 的 difference 的 and、or (半減器)，這樣一直串下去，得知最長 delay 為 $4n + 4(s)$ 。故四位元漣波借位減法器 delay 為 20s。

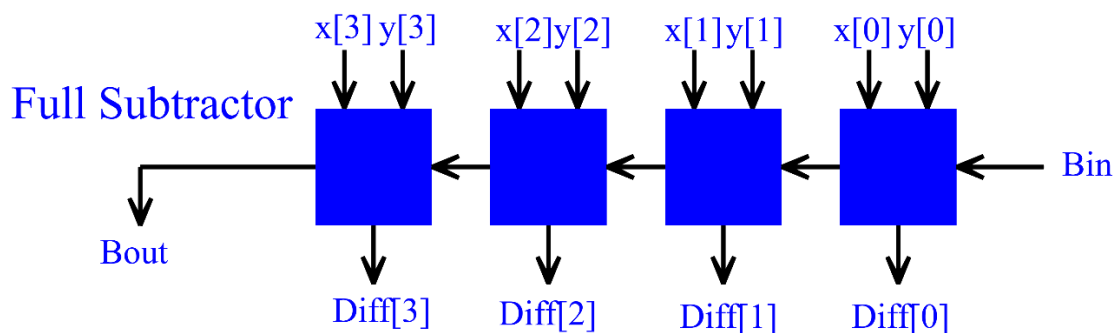


Fig. 7 四位元漣波借位減法器 電路方塊圖

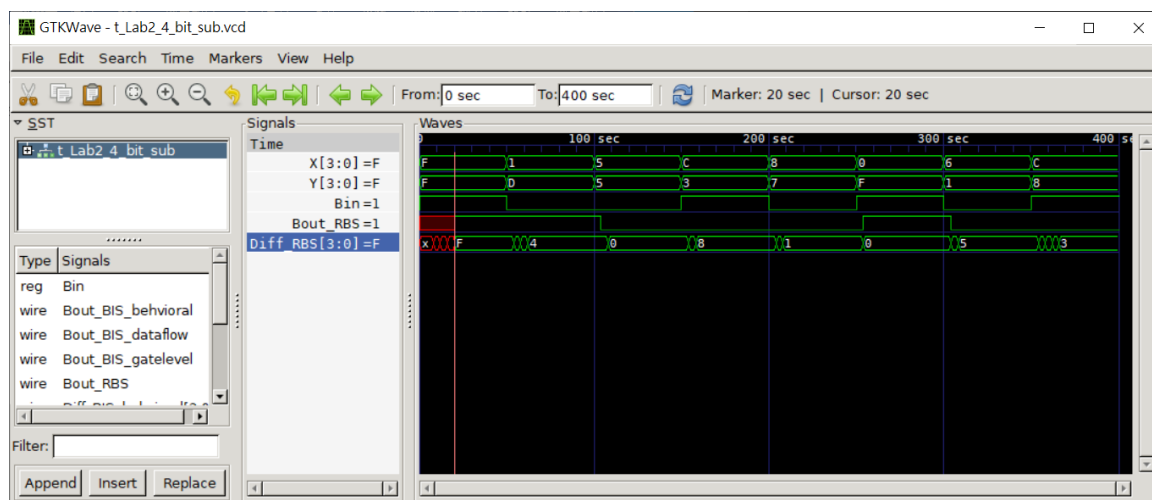


Fig. 8 四位元漣波借位減法器 波形圖

首先推得出特定 8 組測資的對照表，如 Fig. 9 所示。而我們將每組測資，依照每隔 50s 切換，據 Fig. 6，Bout 應於切換為第 1, 2, 6 組測資時為 1，其餘為 0，然後再考慮 propagation delay (20s)，Diff 的值以十六進位表示對到各組測資的答案應為 F, 4, 0, 8, 1, 0, 5, 3 然後再考慮 propagation delay (20s)，固可得知其波形正確。

X	Y	Bin	Bout	Diff
1111	1111	1	1	1111
0001	1101	0	1	0100
0101	0101	0	0	0000
1100	0011	1	0	1000
1000	0111	0	0	0001
0000	1111	1	1	0000
0110	0001	0	0	0101
1100	1000	1	0	0011

Fig. 9 特定測資 Answer

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。

以 lookahead 的方法求出 borrow out，需定義 Pi、Gi；Gi 表示此 bit 運算會產生 borrow out 的狀況 ($G_i = X_i'Y_i$)；Pi 表示此 bit 會將 borrow in 的狀態傳到 borrow out 的情況 ($P_i = (X_i \oplus Y_i)'$)；則 $D_i = (P_i \oplus B_i)'$ ； $B_i + 1 = G_i + P_i B_i$ ：

$$B_1 = G_0 + P_0 B_0,$$

$$B_2 = G_1 + P_1 G_0 + P_1 P_0 B_0,$$

$$B_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 B_0,$$

$$B_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 B_0 ;$$

其中 $B_0 = \text{Bin}$, $B_{\text{out}} = B_4$, $\text{Diff} = \{D_3, D_2, D_1, D_0\}$ 。而其最長 delay 為產生 Diff 的路徑 xnor (Pi)、and、or (Bi)、xnor (Di)，故其 propagation delay 為 12s。

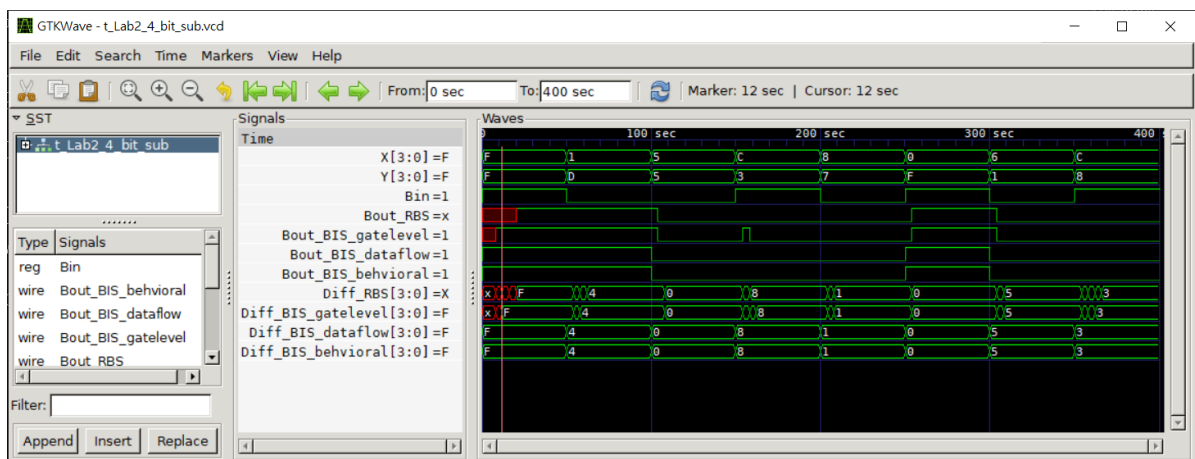


Fig. 10 四位元前看借位減法器 波形圖

其三種不同設計設計方式出來的結果，在忽略所有 delay 的狀況下應與 Fig. 8, 9 的狀況一樣。再考慮 gate level propagation delay 12s、dataflow、behavioral 沒有設置 delay，可得知 Fig. 10 的波形圖正確。

5. 詳述五輸入優先編碼器之電路設計，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.iv(五輸入優先編碼器) 之模擬結果，並說明是否正確。

根據優先編碼器之真值表，可以得 V 為全部 bit 是 0 時為 0 其餘為 1；A[2]則為 D[0]~D[3]為 0、D[4]為 1 時為 1，其餘為 0；A[1] 則為 D[0]~D[1]為 0、D[2]為 1 時或 D[0]~D[2]為 0、D[3]為 1 時為 1，其餘為 0；A[0] 則為 D[0]為 0、D[1]為 1 時或 D[0]~D[2]為 0、D[3]為 1 時為 1，其餘為 0；可得以下布林代數式：

$$V = D[0] + D[1] + D[2] + D[3] + D[4] \text{ 。}$$

$$A[2] = D[0]'D[1]'D[2]'D[3]'D[4] \text{ 。}$$

$$A[1] = D[0]'D[1]'D[2] + D[0]'D[1]'D[2]'D[3] = D[0]'D[1]'D[2] + D[0]'D[1]'D[3] \text{ 。}$$

$$A[0] = D[0]'D[1] + D[0]'D[1]'D[2]'D[3] = D[0]'D[1] + D[0]'D[2]'D[3] \text{ 。}$$

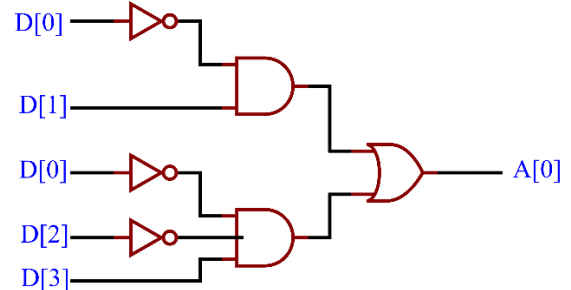
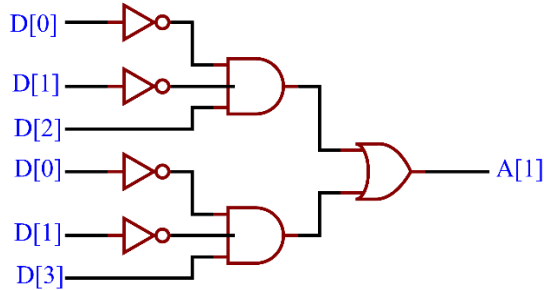
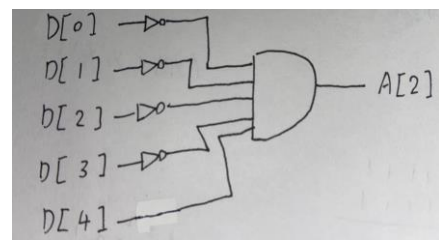
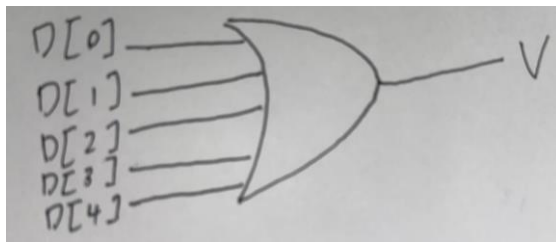


Fig. 11 五輸入優先編碼器 電路方塊圖

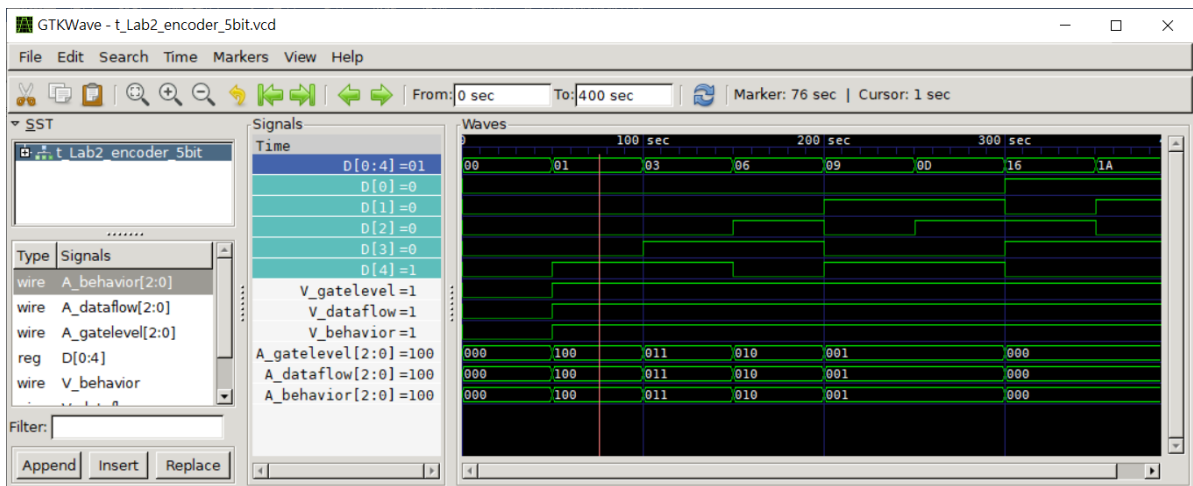


Fig. 12 五輸入優先編碼器 波形圖

首先推得出特定 8 組測資的對照表，如 Fig. 13 所示。而我們將每組測資，依照每隔 50s 切換，由波形圖可以看到三種寫法其波形一致，且 V 只於前 50s 為 0 後皆為 1，A 的值只須看 V 為 1 時的狀況，而其值對應與 Fig. 13 相同，固可得知其波形正確。

D	A	V
00000	xxx	0
00001	100	1
00011	011	1
00110	010	1
01001	001	1
01101	001	1
10110	000	1
11010	000	1

Fig. 13 特定輸入 真值表

6. 心得與感想、及遭遇到的問題或困難。

寫到 4-bit Ripple Borrow Subtractor 時，感覺還不錯，滿有心得的，但寫到 4 bit Borrow Lookahead Subtractor 就開始有點崩潰，太繁瑣了，原本想說 test moudle 總算有比較好的方法去測試所有輸入組合，結果又來一個更繁瑣的，重複性高的電路要設計，又因要維持 lookahead 不能簡化它，覺得滿麻煩的，其餘就沒什麼特別的問題，大概就剩要注意 propagation delay 不然波形會是錯的。