

投稿類別：工程技術類

篇名：

漣波進位加法器和前瞻進位加法器的實作與研究

作者：

吳承濤。臺北市立松山工農。電子科二年級智班
陳尚緯。臺北市立松山工農。電子科二年級智班
鍾秉融。臺北市立松山工農。電子科二年級智班

指導老師：
林麗雲老師

壹、前言

一、研究動機

進來電子科已經一年了，使用過了許多的 IC，在以前老師解說電路構造時，總是很好奇為何一顆小小的 IC 就可以處理許多電路上所需的功能。而現在的我們專業知識較以前多，已經可以開始研究。在上課時，老師有特別講過，經過邏輯閘會造成延遲時間，而延遲時間會對電路造成一些影響，所以我們決定探討不同加法器的運算速度。

二、研究目的

現在的我們正在學到的各式各樣的加法器，透過組合與應用，我們可以創造出各種不同功能的電路。查詢資料後，我們發現不同組合的加法器，雖可以實現相同功能，但延遲時間卻不同。因此我們想要做一個透過改進一加法器來減少其所造成的延遲時間並探討差異地方並整理與實驗。

三、研究方法

我們利用 Altera Quartus II 這個軟體，這個軟體支援圖形編輯設計與硬體描述語言，而這次我們使用圖形編輯設計，畫出漣波進位加法器和前瞻進位加法器的電路圖，藉由軟體內建的波形模擬的功能，完成並驗證波形，若正確，再將電路圖燒入 CPLD 邏輯元件 MAX II 系列的 EPM1270T144C5 晶片之中，並實際使用指撥開關與 LED 燈的亮暗來驗證數值是否正確。透過計算理論數值去驗證是否與指撥開關和 LED 燈的亮暗符合並記錄，最後統整實驗記錄並比較差異。

貳、正文

一、相關知識

在一般認知中，我們普遍都會認為 1 為高電位（開）、0 為低電位（關），這是所謂的高態動作（Active High）如圖 1。但這次我們所使用的是低態動作 LED，故 1 為低電位、0 為高電位，也就是說當我們將開關撥為 1 時，LED 將不會亮，而當我們把開關撥為 0 時，LED 才會亮。這是因為電路的接法不同，低態動作 LED 為 LED 陰極接到測試機台的接腳，而高態動作 LED 為 LED 陰極接地，兩種各有其用途。

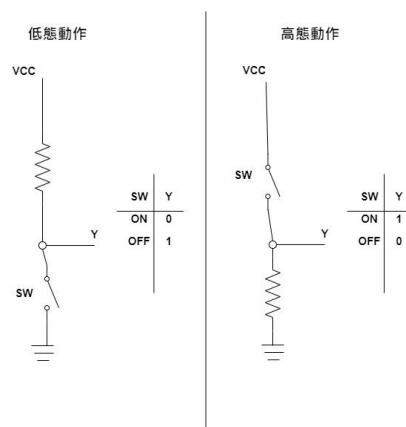


圖 1 高低態動作圖

(一) 半加器

半加器是將 2 位元的 2 進制進行加法運算而不考慮前級是否有進位輸入，得出的結果整理成真值表（如表 1），故我們可以知道得出的兩輸出各為商（互斥或閘）與進位（及閘）。所以將互斥或閘和及閘進行組合得出以下結果，如圖 2。被加數 A 和加數 B 經過一個互斥或閘得到 S（和）， $S=A \oplus B$ ，A 和 B 再經過一個及閘得到 C（進位）， $C=AB$ 。

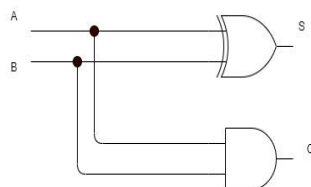


圖 2 半加器

表 1 半加器真值表

被加數	加數	進位	和
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(二) 全加器

與半加器不同的是，全加器多了"進位"的功能，使他可以進行 2 位元以上的加法運算，它可以用一個三輸入的互斥或閘、3 個兩輸入及閘、一個兩輸入或閘組合，亦可以用兩個半加器、一個兩輸入或閘組合而成，如圖 3。

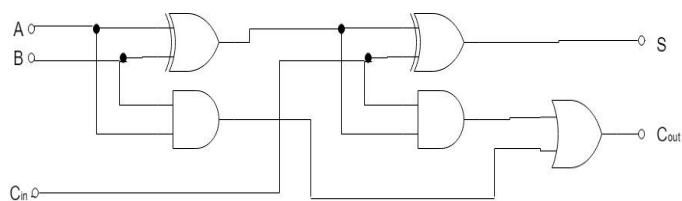


圖 3 全加器

表 2 全加器真值表

被加數	加數	進位	和	進位輸出
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

被加數 A 和加數 B 經過一個互斥或閘，得到的結果再和 C_{in} 經過一個互斥或閘，得出的結果為 S (和)， $S = A \oplus B \oplus C$ 。被加數 A 和加數 B 經過一個或閘得出的結果再和被加數 A 和加數 B 經過一個互斥或閘，得到的結果和 C_{in} 經過一個或閘得到的結果經過一個或閘，得出的結果為 C_{out} (進位)， $C_{out} = AB + AC_{in} + BC_{in}$ 。

(三) Quartus II 介紹

Altera 的 Quartus II 9.1 設計軟體為可程式邏輯電路設計開發工具之一，透過繪製電路圖或寫 VHDL 程式，使我們完成 CPLD 的製作，並可以模擬和檢測其波形圖與除錯功能，方便我們檢查電路功能。

二、本小論文實驗接線圖

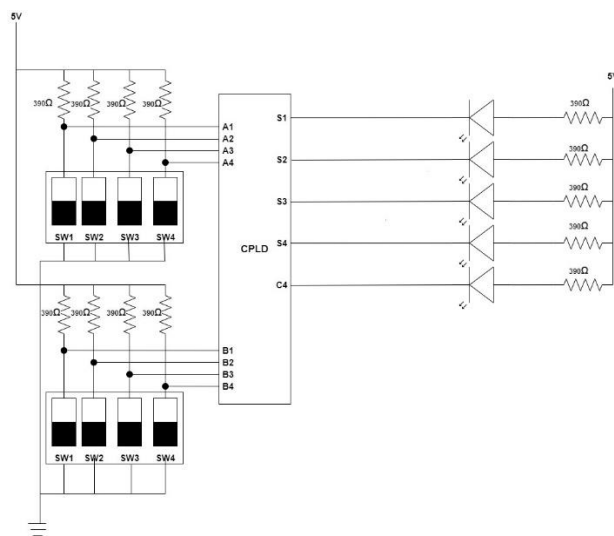


圖 4 本小論文實驗接線圖

如圖 4 透過控制指撥開關，將加數與被加數燒入進 CPLD 邏輯元件 MAX II 系列的 EPM1270T144C5 晶片之中，經過 CPLD 內部的進位加法運算，將結果使用 LED 的亮暗來呈現出結果。

三、研究流程圖

運用 1 位元的全加法器，深入探討 4 位元加法器。研究漣波加法器與前瞻進位加法的原理與構造，了解兩種加法器後，藉著老師的指導，推導布林代數式，畫成邏輯閘，與得出真值表，藉由在麵包板上實作，實際觀察來驗證手算的概念是否正確使用指撥開關控制輸入端（加數&被加數）、LED 控制輸出端（進位&和數），將實驗結果紀錄，用兩者的實驗記錄得出相似點、差異，整理成圖表，並延伸討論出兩者在製作及使用上的優缺點。我們研究過程為如圖 5。

漣波加法器和前瞻進位加法器的實作與研究

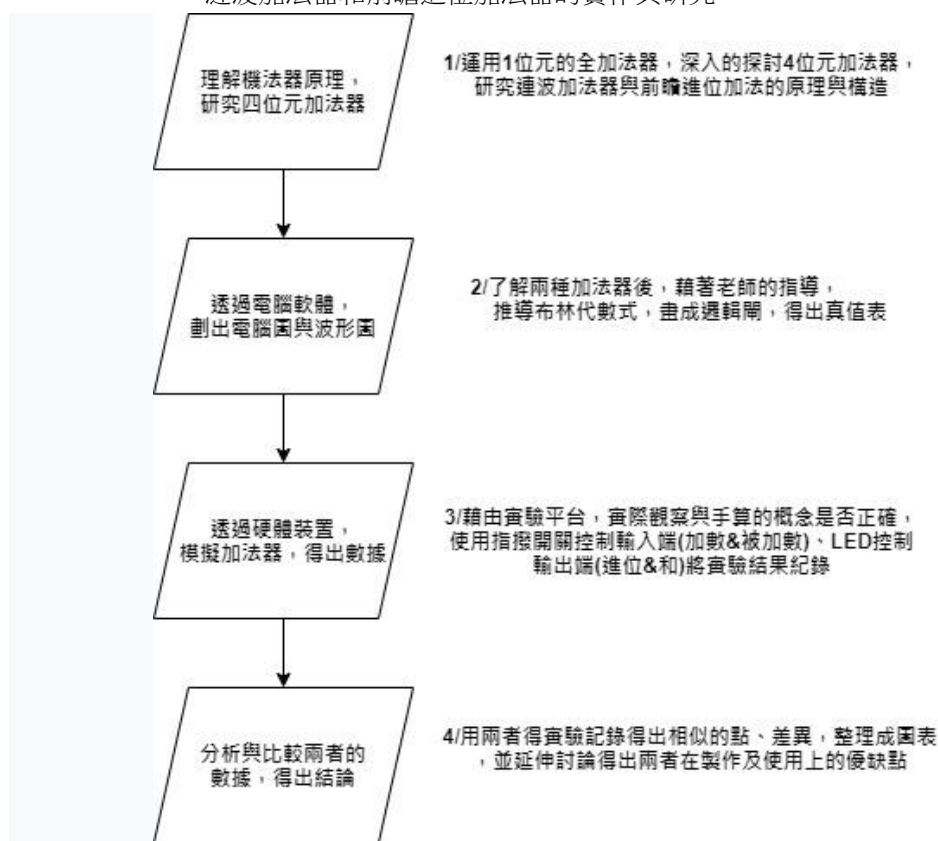


圖 5 研究過程圖

四、研究結果

(一) 漣波加法器

漣波加法是按部就班的 1 次進行 1 個位元的加法（進行 4 次加法）才能得出結果，也就是必須由 2 個數值的最低位元先相加得出的商和進位才能進行第二次的運算，進行 4 次。

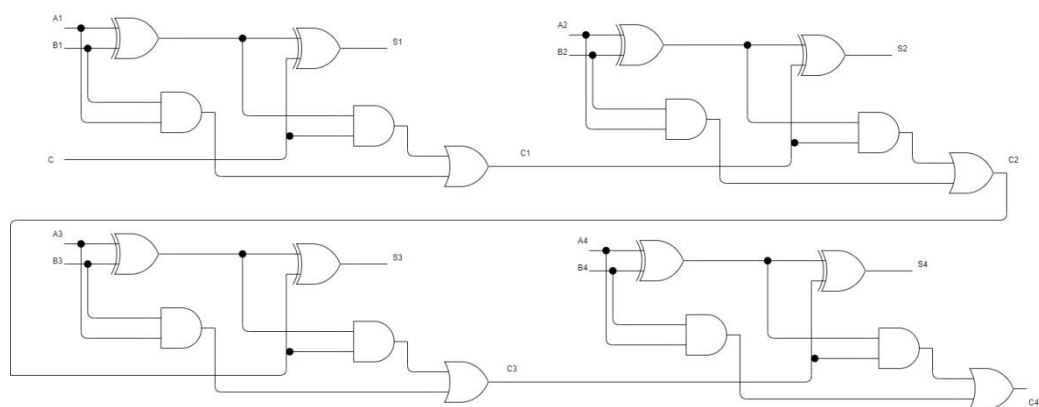


圖 6 漣波進位加法器電路圖

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$

$$S = A_i \oplus B_i \oplus C_{i-1}$$

依照上面的布林代數式，我們可以設輸出和 S_i 為 $A_i \oplus B_i \oplus C_{i-1}$ 、輸出進位 C_i 為 $A_i B_i + A_i C_{i-1} + B_i C_{i-1}$ 。依照圖 6 我們可以知道，第一位元輸出和 S_1 為 $A_1 \oplus B_1 \oplus$

漣波加法和前瞻進位加法的實作與研究

C_{gnd} 、第一位元輸出進位 C_1 為 $A_1B_1 + A_1C_{gnd} + B_1C_{gnd}$ 。第二位元輸出和 S_2 為 $A_2 \oplus B_2 \oplus C_1$ ，第二位元輸出進位 C_2 為 $A_2B_2 + A_1C_1 + B_2C_1$ 。第三位元輸出和 S_3 為 $A_3 \oplus B_3 \oplus C_2$ ，第三位元輸出進位 C_3 為 $A_3B_3 + A_3C_2 + B_3C_2$ 。第四位元輸出和 S_4 為 $A_4 \oplus B_4 \oplus C_3$ ，第四位元輸出進位 C_4 為 $A_4B_4 + A_4C_3 + B_4C_3$ 。由上述式子可以得知，第二位元的進位輸出需要等待第一位元的進位輸出後才能得出，第三或第四進位輸出也是以此類推。因此我們可以得知，如果邏輯閘的數量過多，要得到進位輸出所需的時間將會依照邏輯閘數量提升，非常不具效率。我們使用 Altera Quartus II 的圖形編輯編輯出圖 7，並使用波形編輯器模擬時序圖得如圖 8。若檢查功能無誤，再將其燒入 CPLD，進行測試，比對我們所計算出的結果，利用表 3，驗證答案。

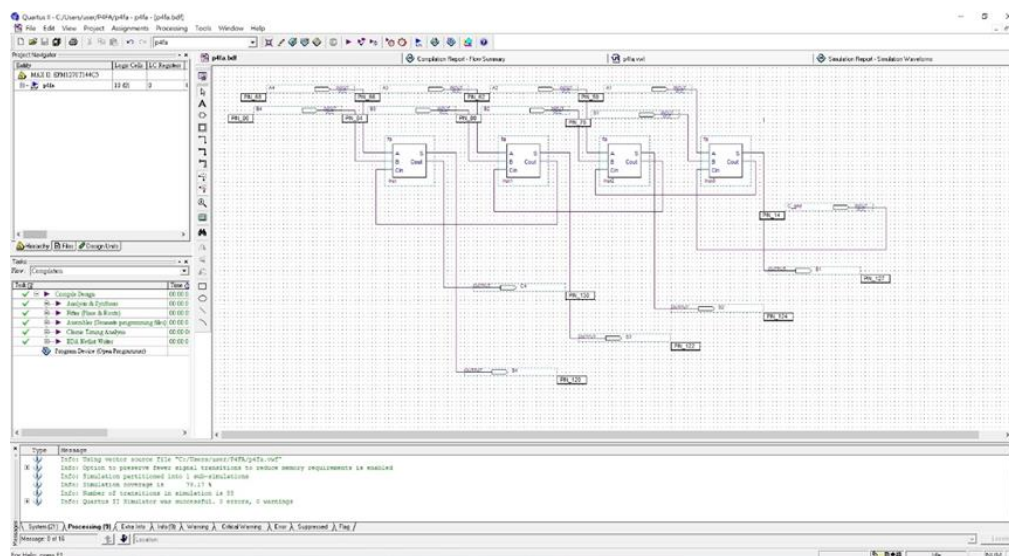


圖 7 漣波進位加法器電路圖

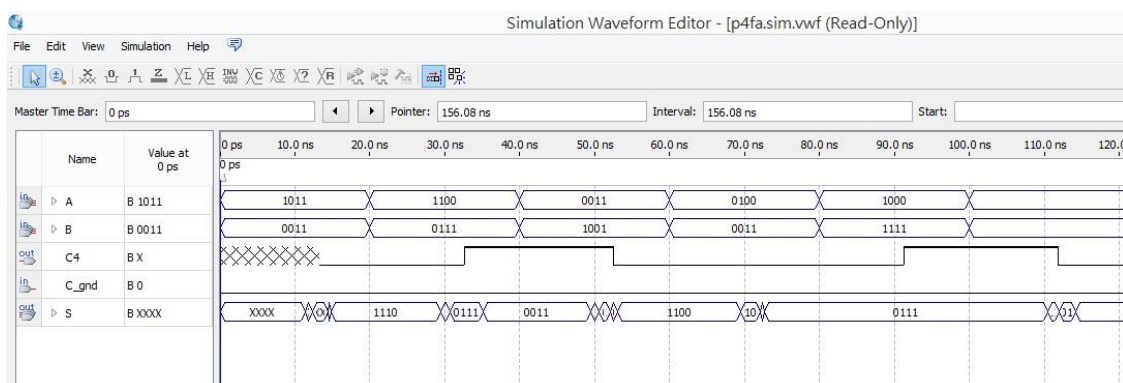


圖 8 漣波進位加法器時序圖

表 3 漣波加法器 真值表

C_{gnd}	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	S_4	S_3	S_2	S_1	C_4
0	1	1	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	0	0	1	0	0	1	1	0
0	0	1	0	0	0	0	1	1	1	1	0	0	0
0	1	0	0	0	1	1	1	1	0	1	1	1	0
0	1	0	1	1	0	0	1	1	0	1	1	0	1

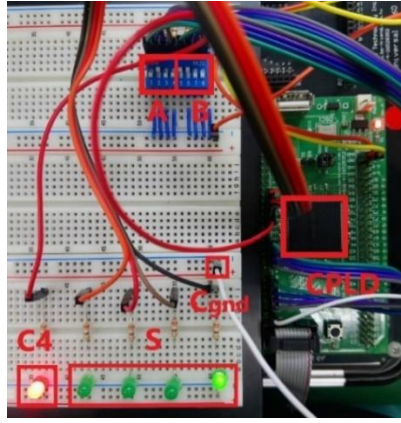


圖 9 漣波進位加法器實作結果

由表 3 的第五行數值， $A_4 = 1$ ， $A_3 = 0$ ， $A_2 = 1$ ， $A_1 = 1$ 、 $B_4 = 0$ ， $B_3 = 0$ ， $B_2 = 1$ ， $B_1 = 1$ ，得出的結果為 $C_4 = 0$ ， $S_4 = 1$ ， $S_3 = 1$ ， $S_2 = 1$ ， $S_1 = 0$ ，如上圖 9。

(二) 前瞻進位加法器

前瞻進位加法器是 1 次進行 4 個位元的加法，不需要等到最低的兩位元加法得出的商，在進行下一個位元的運算，因此可以省下許多時間。

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_i = G_i + P_i C_{i-1}$$

$$C_1 = G_1 + P_1 C_{gnd}$$

$$C_2 = G_2 + P_2 C_1 = G_2 + P_2 (G_1 + P_1) = G_2 + P_2 G_1 + P_2 P_1 C_{gnd}$$

$$C_3 = G_3 + P_3 C_2 = G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 C_{gnd})$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_{gnd}$$

$$C_4 = G_4 + P_4 C_3$$

$$C_4 = G_4 + P_4 (G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_{gnd})$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_{gnd}$$

以四位元加法器為例，依照圖 10 我們可以先設定 i 為第幾位元， $A_1 \sim A_4$ 為四位元被加數輸入代數， $B_1 \sim B_4$ 為四位元加數輸入代數， A_i 互斥 B_i 得到的結果為 P_i ， A_i 及 B_i 得到的結果為 G_i 。依照圖 10，第一位元相加和的輸出為 S_1 ，布林代數式為 $S_1 = P_1$ ，第一位元進位輸出為 C_1 ，布林代數式為 $C_1 = G_1 + P_1 C_{gnd}$ ，由第一位元的相加和與進位的布林代數式，我們可得進位的布林代數式為 $C_i = G_i + P_i C_{i-1}$ ，相加和的布林代數式為 $S_i = P_i \oplus C_{i-1}$ ，通過布林代數式我們可以求出第二位元的進位輸出布林代數式為 $C_2 = G_2 + P_2 C_1$ ，第二位元相加和的輸出布林代數式為 $S_2 = P_2 \oplus C_1$ ，第三位元的進位輸出布林代數式為 $C_3 = G_3 + P_3 C_2$ ，第三位元相加和的輸出布林代數式為 $S_3 = P_3 \oplus C_2$ ，第四位元的進位輸出布林代數式為 $C_4 = G_4 + P_4 C_3$ ，第四位元相加和的輸出布林代數式為 $S_4 = P_4 \oplus C_3$ ，轉換成邏輯閘，可得到如圖 10，確認無誤後，再透過 Altera 的 Quartus II 的圖形編輯設計得到圖 11 並取其時序圖 12，將其整理成如表 5 來做分析與比較。前瞻加法器是透過將前一進位分解，直接連上前幾進位的 P 與 G 藉此可不用透過前一位進位運算完後，在進行加法，等於在做完各

漣波加法和前瞻進位加法器的實作與研究
位元的互斥與及聞後，將數字發送給各個進位，並同步進行加法與進位。

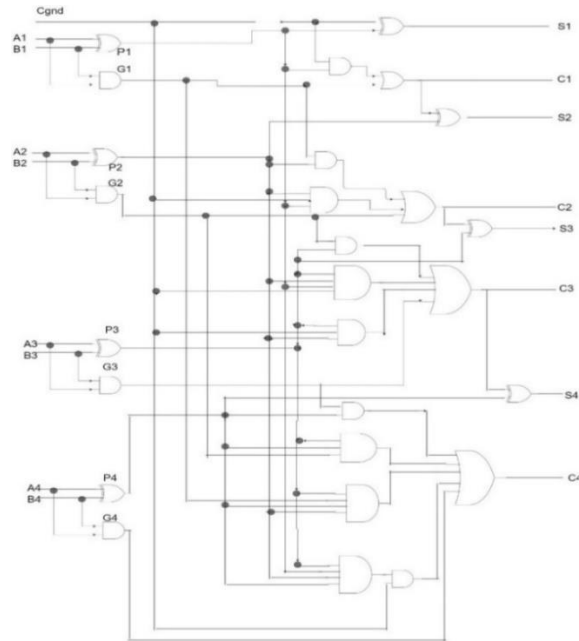


圖 10 前瞻進位加法器電路圖

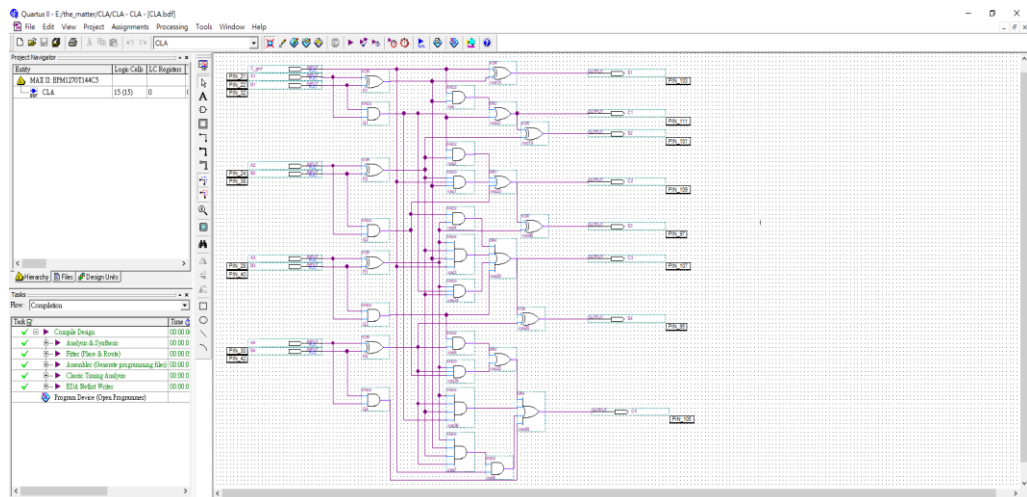


圖 11 前瞻進位加法器電路圖

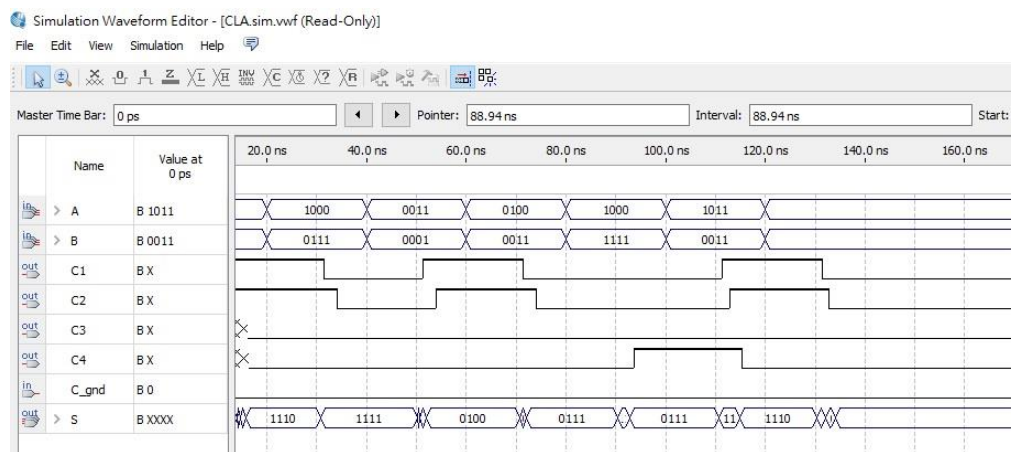


圖 12 前瞻進位加法器時序圖

表 4 前瞻進位加法器 真值表

C_{gnd}	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	S_4	S_3	S_2	S_1	C_4
0	1	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	1	1	0	1	1	1	0
0	1	0	0	0	1	1	1	1	0	1	1	1	0
0	1	0	1	1	0	0	1	1	1	1	1	0	1

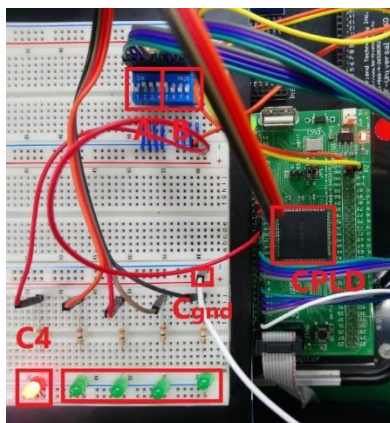


圖 13 前瞻進位加法器實作結果

由表 4 的第一行的數值， $A_4 = 1$ ， $A_3 = 0$ ， $A_2 = 0$ ， $A_1 = 0$ 、 $B_4 = 0$ ， $B_3 = 1$ ， $B_2 = 1$ ， $B_1 = 1$ ，得出的結果為 $C_4 = 0$ ， $S_4 = 1$ ， $S_3 = 1$ ， $S_2 = 1$ ， $S_1 = 1$ ，如上圖 13。

(三) 本小論文研究分析

經過一連串的整理與分析，我們可以藉由整理好的表 5，看出前瞻進位加法器與漣波加法器，得出各位元進位與和所需經過的邏輯閘數量，由表 5 中的列表可得知，從第三位元開始，所需經過的邏輯閘數量明顯增加。

表 5 漣波進位加法器和前瞻進位加法器的比較表（以四位元為例）

名稱	漣波進位加法器			前瞻進位加法器		
運算時輸入到輸出單一路徑所經過邏輯閘數量的比較表		最高位元的進位	最高位元的和		最高位元的進位	最高位元的和
	1 位元	3	2	1 位元	3	2
	2 位元	5	4	2 位元	3	4
	3 位元	7	6	3 位元	3	4
	4 位元	9	8	4 位元	4	4
運算速度	慢			快		

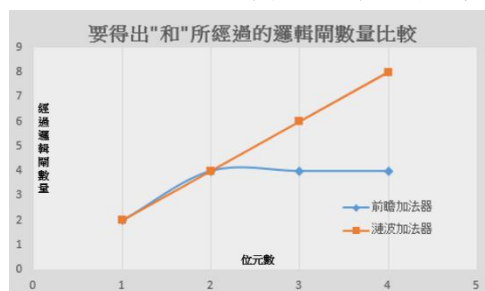


圖 14 經過和的邏輯閘數量比較圖

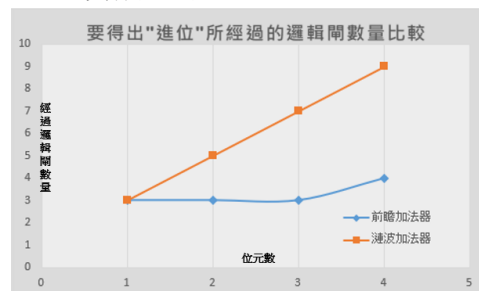


圖 15 經過進位的邏輯閘數量比較圖

圖 14 與圖 15 可以得知，前瞻加法器要得出「和」與「進位」所需經過的邏輯閘數量較漣波加法器的少，故前瞻加法器所造成的延遲時間較少。如果執行加法要快的話，我們可以選擇前瞻加法器，因為漣波加法器的運算時間較長。

參、結論

一、本小論文二種加法器的比較（以四位元加法為例）

漣波加法器的電路，速行運算時速度慢，但運用的邏輯閘數比較少，而前瞻加法器執行運算起起來速度快，但缺點是前整體的邏輯閘數比較器多。

二、問題與解決方法

問題 1:當使用電路繪圖時，找不到有 5 個輸入腳的互斥或閘。

解決方法 1:本小論文可以利用二組 3 個輸入接腳的互斥或閘來替代。

問題 2:運算結果不符

解決方法 2:透過劃出電路圖和手算的資料核對 4 個位元個別的輸出及進位。得出的結果除了第一位原有得出結果其他的位元得出得結果都是亂碼，發現是第 2，3，4 位的進位未接到接地線導致亂碼。

三、研究心得

經果這次的實作與討論，我們更加的了解各種邏輯閘的功能，像是我們了解到同樣一種功能，可以用不同的邏輯閘組合創造出相同的結果並改進其延遲時間，由於邏輯閘運算時會有延遲時間，所以在一個電路中的經過的邏輯閘越多，運算的延遲時間越長，相對的，前瞻加法器經過的邏輯閘較少，故他的運算時間會比漣波加法器來的更加的少。

肆、引註資料

維基百科。加法器-。2020 年 9 月 12 日，取自

<https://zh.wikipedia.org/wiki/%E5%8A%A0%E6%B3%95%E5%99%A8>

黃慶璋（2019）。數位邏輯設計。新北市：全華大圖書 股份有限公司。

劉國棋（2019）。可程式邏輯設計實習。新北市：台科大圖書 股份有限公司。

漣波加法器和前瞻進位加法器的實作與研究

賈證主（2007）。VHDL 數位系統設計與應用。新北市：台科大圖書 股份有限公司。