

進位選擇加法器之設計

謝韶徽、董秋溝、李文益

勤益技術學院電子工程系

摘要

進位選擇加法器 (Carry Select Adder; CSA) 最初的設計是為了可以執行高速加法運算，但是其代價為所使用的電路面積太大，所以，在之後的 CSA 設計大多是討論如何減少 CSA 電路的面積為主。CSA 可由雙漣波進位加法器 (Ripple Carry Adder; RCA) 的電路結構所組成，清華大學張慶元教授 [2] 提出以一個漣波進位加法器與一個「加 1 (Add One) 電路」的電路結構取代。在本文中，提出一種新的加 1 電路及多工器的結構，使得 CSA 在電路面積上可以大幅減少；以 64 位元加法器而言，使用 UMC 0.5 μ m 之製程技術，經由實驗結果得知：本文提出之改良式進位選擇加法器結構較之前的進位選擇加法器結構較之前的進位選擇加法器在電晶體個數上少了將近 13.74% 至 27.17%，而延遲時間則有 2.34% 至 22.33% 獲益。

關鍵詞：進位選擇加法器、計算機算術、加法器設計、算術電路設計

1. 研究背景

加法運算為計算機中最基本也是最重要的運算，因其大量應用於數位信號處理器、濾波器、積分器、乘法器、除法器、高速運算電路等的設計，故加法器 (Adder) 實為一切需要大量計算的處理器之構成基石，故若能提升加法器的效能意即可大幅的改善計算機或運算處理器積體電路的效能。加法器電路設計方式可區分為兩大類：第一類具有結構規則及電路簡單之特性，但是運算複雜度卻與加法運算元之位元數 n 成正比，漣波進位加法器 (Ripple Carry Adder; RCA) 即屬此類；另一類則著重改善運算速度，使得運算複雜度降低為加法運算元之位元數的對數值，但是電路較複雜而且結構不規則，因而需要額外之晶片面積實現此類加法器電路以前瞻式進位加法器 (Carry Lookahead Adder; CLA)、進位選擇加法器 (Carry Select Adder; CSA)、進位跳躍加法器 (Carry Skip Adder; CSK) 等為代表。但此二者電路特性皆與全加器設計之優劣有關，並皆廣泛使用於不同應用目的之高速運算電路設計中。

加法器的速度與面積對所設計的電路有著舉足輕重的影響。RCA 提供了一個規則性的設計，且其在面積的使用方面也很少，但是其本身的延遲時間很長，而 CLA 主要是為了改善 RCA 在延遲時間上的缺點，所提出的一種設計，但其使用的晶片面積太大，所以後來提出的 CSA 及 CSK 都是為了在面積與速度之間取得一個平衡，而兩者在電路的面積與速度上也有著不一樣特性，CSA 的速度較 CSK 快，但所使用的面積較 CSK

大。如表 1 所示，RCA 本身提供一個簡潔的設計，因其晶片面積複雜度為 $O(n)$ ，但其計算複雜度為 $O(n)$ ，需要花費的時間較長。而 CLA 計算複雜度為 $O(\log n)$ ，雖然速度快，但其花費面積很大，其晶片面積複雜度為 $O(n \log n)$ 。由於 RCA 與 CLA 在對於面積及速度上不能兼顧，所以後來才有 CSA 及 CSK 的產生。CSA 與 CSK 此兩種加法器是折衷於 RCA 與 CLA 之間的加法器，就面積而言，CSK 較 CSA 少了將近 35%，而 CSA 在時間上比 CSK 快了 30%。如果，在一個中等執行速度的加法器，使用 CSK 會比使用 CSA 為佳，因為在不要求速度的情況下，當然選擇面積上較佔優勢的 CSK。

表 1 加法器之比較[9]

	計算複雜度	晶片面積複雜度
RCA	$O(n)$	$O(n)$
CLA	$O(\log n)$	$O(n \log n)$
CSA	時間上較 CSK 快了將近 30%	介於 $O(n)$ 與 $O(n \log n)$ 之間
CSK	介於 $O(n)$ 與 $O(n \log n)$ 之間	較 CSA 減少了 35% 的面積

註：在一個中等執行速度的加法器，通常使用 CSK

如圖 1 所示，CSA 的基本概念，乃是將兩組漣波進位加法器分別以進位輸入 (Carry-In; C_{in})信號 $C_{in}=0$ 及 $C_{in}=1$ 做為假設，去平行執行兩組加法運算，產生各自的輸出，再以正確的 C_{in} 信號控制多工器選擇出正確的結果；如此，可以不用像 RCA 一樣等待前級進位，而先行運算結果，便能省下原來等待前級進位的時間。而在 CSA 中最重要的元件為多工器，如果沒有多工器，便無法執行選擇的動作。基本上，只要能完成加法運算功能的電路，便可使用在 CSA 的加法區塊中。目前 CSA 的加法區塊都是使用 RCA 與 CLA 的電路模組。

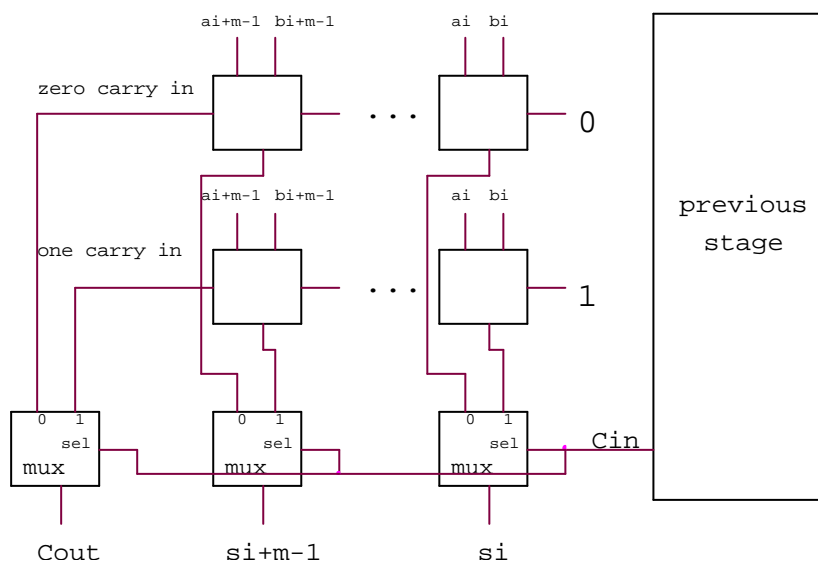


圖 1：基本進位選擇加法器結構圖

2. 進位選擇加法器之設計

2.1 傳統進位選擇加法器

傳統 CSA 是使用兩個相同的加法區塊，分別假設 $C_{in}=0$ 及 $C_{in}=1$ 去進行加法運算，如圖 2 (a) 是一個 CSA 使用雙 RCA 模組的電路，其每一個加法區塊都可以先行假設 C_{in} 去進行運算，然後分別將運算結果送至多工器，等待正確的 C_{in} 來選擇出正確的結果輸出，這樣雖然可以縮短漣波進位的長等待時間，但是，如此的作法其面積會增加至原來的兩倍，所以這種作法不算是最好的，下面就是要介紹另外的電路結構，會使得其電路面積減少至原本的 75 %。

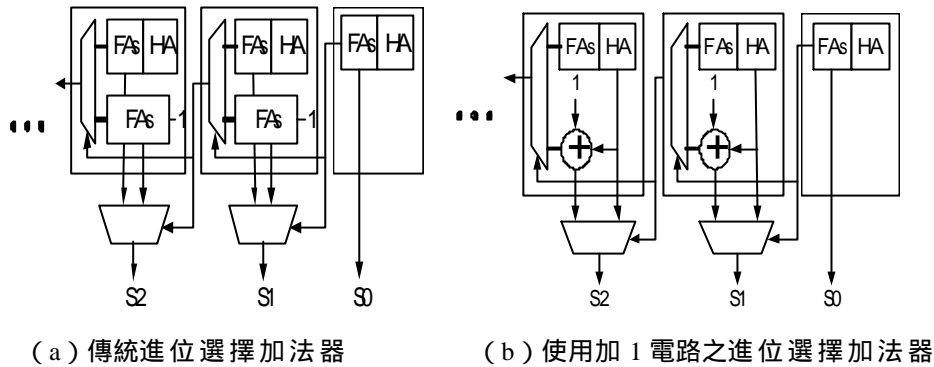


圖 2：進位選擇加法器結構圖[10]

2.2 使用加 1 電路之進位選擇加法器

此方法乃是以一個 RCA 與加 1 電路，去取代原本雙 RCA 的電路，如圖 2 (b)，我們以一個 RCA 去運算 $C_{in}=0$ 的結果，再以加 1 電路將此結果加 1，產生以 $C_{in}=1$ 的運算結果，如此便可完成分別以 $C_{in}=0$ 及 $C_{in}=1$ 的計算，達到 CSA 的動作要求。這是一種新的作法可以不透過加法運算得到 $C_{in}=1$ 的結果，由圖 3 例子可知，加 1 電路是為了使 $C_{in}=0$ 的運算結果加 1，而得到 $C_{in}=1$ 的結果，但是，我們發現如果由最後一個位元開始往前尋找第一個的零(稱為 First Zero)，再將 First Zero 及其較低的位元反相，其結果與 $C_{in}=0$ 經過加法運算加 1 的結果相同，如此便不用透過加法運算得到 $C_{in}=1$ 的結果。First Zero 決定每個位元是否需要被反相，First Zero 找尋電路可由圖 7 中間的 NMOS 及 PMOS 鏈構成。

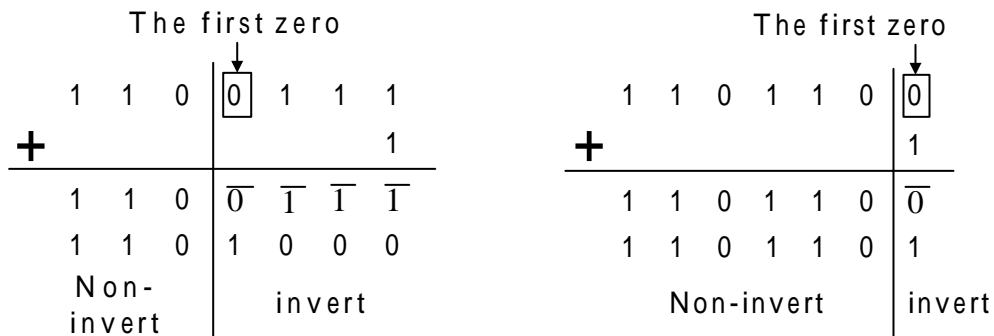


圖 3：First Zero 決定每個位元是否需要被反相的舉例[10]

加 1 電路(Add-One Circuit)如圖 4 中間所標示出來的部份,全加器產生所有的和及補數的和,經由 NMOS 及 PMOS 鏈構成的 First Zero 找尋電路決定出 $C_{in}=1$ 的結果,再由 C_{in} 控制多工器選擇出 S_k (其中 $k=0,1,2,3$)或補數 S_k ,如果,控制訊號為零,多工器選擇 S_k ,否則,多工器選擇補數 S_k 。

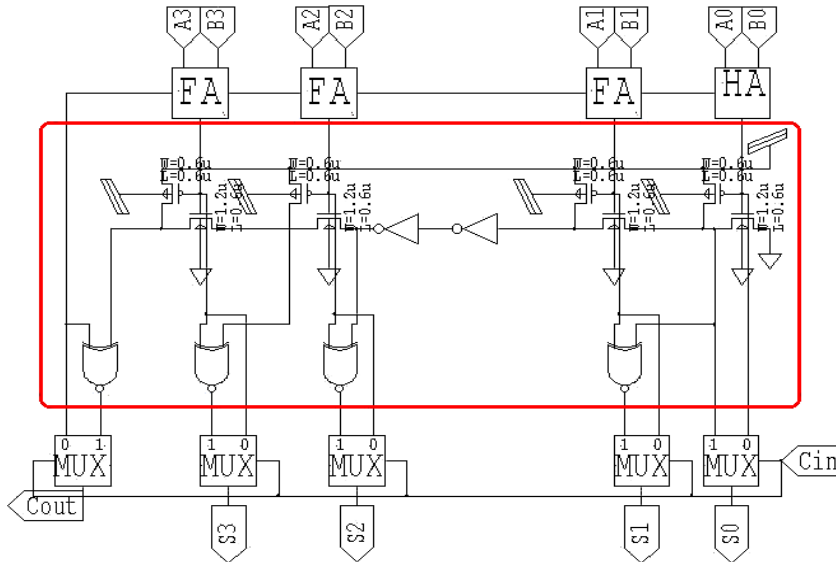


圖 4：使用加 1 電路之進位選擇加法器 [2]

如圖 4 所示,在 CSA 電路中最重要應該是多工器,如果沒有多工器便無法達到「選擇」的動作。而多工器組成的方式有很多,而在清華大學張慶元教授提出之「使用以一個漣波進位加法器(RCA)與加 1(Add One)電路」設計的進位選擇式加法器 [2]所使用便是以 NAND 電閘組成的多工器,如圖 5 中展示,其電路結構及真值表,此種多工器需要使用到 3 個 NAND 電閘及 1 個 NOT 電閘,而將此多工器套用在圖 4 電路中,將會使用到較多的電閘,加上其所有的模組皆是採用全互補式 CMOS 電路設計技巧,所以導致電晶體數量很多,所以在面積的考量上並不算是最理想。

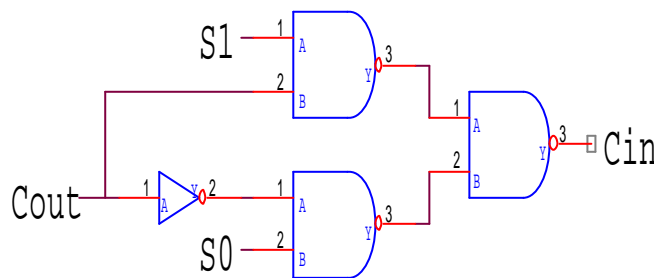


圖 5：由 NAND 電閘組成之多工器

2.3 改良式使用加 1 電路之進位選擇加法器

在圖 4 電路中,所使用到的多工器結構,為雙層多工器(Two Level Multiplexer; TMUX),所以每個位元都需要用到兩倍的多工器面積,因此提出一種以 1 個多工器及 1 的 NAND 電閘來完成這種 TMUX 之結構,如圖 6 所示,其式子推導如下：

$$\begin{aligned}
 S &= (A \cdot Sel1 + B \cdot \overline{Sel1}) \cdot Sel2 + A \cdot \overline{Sel2} \\
 &= A \cdot Sel1 \cdot Sel2 + B \cdot \overline{Sel1} \cdot Sel2 + A \cdot \overline{Sel2} \\
 &= A(Sel1 \cdot Sel2 + \overline{Sel2}) + B \cdot \overline{Sel1} \cdot Sel2 \\
 &= A(\overline{Sel1} + Sel2) + B \cdot \overline{Sel1} \cdot Sel2 \\
 &= A(\overline{Sel1} \cdot Sel2) + B(\overline{Sel1} \cdot Sel2) \quad \text{----- (1)}
 \end{aligned}$$

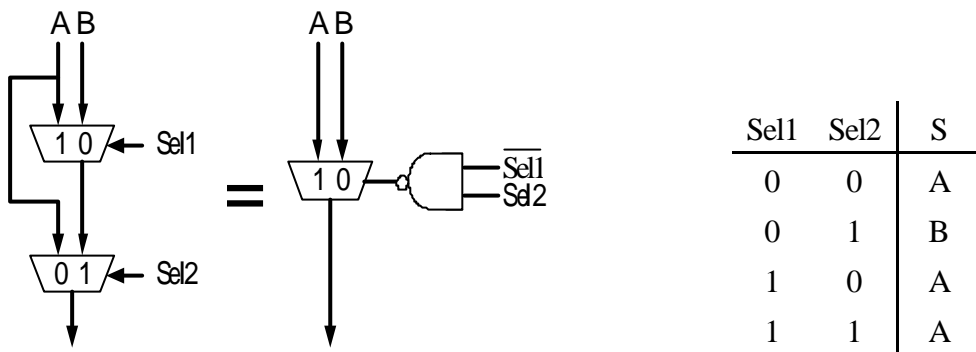


圖 6：雙層多工器結構及真值表

如此，即可將原本 TMUX 等效成 1 個多工器及 1 個 NAND 電閘，圖 6 便是這兩種雙層多工器的結構及真值表，由真值表可知此兩種多工器的功能相同，但第二種 TMUX 其本身的多工器，乃是採用由傳輸閘(Transmission Gate ; TG)組成之多工器，TG 多工器是一種我們所熟悉的多工器，如圖 7 所示為其結構及真值表，所以會較圖 5 的多工器少掉了許多電晶體，會比原本的電路結構更節省面積。將此 TMUX 電路應用在使用加 1 電路之進位選擇加法器中，此便為改良式使用加 1 電路之進位選擇加法器，如圖 8 所示[10]，其電路修正如圖 9 所示，如此，就能保持 Kim's CSA 電路正確動作。

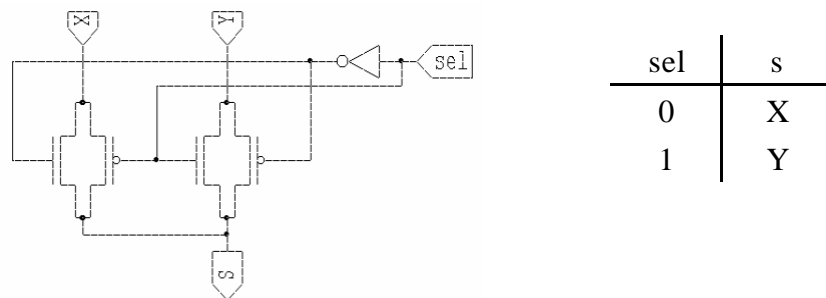


圖 7 由傳輸閘組成之多工器及真值表

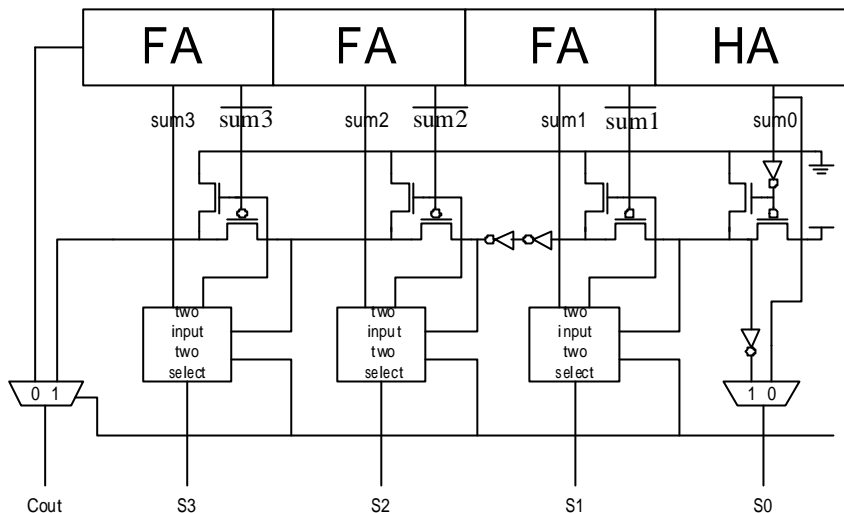


圖 8 原始 Kim's CSA [10]

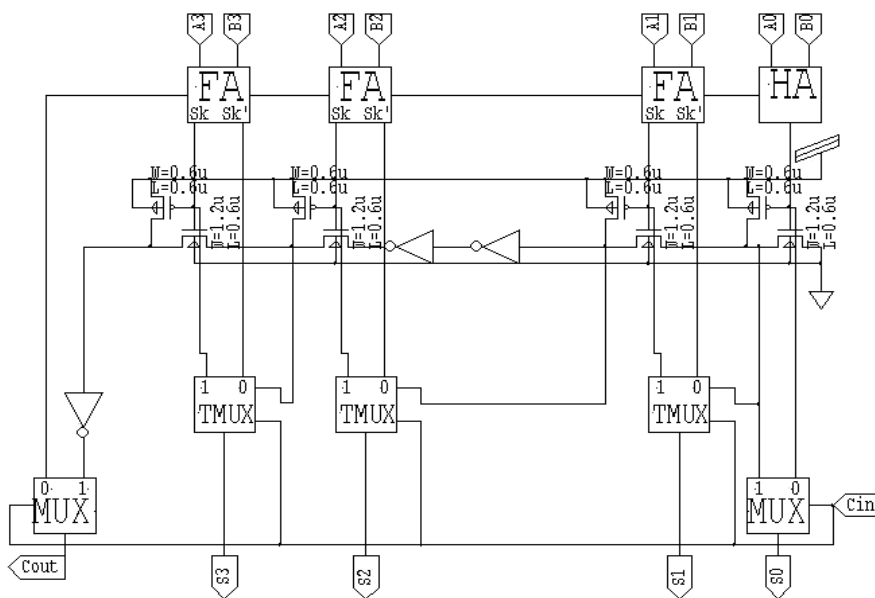


圖 9 修正 Kim's CSA

2.4 提出之改良式進位選擇加法器

我們提出之改良式進位選擇加法器電路，主要將前述兩種進位選擇加法器電路做一個整合，目的是減少進位選擇加法器電路面積，使其電晶體個數減少，其電路如圖 10 所示。而 TMUX 方面，採用通傳電晶體邏輯 (Pass Transistor Logic ; PTL) 設計方式，將電路結構加以改良，如此，原本在 Kim's CSA 需要 12 個電晶體，現在只需要 6 個電晶體就可以完成，如圖 11 電路所示，而使用此電路會使其電晶體個數大幅減少，在總體的效能方面也會提昇許多。

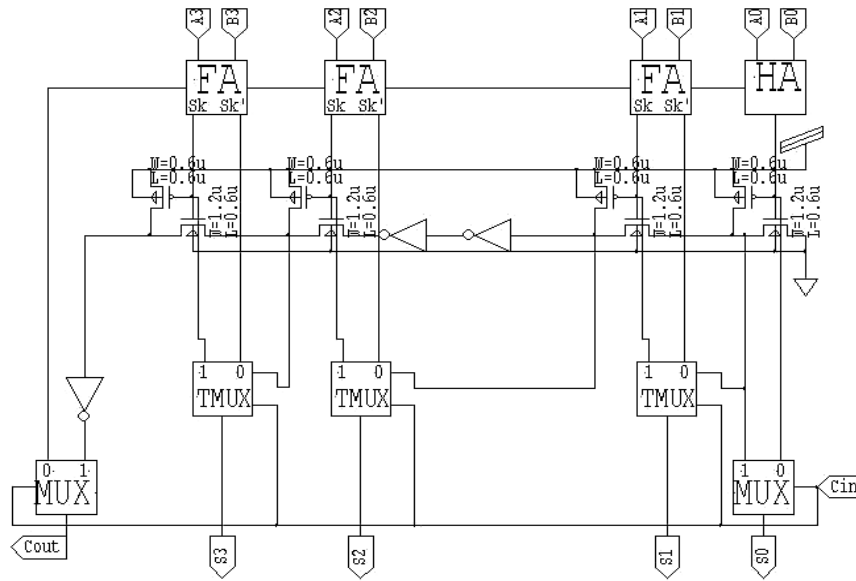


圖 10 提出之改良式進位選擇加法器

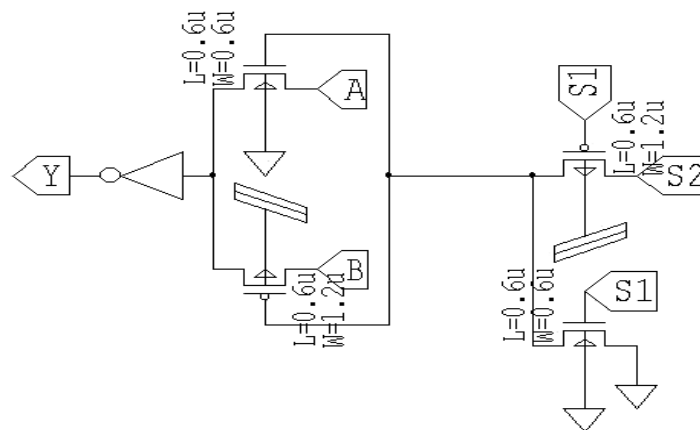


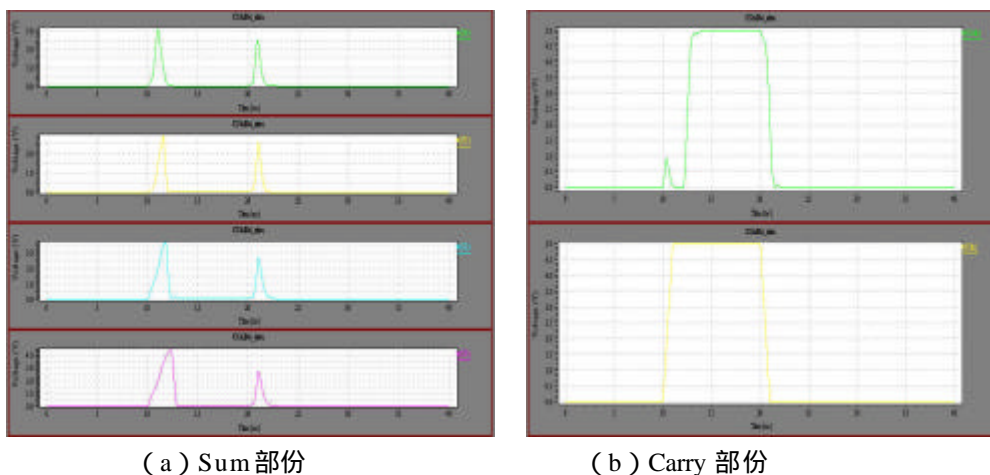
圖 11 提出之改良式 TMUX

3、實驗結果與討論

由於本研究注重的是 CSA 的改良與設計,所以直接改良使用其他研究論文[1][3-8]的電路,並套用至 CSA 電路,以便測量其效能;基於 UMC 0.5um2p2m 製程技術,本文的延遲時間量測皆是以 Worst-Case (亦即所有的 $A_n=1$ 、 $B_n=0$ 及 $C_{in}=1$,使其每一階都有進位之情形產生)的情況去執行測試,其輸出入波型如圖 12 所示

經由我們針對 64 位元進位選擇加法器的測試結果得知,使用加 1 電路之進位選擇加法器 (Chang's CSA)[2]不論在電晶體個數或延遲時間上的考量,都不是很理想,而改良式加 1 電路之進位選擇加法器 (Kim's CSA) [10]的電路結構改變,所以延遲時間較 Chang's CSA 的延遲時間短,但其電晶體個數並未減少很多,我們提出之改良式進位選擇加法器 (Ours Proposed CSA)則同時考量到電晶體個數及延遲時間這兩個因素,所以在

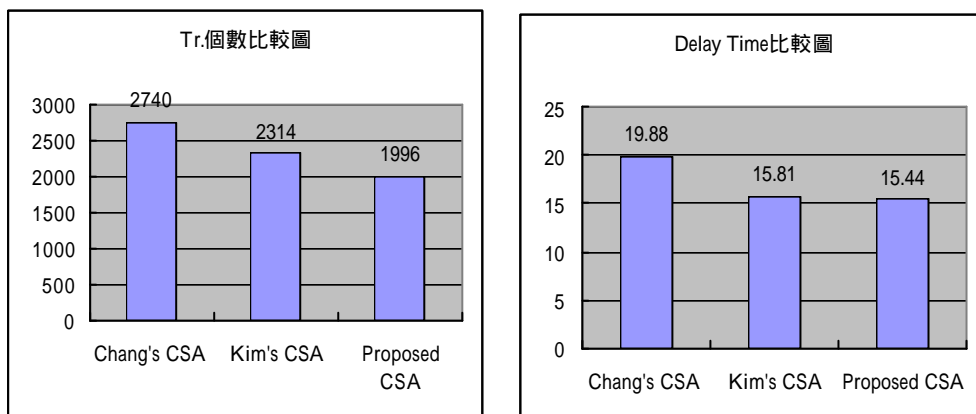
此三種進位選擇加法器的比較上，呈現比較優良的結果，在電晶體個數上減少 13.74% 至 27.17%，而延遲時間則有 2.34% 至 22.33% 獲益，如圖 13 所示。我們提出之改良式進位選擇加法器佈局圖則如圖 14 所示。



(a) Sum 部份

(b) Carry 部份

圖 12：CSA(4-bit)模擬波形



(a) Tr.個數比較圖

(b) Delay Time 比較圖

圖 13：三種進位選擇加法器的比較

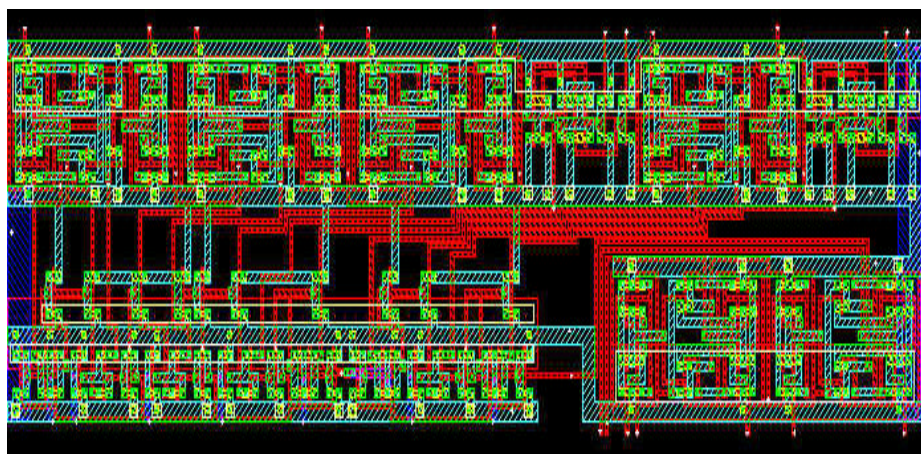


圖 14：我們提出之改良式進位選擇加法器佈局圖

4. 結論

CSA 可由 RCA(Ripple Carry Adder)與一個「加 1 電路」的電路結構取代雙 RCA 的電路結構,而本研究中所提出新的加 1 電路及 Two-Level Multiplexer (TMUX)的結構,使得我們提出之改良式進位選擇加法器電路在電晶體個數及效能上都能提昇,以 64 位元加法器而言,本研究所提出的改良式進位選擇加法器在電晶體個數上少了將近 13.74% 至 27.17%,而延遲時間則有 2.34% 至 22.33% 獲益。

參考文獻

- [1] E. Abu-Shama and M. Bayoumi, "A new cell for low power adders", IEEE International Symposium on Circuits and Systems (ISCAS '96), Vol IV, pp.49 -52, 1996.
- [2] T.-Y. Chang and M.-J. Hsiao, "Carry-select adder using single ripple-carry adder", Electronics Letters, Vol.34, Issue:22, pp. 2101 -2103,1998.
- [3] M. B. Ghaznavi-Ghoushchi and A. R. Nabavi, "Isomorphic structured synthesis of half adder and full adder", Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2002), Vol.2, pp.634 -639, 2002.
- [4] H.Lee and G E Sobelman, "A new low-voltage full adder circuit", Proceedings. Seventh Great Lakes Symposium on VLSI, pp.88 -92, 1997.
- [5] N. Weste and K. Eshraghian, "Principles of CMOS VLSI design, a system perspective", Reading, MA: Addison-Wesley, 1993.
- [6] A.M. Shams and M.A. Bayoumi, "A novel low-power building block CMOS cell for adders", Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98), Vol.2, pp.153 -156, 1998.
- [7] A.M. Shams and M.A. Bayoumi, "A new full adder cell for low-power applications", Proceedings of the 8th Great Lakes Symposium on VLSI, pp.45 -49, 1998.
- [8] D. Soudris, V. Pavlidis, A. Thanailakis, "Designing low-power energy recovery adders based on pass transistor logic", The 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2001), Vol.2, pp.777 -780, 2001.
- [9] A. Tyagi, "A reduced area scheme for carry-select adders", Proceedings 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors, (ICCD '90), pp.255 -258, 1990.
- [10] Youngjoon Kim and Lee-Sup Kim, "A low power carry select adder with reduced area", The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001), Vol4, pp.218 -221, 2001.

