

建造一族群扇出為 2 的最佳平行前置電路 Constructing a Family of Optimal Parallel Prefix Circuits with Fan-Out 2

葉青松 林彥君 吳其政
Ching-Sung Yeh Yen-Chun Lin Chi-Cheng Wu
M8502040@mail.ntust.edu.tw yclin@et.ntust.edu.tw jesse@elc1.lhjc.edu.tw

國立台灣科技大學 電子工程技術研究所
Department of Electronic Engineering
National Taiwan University of Science and Technology, Taipei 106, Taiwan

摘要

前置計算乃是針對 n 個元素 v_1, v_2, \dots, v_n ，以及具有結合性的二元運算 o ，計算出 n 個前置項 $v_1 o v_2 o \dots o v_i, 1 \leq i \leq n$ 。平行前置計算的組合電路稱為前置電路。依據 n 值的不同，在 $2 \lfloor \lg n \rfloor - 1 \leq t \leq n - 2$ 或 $2 \lfloor \lg n \rfloor \leq t \leq n - 2$ 的範圍，存在深度為 t ，且扇出為 2 的最佳前置電路。雖然已可以根據 n 值的不同，建造出深度為 $2 \lfloor \lg n \rfloor - 1$ 或 $2 \lfloor \lg n \rfloor$ ，且扇出為 2 之最佳前置電路，可是還沒有好的方法來建造其他深度的電路。本文提出一個只需 $O(\lg n)$ 時間的方法，來建造出深度 t, t 範圍為 $2 \lfloor \lg n \rfloor - 2 \leq t \leq n - 2$ 或 $2 \lfloor \lg n \rfloor - 1 \leq t \leq n - 2$ ，且扇出為 2 之最佳前置電路。

關鍵字：平行前置計算，最佳前置電路，有限扇出

Abstract

Given n values v_1, v_2, \dots, v_n and an associative binary operation, denoted by o , the prefix problem is to compute the n prefixes $v_1 o v_2 o \dots o v_i, 1 \leq i \leq n$. Combinational circuits for solving the prefix problem are called prefix circuits. It has been shown that there exist optimal parallel prefix circuits with fan-out 2 whose depth can be any integer either between $n - 2$ and $2 \lfloor \lg n \rfloor - 1$ or between $n - 2$ and $2 \lfloor \lg n \rfloor$, depending on the value of n . In this paper, we present an $O(\lg n)$ time method to construct optimal prefix circuits with fan-out 2 whose depth can be any integer t in the range $2 \lfloor \lg n \rfloor - 2 \leq t \leq n - 2$ or $2 \lfloor \lg n \rfloor - 1 \leq t \leq n - 2$.

Keywords: Bounded fan-out, parallel prefix, optimal prefix circuit

This research was supported in part by the National Science Council of the R.O.C. under contracts NSC 86-2213-E-011-025.

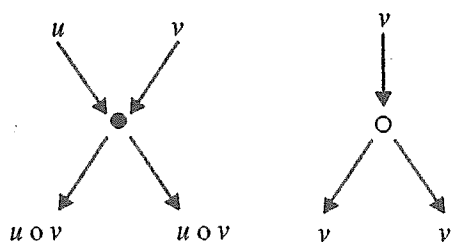
1. 簡介

前置計算 (prefix computation) 乃是針對 n 個元素 v_1, v_2, \dots, v_n ，以及具有結合性的 (associative) 二元運算 (binary operation) o ，計算出 n 個前置項 (prefixes) $v_1 o v_2 o \dots o v_i, 1 \leq i \leq n$ 。前置計算有許多的應用，例如：迴圈的平行化 (loop parallelization)，線性遞迴之求解 (solution of linear recurrences)，進位前看加法 (carry-look-ahead addition)，多項式的求值和插入法 (polynomial evaluation and interpolation)，以及數位濾波 (digital filtering) 等 [1, 4, 6, 8, 13, 15]。因為前置計算的重要性，許多稱為前置電路 (prefix circuits) 的平行組合電路 (combinational circuits) 已被設計出來解決前置計算的問題 [2, 3, 5, 7-12, 14]。在這篇論文中，我們用 n -輸入 (n -input) 前置電路來稱呼輸入個數為 n 的前置電路，並且以 $i:j (i \leq j)$ 表示 $v_i o v_{i+1} o \dots o v_j$ 的計算結果。

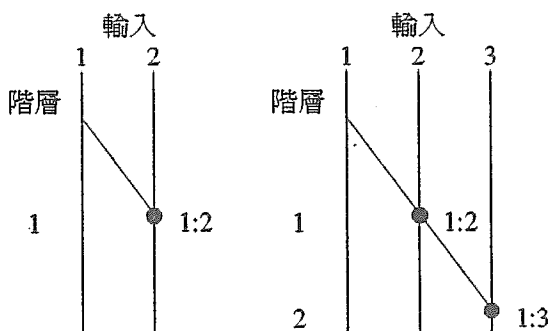
一個 n -輸入前置電路可用有向非循環圖 (directed acyclic graph) 來表示。它具有 n 個輸入節點 (input node)， n 個輸出節點 (output node)，至少 $n - 1$ 個運算節點 (operation node)，甚至於可能有一個以上的複製節點 (duplication node)。如圖一所示，運算節點以實心黑點表示，其入度 (indegree) 與出度 (outdegree) 皆為 2，針對兩個輸入值執行 o 運算，產生兩個相同的輸出。複製節點以空心圓表示，其入度為 1，出度為 2，表示將輸入的資料複製兩份後輸出。因為只有複製節點的入度為 1，出度為 2，所以我們不必特別用空心圓標示複製節點。輸入節點代表一個輸入項目，其入度為 0，出度為 1；而輸出節點代表一個輸出項目，其入度為 1，出度為 0。由於輸入節點與輸出節點很容易辨認，所以不需要特別標示。另外，

圖一中的箭頭表示資料傳送的方向；由於本文中全部的箭頭方向都是向下的，為了簡單起見，我們不再標示出箭頭。

圖二顯示 2-輸入以及 3-輸入的循序前置電路 (serial) 前置電路。圖左邊的數字分別表示其右邊運算節點的深度階層，圖上方整數 i 的下面為輸入節點，表示輸入 v_i ；以 3-輸入循序前置電路中的 1:2 為例，表示其左側節點計算出 1:2 的結果。我們用 $S(n)$ 表示 n -輸入循序前置電路。節點的扇出 (fan-out) 等於它的出度。如果一個節點的扇出不是固定的，而是 n 的函數，則稱此節點的扇出為未限扇出 (unbounded fan-out)。整個前置電路的扇出，為所有節點中扇出的最大值。如果一前置電路的扇出為 2，表示此前置電路每一節點的扇出最大值為 2；如果一前置電路的扇出為未限扇出，表示此前置電路中最少有一個節點的扇出為未限扇出。在超大型積體電路中，如果扇出越大，則所佔的面積會越大，而且速度也會越慢 [16]；所以，扇出最好是有限扇出 (bounded fan-out)，而且越小越好。



圖一：運算節點與複製節點



圖二：2-輸入與3-輸入的循序前置電路

n -輸入前置電路 $G(n)$ 的節點數，定義為 $G(n)$ 中運算節點的個數，用 $s(G(n))$ 表示； $G(n)$ 的深度 (depth) 定義為 $G(n)$ 中任一有向路徑上，運算節點個數的最大值，用 $d(G(n))$ 表示。例如：圖二中 3-輸入循序前置電路的 $s(G(n)) = d(G(n)) = 2$ 。Snir 已經證明了 $d(G(n)) + s(G(n)) \geq 2n -$

2 [14]。因此，當 $d(G(n)) + s(G(n)) = 2n - 2$ ， $G(n)$ 為深度與節點數最佳的前置電路。在深度固定的情況下，如果一個前置電路有最小的節點數，則此電路為最佳前置電路 (optimal prefix circuit) [5]。所以，深度與節點數最佳的前置電路，也是最佳前置電路。

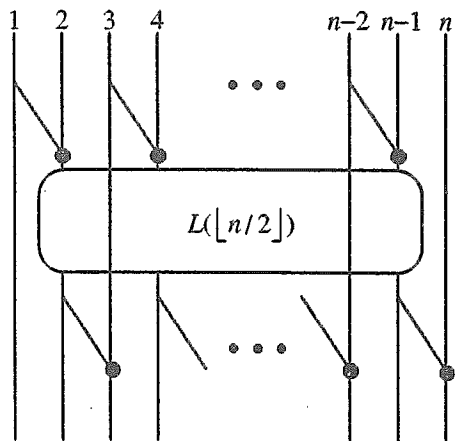
當 $n \geq 2$ ，依據不同的 n 值，在 $2 \lfloor \lg n \rfloor - 1 \leq t \leq n - 1$ 或 $2 \lfloor \lg n \rfloor \leq t \leq n - 1$ 的範圍，存在著一族深度為 t ，且扇出為 2 的最佳前置電路 [10]。而且，已有方法可建造出深度為 $2 \lfloor \lg n \rfloor - 1$ 或 $2 \lfloor \lg n \rfloor$ ，且扇出為 2 之最佳平行前置電路 $L(n)$ [10]。另外，循序前置電路 $S(n)$ 即是深度為 $n - 1$ 的最佳前置電路。可是，要建造該族群中其他深度的電路，在此之前卻沒有簡單又快的方法。

本文提出一個有系統的方法，只要花 $O(\lg n)$ 的時間，即可建造出一個扇出為 2 的 n -輸入平行最佳前置電路，其深度 t 可以是在範圍 $2 \lfloor \lg n \rfloor - 1 < t < n - 1$ 或 $2 \lfloor \lg n \rfloor < t < n - 1$ 。第 2 節將簡單介紹平行前置電路 $L(n)$ 。第 3 節則介紹如何利用 $L(n)$ 與 $S(n)$ 連接成一族群深度介於 $L(n)$ 與 $S(n)$ 的深度之間的最佳平行前置電路。第 4 節為結論。

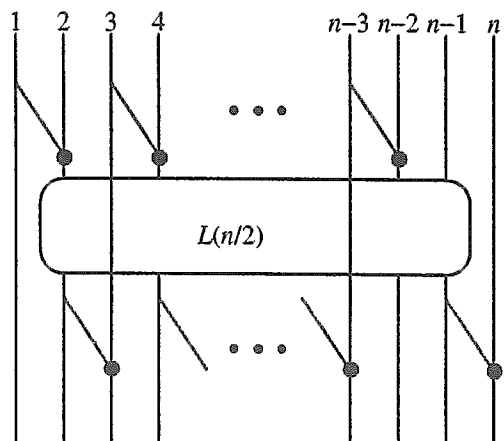
2. 扇出為 2 的最佳前置電路

這一節我們簡單介紹論文 [10] 如何建造出深度為 $2 \lfloor \lg n \rfloor - 1$ 或 $2 \lfloor \lg n \rfloor$ 之 n -輸入最佳平行前置電路 $L(n)$ 。令 $L(2)$ 及 $L(3)$ 分別為循序前置電路 $S(2)$ 與 $S(3)$ ，已見於圖二。以這兩個前置電路為雛型，經由遞迴的 (recursive) 方式，可建造出 $n > 3$ 的平行前置電路。當 n 為奇數且 $n > 3$ 時，圖三表示平行前置電路 $L(n)$ 的建造方法，圖四則是 n 為偶數且 $n > 3$ 時，平行前置電路 $L(n)$ 的建造方法。在圖三， $L(n)$ 的第一層電路包含 $\lfloor n/2 \rfloor$ 個運算節點，其中第 i 個節點計算出 $(2i-1):2i$ 的值。這 $\lfloor n/2 \rfloor$ 個運算節點的輸出再送到電路 $L(\lfloor n/2 \rfloor)$ ，以輸出 $1:4, 1:6, \dots, 1:(n-1)$ 。 $L(n)$ 電路的最後一層也有 $\lfloor n/2 \rfloor$ 個運算節點，用來計算出 $1:3, 1:5, \dots, 1:n$ 。在圖四， $L(n)$ 的第一層電路包含 $n/2 - 1$ 個運算節點；第一層運算節點的輸出與 $L(n)$ 的第 $n - 1$ 個輸入是 $L(n/2)$ 的輸入。 $L(n/2)$

輸出 $1:4, 1:6, \dots, 1:(n-4), 1:(n-2)$, 以及 $1:(n-1)$ 。 $L(n)$ 電路的最後一層也有 $n/2 - 1$ 個運算節點，用來計算出 $1:3, 1:5, \dots, 1:(n-5), 1:(n-3)$, 以及 $1:n$ 。



圖三： n 為奇數且 $n > 3$ 時的平行前置電路 $L(n)$



圖四： n 為偶數且 $n > 3$ 時的平行前置電路 $L(n)$

論文 [10] 證明了 $L(n)$ 為最佳前置電路，其深度為：

$$2 \lfloor \lg n \rfloor - 1$$

若 $2^i \leq n < 3 \times 2^{i-1}$ 且 $i \geq 1$;

$$2 \lfloor \lg n \rfloor$$

若 $3 \times 2^{i-1} \leq n < 2^{i+1}$ 且 $i \geq 1$ 。

另外，當 $n \geq 2$ 且 t 的範圍如下時，

$$2 \lfloor \lg n \rfloor - 1 \leq t \leq n - 1$$

若 $2^i \leq n < 3 \times 2^{i-1}$ 且 $i \geq 1$;

$$2 \lfloor \lg n \rfloor \leq t \leq n - 1$$

若 $3 \times 2^{i-1} \leq n < 2^{i+1}$ 且 $i \geq 1$ 。

已被證明存在 n -輸入，深度為 t ，且扇出為 2 的最佳前置電路。可是，除了深度為

最大之 $n - 1$ 與最小之 $2 \lfloor \lg n \rfloor - 1$ (或 $2 \lfloor \lg n \rfloor$) 的電路以外，尚未有簡單又快速的設計方法。第 3 節將提出一個只要 $O(\lg n)$ 時間的簡單方法，來建造上述最佳前置電路。

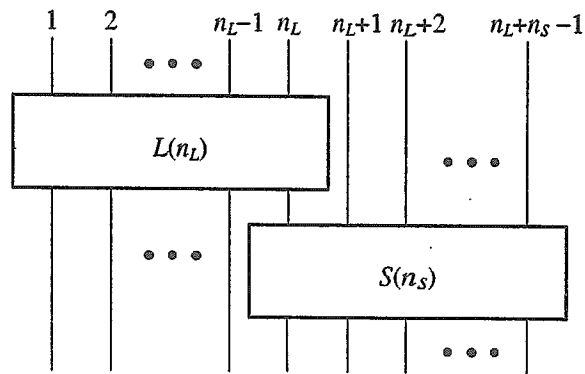
3. 建造方法

當 $d(L(n)) < t < n - 1$ 時，可以將一個平行前置電路 $L(n_L)$ 和一個循序前置電路 $S(n_S)$ 合成起來，使 $L(n_L)$ 的最後一個輸出是 $S(n_S)$ 的第一個輸入，成為深度為 t ，且扇出為 2 的最佳平行前置電路 $H(n)$ ，表示成 $H(n) = L(n_L) + S(n_S)$ ，其中

$$n_L + n_S - 1 = n,$$

$$d(L(n_L)) + d(S(n_S)) = t,$$

如圖五所示。由於 $L(n_L)$ 是最佳前置電路，且 $S(n_S)$ 也是最佳前置電路，所以合成的 $H(n)$ 也是最佳前置電路 [14]。



圖五：合成 $L(n_L)$ 與 $S(n_S)$ ，成為 $H(n)$ 前置電路

建造 $H(n)$ 可以使用下列方式：

1. 算出平行前置電路 $L(n)$ 的深度 $d(L(n))$ ，並計算 $dd := t - d(L(n))$ ， $n_S := dd + 1$ ， $n_L := n - dd$ 。
2. 如果 $d(L(n_L)) = d(L(n))$ ，則 $d(L(n_L)) + d(S(n_S)) = d(L(n)) + dd = t$ ，故確定 $H(n) = L(n_L) + S(n_S)$ ，因而結束。否則表示 $d(L(n_L)) < d(L(n))$ ，要執行步驟 3。
3. 循序前置電路的深度加 1，即 $n_S := n_S + 1$ 。
4. 平行前置電路的輸入個數減 1，即 $n_L := n_L - 1$ 。
5. 若 $d(L(n_L)) + d(S(n_S)) = t$ ，則 $H(n) = L(n_L) + S(n_S)$ ；否則，回到 3。

假設步驟 3、4 執行的次數為 $offset$ ，則顯然：

$$n_s = dd + 1 + \text{offset} \quad (1)$$

由於

$$d(L(n_L)) + d(S(n_s)) = t = d(L(n)) + dd \circ$$

所以，

$$d(L(n)) - d(L(n_L)) = d(S(n_s)) - dd \circ$$

再利用 (1)，可得

$$\text{offset} = n_s - 1 - dd = d(S(n_s)) - dd \circ$$

所以，

$$\text{offset} = d(L(n)) - d(L(n_L)) \circ$$

前面已提及 $d(L(n)) = 2 \lfloor \lg n \rfloor$ 或 $2 \lfloor \lg n \rfloor - 1$ ，因此

$$d(L(n)) \leq 2 \lfloor \lg n \rfloor,$$

以及

$$d(L(n_L)) \geq 2 \lfloor \lg n_L \rfloor - 1 \circ$$

所以，

$$\text{offset} \leq 2 \lfloor \lg n \rfloor - (2 \lfloor \lg n_L \rfloor - 1) = O(\lg n) \circ$$

因此，建造 $H(n)$ 需時間 $O(\lg n)$ 。

4. 結論

在這篇論文中，我們提出了一個方法，可以根據 n 的值，建造出深度為 t ，其中 t 的範圍為 $2 \lfloor \lg n \rfloor - 1 < t < n - 1$ 或 $2 \lfloor \lg n \rfloor < t < n - 1$ ，且扇出為 2 之 n -輸入最佳平行前置電路。該電路是由平行前置電路 $L(n_L)$ 與循序前置電路 $S(n_s)$ 所合成，其中 $n = n_L + n_s - 1$ 。我們的方法只要花 $O(\lg n)$ 的時間即可找出 n_L 與 n_s 之值。

參考資料

- [1] S.G. Akl, *The Design and Analysis of Parallel Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [2] R.P. Brent and H.T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, pp. 260-264, Mar. 1982.
- [3] D.A. Carlson and B. Sugla, "Limited width parallel prefix circuits," *J. Supercomput.*, vol. 4, pp. 107-129, June 1990.
- [4] R. Cole and U. Vishkin, "Faster optimal parallel prefix sums and list ranking," *Infom. Contr.*, vol. 81, pp. 334-352, 1989.
- [5] F.E. Fich, "New bounds for parallel prefix circuits," in *Proc. 15th Symp. on the Theory of Computing*, 1983, pp. 100-109.
- [6] C.P. Kruskal, T. Madej, and L. Rudolph, "Parallel prefix on fully connected direct connection machines," in *Proc. Int. Conf. on Parallel Processing*, 1986, pp. 278-284.
- [7] R.E. Ladner and M.J. Fischer, "Parallel prefix computation," *J. ACM*, vol. 27, pp. 831-838, Oct. 1980.
- [8] S. Lakshmivarahan and S.K. Dhall, *Parallel Computing Using the Prefix Problem*. Oxford, UK: Oxford University Press, 1994.
- [9] S. Lakshmivarahan, C.M. Yang, and S.K. Dhall, "On a new class of optimal parallel prefix circuits with (size + depth) = $2n - 2$ and $\lfloor \log n \rfloor \leq \text{depth} \leq (2 \lfloor \log n \rfloor - 3)$," in *Proc. 1987 Int. Conf. on Parallel Processing*, 1987, pp. 58-65.
- [10] Y.C. Lin, "Optimal prefix circuits with fan-out 2," in *Proc. Int. Conf. on Algorithms*, 1996, pp. 175-181.
- [11] Y.C. Lin and C.M. Lin, "Efficient parallel prefix algorithms on fully connected message-passing computers," in *Proc. 3rd Int. Conf. on High Performance Computing*, 1996, pp. 316-321.
- [12] Y.C. Lin and C.M. Lin, "A family of efficient algorithms for the prefix operation on message-passing computers," in *L.N.C.S. #1067, Proc. Int. Conf. and Exhibition on High-Performance Computing and Networking*, 1996, pp. 1003-1004.
- [13] A. Nicolau and H. Wang, "Optimal schedule for parallel prefix computation with bounded resources," in *Proc. Third ACM SIGPLAN Symp. on Principles & Practice Parallel Programming*, 1991, pp. 1-10.
- [14] M. Snir, "Depth-size trade-offs for parallel prefix computation," *J. Algorithms*, vol. 7, pp. 185-201, 1986.
- [15] H. Wang, A. Nicolau, and K.S. Siu, "The strict time lower bound and optimal schedules for parallel prefix with resource constraints," *IEEE Trans. Comput.*, vol. 45, pp. 1257-1271, Nov. 1996.
- [16] N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1993.