

放鬆波蘭式用於大量模組之平面規劃

方志鵬 簡宗宇 蘇哲彥

國立台北科技大學電機所

jpfang@ntut.edu.tw

論文摘要

用於超大型積體電路平面規劃的表示法有兩種：切割法與非切割法。切割法只能表達切割結構，而非切割法可以表達切割或非切割結構。但有幾個原因使切割法仍具有存在的價值：1、容易實作；2、以切割法得出的電路平面仍需以切割法維護；3、使用事後壓縮的方式一樣可以得出非切割結構的電路平面。切割法以 H 及 V 表示平面切割方向，且典型以波蘭式實作。我們提出放鬆波蘭式，以 O 取代 H 及 V 來放鬆對平面切割方向的限制。另外，我們還提出局部最佳合併的演算法來搭配放鬆波蘭式。實驗結果顯示，此種新方法既能有效減短計算的時間，結果也近似於非切割法。

關鍵詞：平面規劃、切割法、非切割法、放鬆波蘭式、模擬退火

Abstract

The methods to represent a floorplan of VLSI can be classified as slicing and non-slicing. The non-slicing method can express slicing structure as well as non-slicing structure while the slicing method can express slicing structure only. However, there are reasons that make the slicing method survive: (1) Slicing method is simple in terms of implementation; (2) A floorplan designed with slicing method should be maintained accordingly; (3) After compression, a slicing floorplan becomes a non-slicing floorplan. Typically, a slicing method uses 'H' and 'V' to represent the orientation of cut, and uses a Polish expression to represent a slicing floorplan. Instead of 'H' and 'V', we use 'O' in our relaxed Polish expression to relax the constraint on the orientation of cut. In addition, we propose a greedy merging algorithm to work with the proposed relaxed Polish expression. The experimental results show that the proposed methods work efficiently and the result is compatible to that of the existing non-slicing methods.

Keywords: Floorplanning, Slicing, Non-slicing, Relaxed Polish expression, Simulated Annealing

前言

超大型積體電路(VLSI)進行平面規劃時，會以合適的資料表示法及演算法將模組的組合面積及線長降低至某個範圍，並滿足其它限制條件。現今已有相當多的平面規劃表示法，大致上可以分類為兩種主要的方法。

其一為切割(slicing)法，根據 W&L [1]，使每個模組在組合過程當中以水平或是垂直的一條線

切割，並且以波蘭式(Polish expression)去排列各個模組之間的關係。其二為非切割(non-slicing)法，包括 sequence pair [2]、GPE [3]、O-tree [4]、B*-tree [5]、CBL [6]、TCG [7]均被歸類為此種方法。

非切割法尚未問世前，切割法是平面規劃的主要作法。非切割法可以有效地利用全部的空間，但是它和切割法不相容，故無法直接處理切割法所產生的檔案。相對於非切割法，切割法沒辦法有效地利用全部的空間，但卻容易實作，而且如果進一步對切割法得出的結果進行壓縮，它也可以得出和非切割法一樣的結果。所以切割法是學生學習實體設計演算法時常用的方法，也是業者無法廢棄的方法。

本文中，我們針對平面規劃之切割法，提出一種放鬆波蘭式(relaxed Polish expression)，再以局部最佳合併法運算各個模組組合後的面積。除了縮短整體的計算時間外，計算後所得到的面積大致相等於非切割法得出的結果。

接下來的內容中，首先解釋放鬆波蘭式原理，接著介紹局部最佳合併法，然後說明實驗環境與實驗結果，最後為結論。

放鬆波蘭式原理

在一維空間中，有若干個矩形模組，每個模組有各自的長度與寬度，模組相互緊密相鄰且不重疊，如何組合方能使面積為最小？這種模組與模組之間的組合問題為 NP-complete 的問題。

以下我們將介紹切割結構[1]中的波蘭表示法，並以圖 1 說明放鬆波蘭式的原理。

二維之規劃平面以 W&L [1]的方式表示其切割結構時，可以用切割樹也可以用波蘭式來表達。波蘭式中的元素可以區分成兩類，其一為運算子，代表水平或是垂直切割線；其二為運算元，為各個模組的代碼或是名稱。

其中圖 1(a)為由編號 1、2、3、4 等模組組合而成的規劃平面；圖 1(b)區分為 1、2 模組組合以及 3、4 模組組合，此兩組合模組是以水平切割線隔開的上、下關係，波蘭式以 H 表示；圖 1(c)進一步區分 1 與 2 模組以及 3 與 4 模組，均為左、右的關係，波蘭式中以 V 表示垂直切割線；這種模組的組合關係可以用切割樹表示，如圖 1(d)所示，也可以表示成波蘭式，圖 1(d)之波蘭式為 {1,2,V,3,4,V,H}。

圖 1(d)中，波蘭式依相鄰模組間為水平關係或是垂直關係來進行垂直切割或是水平切割。但在放鬆波蘭式中，我們取消相鄰關係的限制，改以無方

向性的切割線 O 取代垂直切割線(H)以及水平切割線(V)。換言之，圖 1(d)的放鬆波蘭式為 {1,2,O,3,4,O,O}。

平面規劃的問題是 NP-complete 的複雜度，所以平面規劃的表示法都會以模擬退火之類的演算法進行擾動(purterbation)，以得出較佳解。而擾動時的鄰接結構(neighboring structure)越簡單，解空間就越小，因此得以用較少次的擾動得出收斂解。因為放鬆波蘭式可以簡化鄰接結構，故可明顯縮小解空間，從而使求解過程快速收斂。

但簡化鄰接結構也產生兩個問題：1、無法確保不產生 non-skew 的模組組合；2、計算成本函數(cost function)時(本論文的成本函數為規劃平面的面積)會耗用更多時間及更多記錄空間。以下將說明我們對這兩個問題的處理方式。

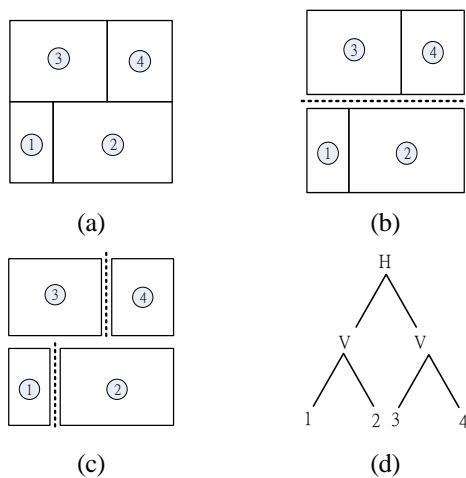


圖 1. 切割結構

局部最佳合併法

在波蘭式中，為了避免冗餘解，規定垂直切割線與水平切割線不可以連續出現，也就是要排除 non-skew 的現象。因此在進行模擬退火程序時，每擾動一次就需先判斷，以避免產生 non-skew 的波蘭式。

但擾動次數很多時，non-skew 的判斷其實也需要耗用不少時間。而且使用 W&L [1]計算面積的演算法時，也非常浪費時間。所以在放鬆波蘭式中，我們不去判斷 non-skew 的現象，而且為了改善 W&L [1]計算面積時大量耗用時間及記憶體的問題，我們改採局部最佳化演算法，以快速計算出每個放鬆波蘭式的面積。

我們以範例說明 W&L [1]計算面積的演算法對計算時間及計算空間不利之處，進而闡述局部最佳合併法的必要性及優點。

如圖 2 所示，假定各模組寬度與長度為：模組 1 (4x2)、模組 2 (8x4)、模組 3 (6x6)、模組 4 (6x4)。W&L [1]計算模組組合面積的方法是由相鄰的模組組合成較大模組，再由較大模組組合成更大模組，如圖 3 所示。每次組合時各模組均嘗試旋轉不

同方位，以計算所有可能的組合面積，並排除明顯較差的解(圖中畫紅叉者)，由樹葉依序往樹根計算，最後得出面積最小者(圖中框紅圈者)。

所以 2 個模組組合時，在最壞情況下必需記錄 3 種組合並分別計算其面積，而較大模組要組合成更大模組時，記錄其組合數所耗用的空間及計算不同組合面積所耗用的時間會呈 $O(n^2)$ 成長(n 為前一層所產生的組合個數)。

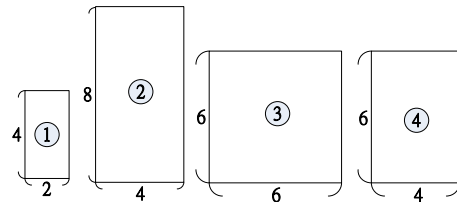


圖 2. 範例中各模組的寬度與長度

非切割法[2-7] 其實也有旋轉模組的問題，但通常它們解決的方法是在擾動時新增一種名為「旋轉」的鄰接結構。但這樣的解決方案雖然容易實作且縮短每次擾動後計算成本函數的時間，但也產生解空間大幅膨脹的缺點。

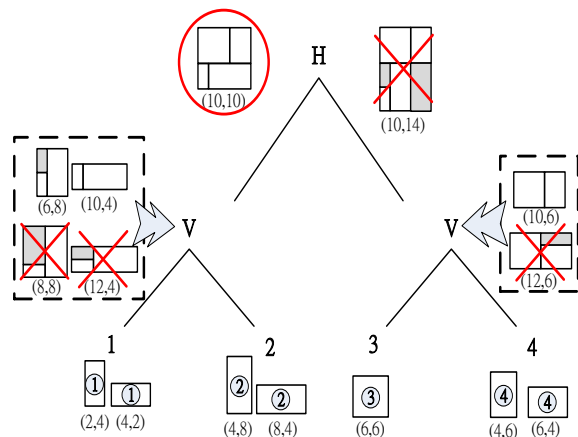


圖 3. W&L [1]計算面積的方式

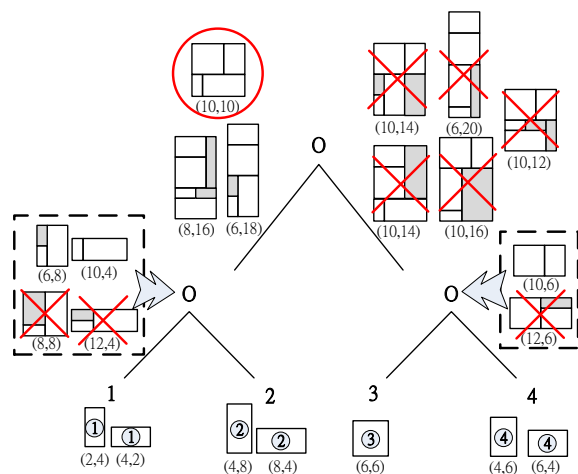


圖 4. 放鬆波蘭式使組合更多元

在放鬆波蘭式中，我們嘗試以 W&L 的方法 [1] 排列出各個模組面積的組合，如圖 4 所示，可以看

到根節點需記錄 8 種組合，排除 5 種明顯較差者，尚需計算 3 種組合的面積，才能得出最佳面積為 10×10 。雖然在放鬆波蘭式中解除了 V 與 H 的限制，讓解空間變小，但因為模組的組合更多元，所以會使每次擾動後的面積計算時間及記錄組合模組所需的記錄空間更大。

我們進一步以 GSRC 的不同測試電路為例，用初始解僅作一次擾動，分割樹的樹根所看到的組合模組個數及計算時間，如表 1 所示。

表 1 中，我們可以發現，隨著模組數越來越大的時候，光是一次擾動，計算面積耗費的時間是以倍數在成長，但因為計算過程中有很多明顯較差的組合，所以組合的數量僅緩慢的提升。不過，這也足以表示 W&L [1] 計算面積的方法雖然可以縮小解空間，但每一個解的計算時間太長了。

表 1. GSRC benchmarks

	Time(sec)	Combinations
n10	0.031	168
n30	8.218	855
n50	30.734	1171
n100	140.375	1625
n200	624.39	2320
n300	2210.69	3572

因為當組合模組時，隨著模組之旋轉與否，會產生多種結果，過濾明顯較差的答案後，然後再進一步與其它組合過的模組進一步組合，會產生出更多種結果。當組合變化結果越來越多的時候，使得計算越來越煩雜；所以我們在此提出局部最佳合併法，使組合結果只會產生一種答案，讓每次的放鬆波蘭式可以快速的運算。

局部最佳合併法是將兩個模組，依直立(S, Stand)及旋轉(R, Rotate)兩種情況組合成 SS、SR、RS、RR 四種，取其中一種組合面積最小的答案，如圖 5 所示。

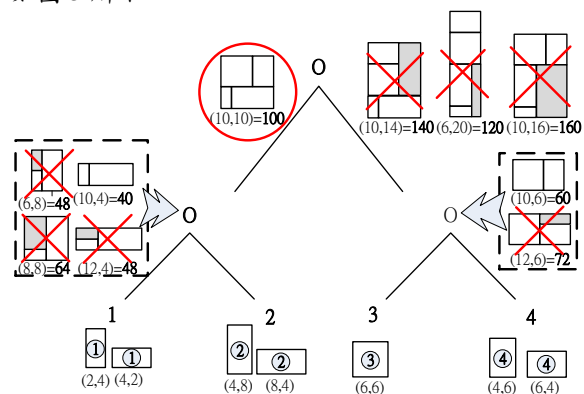


圖 5. 放鬆波蘭式+局部最佳合併法

由圖 5 可以看到分割樹的每一個節點只需記錄 4 種組合，排除其中 3 種。而在樹根也一樣，所

以能算出面積為 10×10 。因為放鬆波蘭式解除 V 和 H 限制，讓各個模組可在每次組合時依面積的計算決定該直立或旋轉。而對組合模組而言，一旦決定了組合模組是直立或旋轉，其實也決定了形成該組合模組之兩模組的上下組合或左右組合關係。因此計算時間明顯大幅縮短，而且也不再記錄多種組合所衍生的耗用空間的問題。

實驗結果

實驗環境描述如下：CPU 為 AMD Phenom(tm) II X6 1090T Processor, 3.21GHz，記憶體為 3 GB DDRⅢ，實驗平台為 Microsoft Windows XP，編譯器為 Microsoft Visual Studio 2010，使用 C# 語言。測試電路包括 MCNC 的 5 個測試檔(apte、hp、xerox、ami33、和 ami49)及 GSRC 的 6 個測試檔(n10、n30、n50、n100、n200、n300)。

我們以類似 W&L [1] 的模擬退火演算法進行實驗，但因為放鬆波蘭式及局部最佳合併法的特性，我們的模擬退火演算法和 W&L [1] 的差異如下所述。

1、W&L [1] 的模擬退火演算法中，有 3 種鄰接結構：(1) 交換運算元、(2) 將運算子串反相、和 (3) 交換運算元/運算子。但在我們的方法中只需 (1) 與 (3) 兩種鄰接結構。

2、W&L [1] 的方法需要顧慮到當運算子與運算元進行交換時，是否有發生 non-skew 的現象。而在我們的方法進行模擬退火的擾動時，不浪費時間去判斷。

我們針對 11 個 benchmark 進行測試，圖 6 至圖 9 是其中較具有代表性的結果。每個圖包含兩部份：其一是模擬退火過程中面積相對於時間的波形圖，浮貼於右上方的則是最佳結果之規劃平面圖。波形圖中的黑點表示成本函數為負值(面積變小)，且為了顯示退火過程的變化，每個測試電路的退火均強制降至最低溫。規劃平面圖中深灰色區域代表無用空間(dead space)。

由圖 6 至圖 9 的波形圖可看出，隨著模組數量的增加，得出最佳解所需的擾動次數也隨之增加。

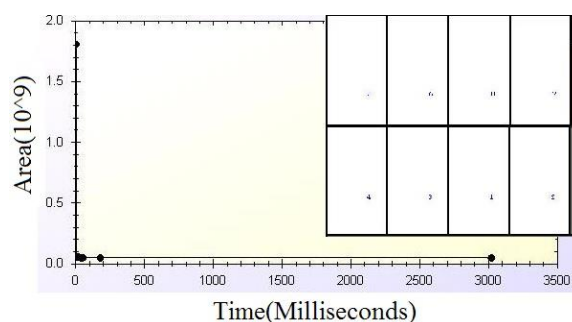


圖 6. Apte

另外表 2 及表 3 則是我們在切割結構所用的方法和現有非切法結構演算法在速度及面積的比較。表 2 的測試對象是 MCNC 的 5 個測試電路，比較的對象是 B*-tree[5] 和 FAST-SP[8]；表 3 的測

試對象是 GSRC 的 6 個測試電路，比較的對象是 MBS-OEA[10]和 ACG[12]。其中標示為 A 之欄位為面積，單位為 mm^2 ；而標示為 T 之欄位代表時間，單位為秒；資料項中標示為 NA 者代表無法取得該項資料。

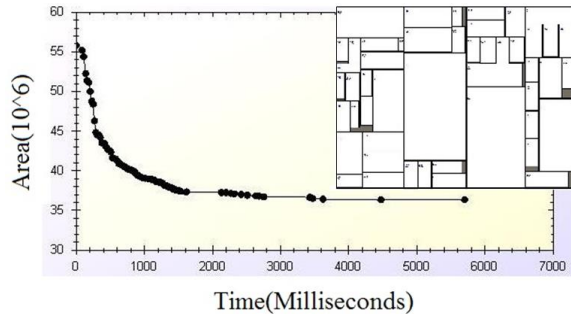


圖 7. Ami49

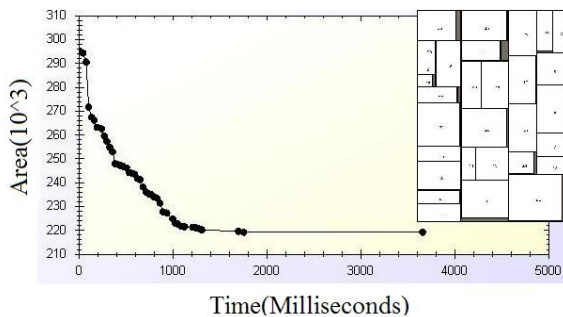


圖 8. N30

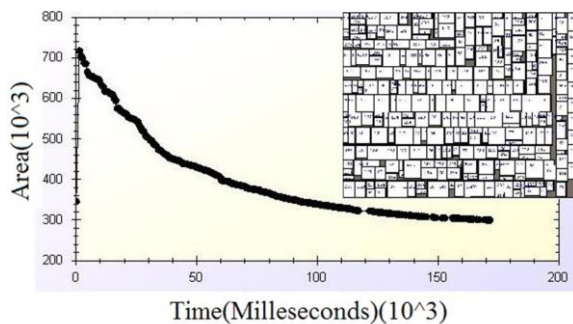


圖 9. N300

表 2. MCNC 測試檔的性能比較

	B*-tree[5]		FAST-SP[8]		Ours	
	A	T	A	T	A	T
apte	46.92	7	46.92	1	48.2	3.075
xerox	19.83	25	19.8	14	20.27	0.919
hp	8.95	55	8.947	6	9.52	0.966
ami33	1.27	3417	1.205	20	1.20	4.634
ami49	36.80	4752	36.5	31	36.34	5.701

由表 2 及表 3 的時間欄可以看出，我們所用的方法在速度上具有極大的潛力，而在模組數量越多

時，我們的方法就具有更大的優勢。另外，就面積而言，得出的最佳面積也非常接近現有非切割法的結果。

表 3. GSRC 測試檔的性能比較

	MBS-OEA[10]		ACG[12]		Ours	
	A	T	A	T	A	T
N10	NA	NA	NA	NA	.236	1
N30	NA	NA	NA	NA	.219	3
N50	.204	123	NA	NA	.207	7
N100	.189	1706	.187	28	.186	30
N200	.192	12114	.187	129	.187	96
N300	.302	80984	.293	322	.300	171

結論

在此篇論文中，我們提出局部最佳合併的演算法來搭配放鬆波蘭式。此種新方法將其解空間縮小以快速獲取更佳結果，而且對每個放鬆波蘭式都能以有效率的方法算出面積。以模擬退火程式對 MCNC 及 GSRC 測試檔來進行實驗，証實模組數越多，我們的方法就具有更大的優勢，而且得出的最佳面積也非常接近現有非切割法的結果。

誌謝

本文為國科會計畫 NSC 99-2221-E-027 -006 部分研究成果。

參考文獻

- [1] D. F. Wong, and C. L. Liu, "A New Algorithm for Floorplan Design," IEEE Proc. DAC, pp.101 - 107, 1986.
- [2] Nakaya S., Koide T. and Wakabayashi S., "An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair," Proc. IEEE ISCAS, pp. 65 -68, 2000.
- [3] Chang-Tzu Lin, De-Sheng Chen, Yi-Wen Wang., "GPE: A New Representation for VLSI Floorplan Problem," IEEE Proc. ICCD, pp. 531 -533, 2002.
- [4] Y. Pang, C.K. Cheng, and T. Yoshimura, "An enhanced perturbing algorithm for floorplan design using the O-tree representation," IEEE Proc. ISPD, pp. 168-173, 2000.
- [5] Yun-Chih Chang; Yao-Wen Chang; Guang-Ming Wu; Shu-Wei Wu, "B*-trees: A New Representation for Non-slicing Floorplans," IEEE Proc. DAC, pp. 458 - 463, 2000.
- [6] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu, "Corner block list: an effective and efficient topological representation of non-slicing floorplan," IEEE Proc. ICCAD,

- pp. 8-12, 2000.
- [7] Jai-Ming Lin and Yao-Wen Chang, "TCG: A Transitive Closure Graph-Based Representation for Non-Slicing Floorplans," IEEE Proc. DAC, pp.764-769, 2001.
 - [8] X. Tang and D. F. Wong, "FAST-SP: A Fast Algorithm for Block Placement based on Sequence Pair," Proc. ASP-DAC, pp. 521-526, 2001.
 - [9] J. Hu, Y. Shin, N. Dhanwada, and R. Marculescu, "Architecting voltage islands in core-based system-on-a-chip designs," in Int'l Symposium on Low Power Electronics and Design, pp. 180-185, 2004.
 - [10] J. Liu et al., "Moving Block Sequence and Organizational Evolutionary Algorithm for General Floorplanning With Arbitrarily Shaped Rectilinear Blocks," IEEE Trans. On Evolutionary Computation, Vol. 12, No. 5, 2008.
 - [11] Gigascale systems research center 2005, <http://www.cse.ucsc.edu/research/surf/GSRC/progress.html>
 - [12] H. Zhou et al., "ACG-adjacent constraint graph for general floorplans," Proc. ICCD, pp. 572-575, 2004.
 - [13] H.Y. Wang et. al., "Multiagent evolutionary algorithm for floorplanning using moving block sequence," Proc. CEC, pp. 4372-4377, 2007.