

行政院國家科學委員會專題研究計畫 成果報告

系統晶片後端設計變更之繞線研究

計畫類別：個別型計畫

計畫編號：NSC92-2218-E-009-013-

執行期間：92年08月01日至93年07月31日

執行單位：國立交通大學資訊科學學系

計畫主持人：李毅郎

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 93 年 11 月 29 日

行政院國家科學委員會專題研究計畫期末報告

系統晶片後端設計變更之繞線研究

An Efficient ECO Router for SoC Designs

計畫編號：NSC 92-2218-E-009-013

執行期限：92/08/01 ~ 93/07/31

主持人：李毅郎 交通大學資科系

E-mail:ylli@cis.nctu.edu.tw

一、中文摘要

Engineering Change Order (ECO) 主要是利用加入或修改元件與繞線來解決電路功能異常、時序或是雜訊的問題。由於 ECO 繞線問題具有已存在龐大數目的障礙物以及不同的設計規則的特性，這些都增加問題的困難度。與網格式繞線器相較，非網格式繞線器更能夠克服前述問題的瓶頸。本計劃的目地在建置一個非網格式的點對點繞線器，能夠快速執行數百萬閘級設計的 ECO 繞線。

英文摘要

ECO (Engineering Change Order) routing is frequently requested in the later design stage for the purpose of delay and noise optimization. ECO routing is very difficult as a result of huge existing obstacles and the requests for various design rules. Gridless router is more applicable to overcome the problem barrier than gridded router. We plan to construct in this project a fast gridless point-to-point router for the ECO routing of multi-million gate designs.

二、計畫的緣由與目的

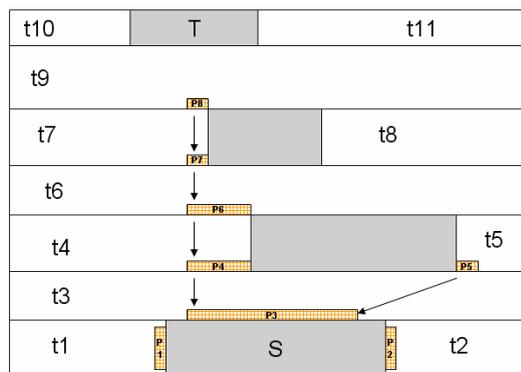
由於製程與電路設計技術顯著的進展，設計複雜度達到數百萬閘級，對佈局最佳化帶來了嚴酷的挑戰。在設計流程的後段，由於生產製造的良率、效能、可靠度等原因，經常造成原先的設計條件限制改變，而須要執行後端變更設計(ECO)繞線。此時實體佈局已接近完成，存在龐大數目的障礙物，原先的繞線資訊與環境有可能已遺失或不適用，加上不同的設計規則使得 ECO 繞線變得非常困難。且為了延遲與雜訊的最佳化或製造因素等，需要可變動的導線尺寸與導線間間距，使用傳統規律網格式繞線器，會需要建立非常密集的網格，這會使得繞線時間變的很慢且需要龐大記憶體。為了克服前述問題的瓶頸，之前已有很多關於非網格式繞線器的研究 [1][2][3][4][5][6]。本計劃的目地在建置一個非網格式的點對點繞線器，並且改善繞線速度，使其能夠快速執行數百萬閘級設計的 ECO 繞線。

三、研究方法

非網格式繞線器主要可分為連結圖模式(connection graph)[1][2][3]和方塊模式(tile-based)[4][5][6]。連結圖模式與方塊模式基本上都可

視為由許多節點與節點之間的連線所組成的圖形。繞線器便是從圖形中找出最好的路徑。通常圖形節點越少，問題複雜度越小也就能越快找到繞線路徑。而使用方塊模式所產生的圖形節點較少[6]，因此我們選用方塊模式來實做非網格式繞線器。雖然方塊模式的圖形節點比連結圖模式少很多，但在做數百萬層級設計的 ECO 繞線時，搜尋速度仍然不夠快，因此我們打算進一步減少方塊模式的圖形節點個數。

1. 方塊式繞線器



圖一

方塊式繞線器是用邊角縫織 (Corner Stitching) 資料結構形式來儲存實體布局[7]；佈局的實體幾何形狀和空間的區域會被切成一塊一塊的方塊，如圖一：灰色的方塊是佈局的實體幾何形狀，白色的方塊是空間的區域。使用邊角縫織的好處是可以在常數時間內找到與某個方塊相鄰的方塊，方便將來作方塊傳播 (tile propagation) 動作，另一個好處是可快速搜尋指定範圍的方塊，在多層 (multi-layer) 繞線時可以有效的對相鄰繞線層作查詢動作，找出 Via 區域以決定是否能夠轉換繞線到相鄰的繞

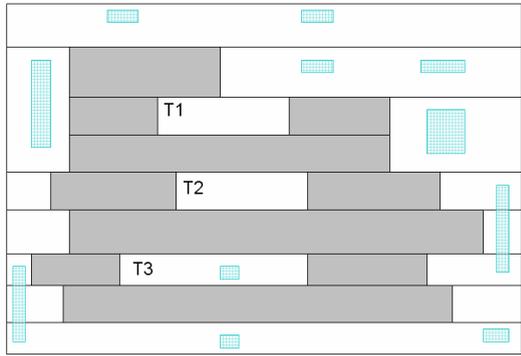
線層。以下簡介方塊傳播的過程，如圖一：方塊 S 和方塊 T 是我們要做繞線的出發點和目的地。首先在方塊 S 的相鄰方塊 t1, t2, t3 分別產生路徑節點資料 P1, P2, P3，記錄相關位置及成本函式，成本函式是由來源成本和目標成本組成。來源成本為出發點到目前路徑節點所走的距離，可準確算出；目標成本由路徑節點到目的地的曼哈頓距離 (Manhattan distance) 來評估。一開始 P1, P2, P3 的來源成本定為 0，放入堆疊 (Heap) 中依總成本排序。接下來由總成本最小的 P3 來做路徑節點的傳播 - 也就是方塊傳播。在 t3 的相鄰方塊 t4, t5 產生新路徑節點 P4, P5，分別計算其成本函式並使用倒退指標連到 P3。不斷的傳播路徑節點，直到遇到目標方塊。此時經由倒退指標可找到一組方塊串列：t3-t4-t6-t7-t9，再交給路徑建構函式，建出一條最短距離及最少成本的路徑。

2. 減少方塊模式的圖形節點個數

(1). 移除多餘無用的方塊

在方塊傳播的過程中，有些空間方塊傳播進去後是死路一條，無法再傳播出去。或者有些空間方塊是封閉的，根本不可能被傳播到。這些空間方塊可以說是多餘無用的方塊。以下介紹判別這些方塊的方式：

- (a) 如圖二中的方塊 T1，只有一個相鄰的空間方塊，且沒有 Via 區域。
- (b) 如圖二中的方塊 T2，沒有相鄰的空間方塊，也沒有 Via 區域。
- (c) 如圖二中的方塊 T3，沒有相鄰的空間方塊，有一個 Via 區域，但是這



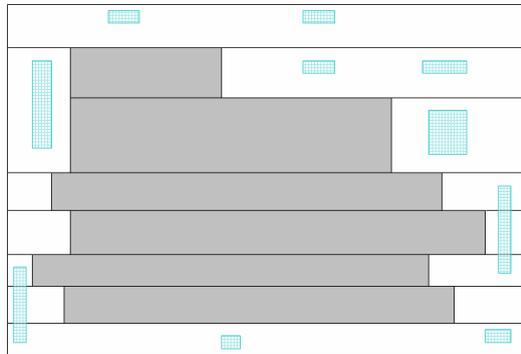
Block Tile
 Space Tile
 Via Region

No. of Block Tiles = 10 No. of Space Tiles = 16

圖二

個 Via 區域沒有足夠大小放置兩個 Via。

移除方塊 T1、T2、T3 後，如圖三，由於邊角縫織的特性，空間方塊減少三個。障礙方塊減少四個。

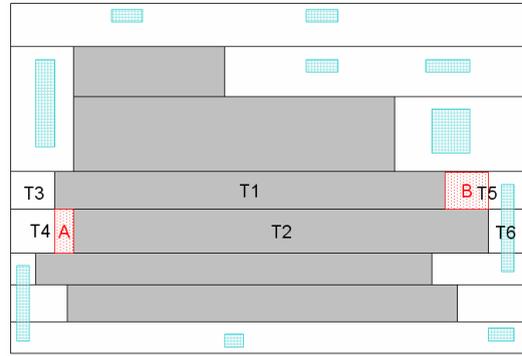


No. of Block Tiles = 6 No. of Space Tiles = 13

圖三

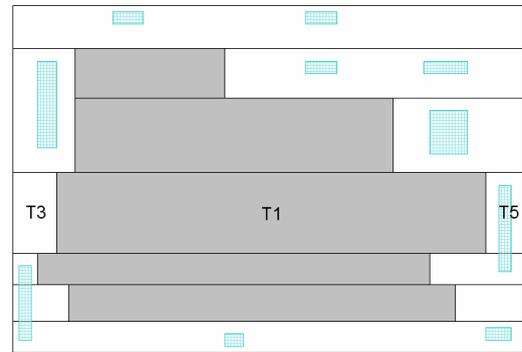
(2). 方塊對齊與結合 - 消除碎塊

依水平方向建立邊角縫織資料結構，雖然水平方向可避免產生小碎塊，但垂直方向則否。同理，依垂直方向建立邊角縫織資料結構，也會有如此現象。我們採用方塊對齊與結合的方式，來減少碎塊的數目。如圖四，紅色區域A和B是方塊T1、T2和其邊界範圍 (Bounding Box) 的差異區域。



圖四

在差異區域A中並無其他障礙方塊及 Via區域的存在，所以方塊T2的左邊界可往左移動到和方塊T1對齊。此時方塊T3和T4可結合為一個方塊。同理，方塊T1的右邊界可往右移動到和方塊T2對齊。此時方塊T5和T6可結合為一個方塊。接者，方塊T1和T2可結合為一個方塊。如圖五。



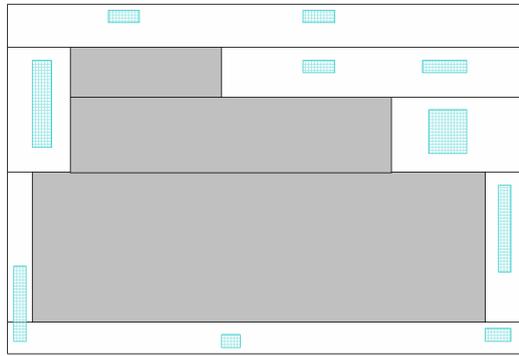
圖五

圖六是做完移除多餘無用方塊與方塊對齊結合後的結果，和原來的圖二相比，空間方塊減少九個。障礙方塊減少七個。

邊角縫織資料結構可快速搜尋所有方塊[7]，我們可在搜尋過程中移除多餘無用方塊及進行方塊對齊結合。當然，在繞線前做這些前置處理需要多花一些額外時間。

四、實驗結果

測試範例是從一個實際IC設計的



No. of Block Tiles = 3 No. of Space Tiles = 7

圖六

Layer		No. of Block Tiles	No. of Space Tiles
Met2	O	1067179	973528
	D	648669	522096
Met3	O	591120	541706
	D	348948	262829

O: without de-fragment, D: de-fragment.

表一

Met2、Met3繞線層得來。表一為建立邊角縫織資料結構後方塊的數目。從表一中可知道移除多餘無用方塊及進行方塊對齊結合後，方塊的數目約減少了一半。隨意選取大範圍的兩點進行點對點繞線，表二為其中五次測試的結果。從表二中我們可知道繞線長度和Via數目並無太大改變，證明減少方塊數目不會影響繞線品質。而其繞線時間平均約減少35%，但是加上額外的處理時間，總時間平均減少25%，不過有的測試只有減少5%的總時間，這是因為其繞線範圍較小，所須繞線時間較短，相對而言de-fragment的處理時間所佔比例也就較大。目前de-fragment是針對所有區域來作，如果只針對繞線範圍的bounding box約略來執行de-fragment，應能維持平均加速全部處理時間25%的水準。

五、結論與討論

我們完成了非網格式的方塊式點

對點繞線器，並以減少方塊數目來達到繞線速度的提升，符合預期的成果。將來希望能減少前置處理時間以及邊角縫織結構的建立時間來進一步提升繞線速度。參與計劃的人員獲得分析與實作的經驗，進而提升繞線器的速度。此方法可供EDA vendor改進其方塊式繞線器的速度。

六、參考文獻

- [1] Y. Wu, P. Widmayer, M. Schlag, and C. Wong, "Rectilinear shortest paths and minimum spanning trees in the presence of rectilinear obstacles," *IEEE Trans. Computers*, vol. C-36, no. 1, pp. 321-331, 1987.
- [2] S. Zheng, J.S. Lim, and S. Iyengar, "Finding obstacle -avoiding shortest paths using implicit connection graphs," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 1, pp. 103-110, Jan. 1996.
- [3] J. Cong, J. Fang, and K. Khoo, "An implicit connection graph maze routing algorithm for ECO routing," *in Proc. Int. Conf. Computer-Aided Design*, pp. 163-167, Nov. 1999.
- [4] C. Tsai, S. Chen, and W. Feng, "An H-V Alternating Router," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 976-991, Aug. 1992.
- [5] J. Dion and L. M. Monier, "Contour: A tile-based gridless router," Western Research Laboratory, Palo Alto, CA, Research Report 95/3.
- [6] Zhaoyun Xing, and Russell Kao, "S-hortest Path Search Using Tiles and Piecewise Linear Cost Propagation," *IEEE Trans.*

Computer-Aided Design, vol.21,
no.2, pp.145–158, Feb. 2002.

CAD-3, pp.87–100, Jan. 1984.

- [7] J. K. Ousterhout, “Corner Stitching:
A data-structuring technique for
VLSI layout tools,” *IEEE Trans.
Computer-Aided Design*, vol.

Test name		T1	T2	T3	Length	Via	R1	R2
Test1	O	0	319.438	344.438	11133.50	500	0.302	0.211
	D	24.101	222.907	272.008	11154.30	500		
Test2	O	0	165.859	190.859	9810.42	390	0.396	0.219
	D	24.005	100.156	149.161	9821.42	390		
Test3	O	0	357.607	382.607	12516.45	502	0.341	0.256
	D	24.125	235.765	284.89	12531.91	502		
Test4	O	0	151.250	176.250	9660.09	396	0.379	0.188
	D	24.153	94.016	143.169	9678.05	396		
Test5	O	0	77.437	102.437	7718.18	330	0.38	0.051
	D	24.188	48.063	97.251	7727.66	330		

O: without de-fragment, D: de-fragment, T1: De-fragment time, T2: routing time, T3: total time.

R1: $(T2_O - T2_D)/T2_O$, R2: $(T3_O - T3_D)/T3_O$

Total time counts in the setup time for corner stitching planes, where 25 sec for both schemes.

表二