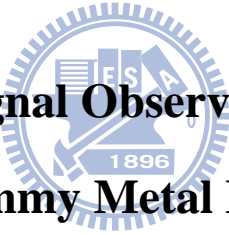


國立交通大學
電子工程學系電子研究所碩士班

碩士論文

利用虛擬金屬填充提昇晶片中訊號可觀察性


**Improving FIB Signal Observability by Utilizing
Dummy Metal Fills**

研究生：涂偉勝

指導教授：趙家佐博士

中華民國九十九年九月

利用虛擬金屬填充提昇晶片中訊號可觀察性

學生：涂偉勝

指導教授：趙家佐

國立交通大學電子工程學系電子研究所碩士班

摘 要

這篇論文提出將電路布局完成後留存的虛擬金屬填充與電路中訊號相連，進而使得可觀察訊號數增加，提昇電路中訊號的可觀察性。首先分析了聚焦離子束技術在觀察電路中訊號時所受到的制限，並且舉出了在未修改電路前聚焦離子束所能觀察到訊號數量在各層間的分布以說明可觀察度與訊號所在層數間的關聯性，以及高層虛擬金屬填充接入訊號與低層訊號在電路總可觀察訊號數量上的取代關係。接下來以同一組設定參數對於不同的測試電路進行修改，都得到小幅改進。最後綜合了之前所得到的數據及實作經驗，對未來可能的改進提出了建議。

Improving FIB Signal Observability by Utilizing Dummy Metal Fills

Student: Wei-Sen Tu

Advisor: Dr. Chia-Tso Chao

**Department of Electronics Engineering
Institute of Electronics
National Chiao Tung University**

Abstract

This thesis proposes a methodology to connecting dummy metal fills with signals in order to increase the observability of signals. At first we present the requirements and restrictions of signal observation by FIB (Focused Ion Beam) technology. The characteristic of the distribution of signals observable by FIB is also discussed after applied observability test on original circuits. By the breakdown charts there shows the relation between signal observability and the highest level the signal could reach. And the substitution effect on observability obviously exists between the new added dummy metal fills and the lower level signals. After all the analysis and discussion, we use a set of parameters to modify each test circuits. The result shows the observability increase marginally in each case. We suggest some possible architecture renovations and the limit of this topic at last.

誌 謝

首先感謝家中支持我讀碩士班。以及指導老師 趙家佐教授在各方面的提點。在校期間與實驗室同學間的討論也使我獲益良多，張琮偉提供讀入電入布局資料及搜尋功能的程式架構，郭淳仁提供關於 CGAL 以及 BOOST. POLYGON 等函式庫的資料，這份研究可以寫出來甚至有再往前推進的可能，都離不開大家適時的幫助。

其他在校期間與我有過互動的同學們也是我能走到這一步的助力：同研究室的穆思邦、陳弘昕、張智為、陳擴安、吳育澤、張啟銘、林政偉、楊皓宇、徐浩文、王易民、黃欽遠、張玟翔，以及蔡佳達、陳昭宏、廖偉翔，還有許多和我一起修課的同學們。

2009 年 9 月



目錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	vi
圖目錄	vii
第一章 介紹	- 1 -
第一章 第 1 節 錯誤診斷.....	- 1 -
第一章 第 2 節 虛擬金屬填充.....	- 1 -
第一章 第 3 節 我們的做法.....	- 2 -
第二章 背景	- 3 -
第二章 第 1 節 FIB 原理與應用	- 3 -
第二章 第 2 節 CGAL.....	- 4 -
第二章 第 3 節 BOOST.POLYGON 函式庫	- 5 -
第三章 問題構成及實作	- 6 -
第三章 第 1 節 簡化條件.....	- 6 -
第三章 第 2 節 實作.....	- 7 -
第三章 2-1 節 讀入 NET 資料	- 7 -
第三章 2-2 節 設定 DETERMINE	- 8 -
第三章 2-3 節 確定訊號的可觀察性	- 9 -
第三章 2-4 節 Dummy Metal Fill 的儲存與管理	- 10 -
第三章 2-5 節 搜尋可能相連線段	- 10 -
第三章 2-6 節 連線策略	- 11 -
第四章 數據	- 14 -

第五章 結論..... - 16 -

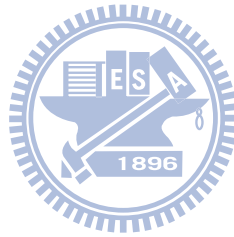
第六章 參考書目..... - 18 -



表目錄

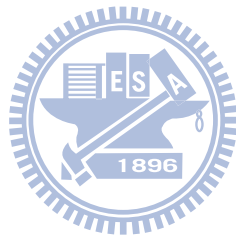
表格 1 分層可觀測訊號及總訊號數比較..... 11

表格 2 修改後訊號可觀察度比較..... 14



圖目錄

圖表 1 FIB 原理示意圖.....	3
---------------------	---



第一章 介紹

第一章 第1節 錯誤診斷

在晶片設計的過程中，為了對付越來越短的上市時間及不易預測的製程影響，有許多技術已經廣泛應用在晶片的功能測試及錯誤分析中。在發現晶片有問題之後，”診斷問題發生的原因”便成為許多研究探討的問題。但是因為晶片設計與製程技術的越加複雜，要確認錯誤並找出原因也變的更加昂貴且困難。製程的推進對於診斷問題發生的原因的影響在於，不容易在做出第一批晶片送回設計者手中之前就猜出可能硬體上可能出現的錯誤並加入模擬，要在做出了初步的產品之後，才能發現有地方出錯了。而且這些事前沒有考慮到的錯誤不止難以模擬、分析，甚至連重現錯誤都有困難[2]。就算許多電路都已經採用全掃描鏈設計，電路中大部分訊號還是無法觀察到的[5]。因此，為了觀察或修正電路中的訊號，有時候就得使用聚焦離子束（Focused Ion Beam，簡稱 FIB）改動晶片的電路結構，如何讓改動的範圍最小以提高改動的成功率也成了研究的題目。

對於如何系統性的找出晶片上錯誤的原因，許多人由不同的切入點提出了不同的解決方法。初期的作法是建立 fault dictionary [7]，後來也有人開始探討電路中同時多個錯誤發生時的分區錯誤診斷[1][11]。接著有人提出了 SLAT（single location at a time）測試輸入的概念[4]，使得有一連串的研究將 SLAT 測試輸入在錯誤診斷推廣到了拜占廷錯誤及橋接錯誤之類的複雜錯誤中[9][10][12]。

另一種研究方向是修改 ATPG（Automatic Test Pattern Generation）演算法，為電路增加診斷錯誤解析度的測試輸入[9][13]，這個方向主要探討不同錯誤模型下的測試輸入的產生，以及一次考慮的多個錯誤一起發生時是否還能正確的標定錯誤的發生點。有些研究則是在電路中加入 BIST（Built-in self test），希望能夠更快速的發現出錯的電路，甚至提供線索供設計者檢驗錯誤可能出現的位置[6]。這種作法主要應用在已經量產後產品的錯誤分析。

第一章 第2節 虛擬金屬填充

虛擬金屬填充（Dummy Metal Fill）技術在半導體製程走入 65nm 之後變的越加重要。造成這種現象的原因之一是由於化學機械研磨（Chemical-mechanical polishing，以下簡稱 CMP）及蝕刻工序的影響。在電路中加入 Dummy Metal Fill 的目的是維持電路中的金屬密度（metal density），而金屬密度對 CMP 及蝕刻的進行都有一定的影響。

CMP 的重要性在 130nm 製程開始發展之後，有了相對於過去非常多層的金

屬繞線層，因此各層間是否平整會嚴重影響到上面各層的平整度，同時也會因為許多製程中的曝光焦平面太薄造成顯影失敗的問題。為此，CMP 在晶片每作好一層金屬層之後就要使用一次。但在研磨的過程中，會因為作為介電質的二氧化矽和作為導線的金屬材料的硬度不同，使得打磨的結果和理想中有所差異。虛擬金屬填充（Dummy Metal Pattern）的出現就是要盡量使晶片上各部分在 CMP 過程中產生的差異降低，避免在 CMP 時發生某些區域的導線被過度研磨，或是介電質和導線一起被磨蝕的情形發生 [8]。

第一章 第3節 我們的做法

在這份研究中，我們著眼於利用在電路合成時已經加入的 Dummy Metal Fill 作為將原本無法觀察到的訊號，經由 Dummy Metal Fill 傳導到可用 FIB 技術觀察的區域。可觀察區域的限制其實還不少，因此在未經修改之前一個電路中可以使用 FIB 技術觀察到的線路並不算多。我們提出的方法約略的提昇了電路中可觀察的訊號數量，並且分析在修改電路的過程中所要面對的問題，以及未來可能再改進的部分。

在之後的章節中，第二章簡略描述 FIB 的原理以及應用範圍，第三章說明問題構成以及限制條件，並描述實作方法，第四章為實驗結果，第五章為結論以及可能改進方向。

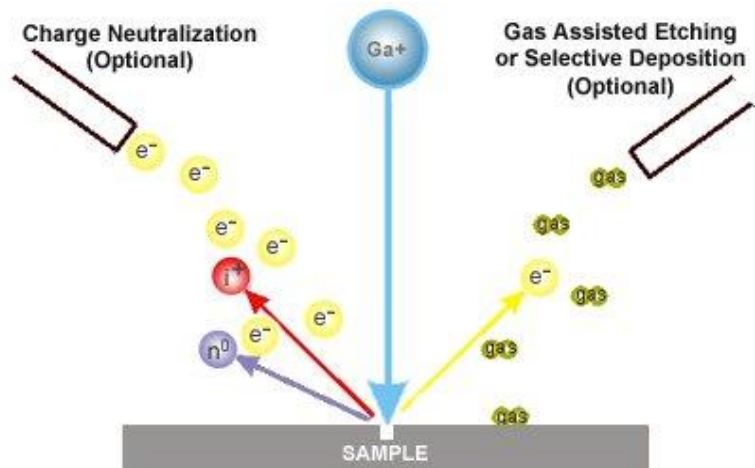


第二章 背景

第二章 第1節 FIB 原理與應用

FIB 是 Focused Ion Beam 的簡稱。中文一般譯成聚焦離子束，原理與掃描式電子顯微鏡 (SEM, scanning electron microscope) 相似，都是利用帶電粒子在電場中加速後打向試樣的技術。右圖為表示 FIB 原理的示意圖。

Ga^+ 離子束朝試樣射擊，把試樣表面的材質打成了 i^+ (帶正電的離子)、 i^- (帶負電的離子)、 n^0 (電中性的原子) 以及被離子束打到之後從試樣中彈出的 e^- (電子)。當離子束在試樣上移動時，便可一邊收集彈出的電子。利用收集到的電子數量的不



圖表 1 FIB 原理示意圖

同就可以成像。用低流量離子束觀察時，試樣表面只會有少量材質會被彈起。圖中左側的 Charge Neutralization 管可用來射出電子，電子與電正電的離子束可在彈開試樣表面的原子後中和，取代原本的表面原子堆積起來。圖中右側的輔助蝕刻/沉積氣體管可用來協助對試樣表面蝕刻或沉積。FIB 本身就可以靠離子束對試樣表面的撞擊達成蝕刻的效果，而輔助蝕刻氣體可用來加速蝕刻速度或是提高對於不同材質的蝕刻速度差異，但是 FIB 對於不同材質的蝕刻速度差異不如酸液蝕刻明顯，在使用時。FIB 的沉積 (或稱蒸鍍) 通常是為了要避免試樣表面 (或是蝕刻出來的洞) 受到外來物質入侵而填上絕緣物質或是金屬層。要填上絕緣層時多半是使用 Thermoplastic Elastomers 加上 Siloxane 及 O_2 以鍍上 SiO_2 ，要填入鎢會使用 $\text{W}(\text{CO})_6$ ，要填入鉑則會使用 $\text{C}_9\text{H}_{16}\text{Pt}$ 。

和 SEM、TEM (Transmission electron microscope)、EBID (electron beam-induced deposition) 不同，FIB 技術不使用電子束而是使用離子束，這個差別也就此顯示在對試樣表層的影響上。

離子的粒子比電子大：因為粒子大，所以不容易與試樣表面材質產生原子層次的影響。通常離子束打入之後，只會將表層原子離子化或是打斷試樣基材的化學鍵。就穿透深度上來看，離子束也較電子束的穿透深度低。另外，當離子束鑽

進試樣中之後，也會因體積太大而被固定在試樣中。

離子比電子重：由於重量較重，在同樣加速成能的條件下，離子束的動量約是電子束的 370 倍。雖然在這同時離子束的速度也較電子束慢，但在成像模式下幾乎沒什麼差別，所以這部分的缺點可忽略。同時由於質量較高，在 SEM 中用到的磁場透鏡對離子束幾乎沒有控制力，所以在 FIB 技術中使用靜電場透鏡取代磁場透鏡以集中離子束。

總而言之，離子束的特性是帶正電荷、體積大、質量大、移動緩慢。電子束的特性是帶負電荷、體積小、質量小。這些特性上的差異最直接的影響便是離子束可以從試樣表面上將材質移除，並且因為離子束的射擊位置、時間、大小可以精確控制，所以能夠以高精確度觀察、修改試樣[3]。

FIB 所使用的 Ga^+ 在射進試樣後可以植入試樣中，並把試樣表層非晶態化 (amorphous)，利用這種性質便可在微米尺度對試樣研磨或修改。此外還有在 FIB 過程中加入氣態金屬將金屬原子鍍至試樣表面的作法，這種作法可用在晶片修改電路的部分。

FIB 在半導體業中最常用在晶片修改，特別是將某些有問題的接線切斷、接上某幾個該要導通的訊號，以免去等待修改 layout 後重新經過 foundry 製作所需要的時間，並可以馬上送回驗證部門進行功能上的驗證。一般而言 FIB 在半導體業的應用範圍有：缺陷分析、電路修改、電路量測、光罩修復、TEM 試樣製備。在這次的研究之中所使用到的用途是電路量測。電路量測主要是想要在晶片的運作時期量測晶片中某幾個特定訊號的邏輯值，通常是先對目標位置成像，確定所要觀察的電晶體後，決定好要挖洞的位置，之後填入鉑，再拉線到晶片上的其他區域以供探針觀察邏輯值。以 FIB 觀察晶片中訊號線上的值並不是沒有代價的，除了想要觀察的訊號線上方不能有其他訊號線的阻擋之外，填入的金屬的電阻值相對而言也不小（超過 100 歐姆的等級）。

第二章 第2節 CGAL

這一節以及下一節都是用來簡略介紹之後有可能在改進處理方式時所需要用到的計算幾何上的函式庫。

CGAL (Computer Geometry Algorithm Library) 是設計在 Computer Geometry 中使用的函式庫，大部分以 template 寫成，實作範圍包含了 2D、3D 及多維向量的相關演算法。在我所處理的題目中只有用到 2D 部分，使用的函數主要是 2D binary operation 中所包含的內容。雖然在這次研究中朝這個方向前進的部分沒有完成，但是在之後若要有更進一步的改進，無可避免的要利用 CGAL 或是類似它這樣可以描述幾何結構的技術才行。

CGAL 的特點是對線與線、平面及平面間交錯關係的判斷十分精確。但是為了達到他們所要求的精確度的代價是，函式庫中表示座標或其他一維變量時不使用 int、double 之類的資料型態，而是使用 GSL 中精確度更高的變數型態。選擇

使用這種資料型態最直接的結果就是存一個整數的空間需求大約等同於普通的 C 程式儲存一個 double 的空間需求。在這份研究中要處理的問題的特點之一就是資料量不小，而且又都是相對單純的長方形。若是使用 CGAL 函式庫就必需要把各線段套上二維平面上多邊形的資料結構使用。單純就二維平面上的長方形二元集合運算而言速度不算快（之前測試時，兩個有重疊的四邊形做一百萬次聯集，大約要花上四分鐘）。但如果能夠小心的將要處理的畫面分割的話，速度上應該會變的比較能接受。

第二章 第3節 BOOST.POLYGON 函式

庫

BOOST.POLYGON 做的事情和 CGAL 很像，都是將計算幾何上常用的演算法在 C++ template 中實作的函式庫。比較明顯的差別在 BOOST.POLYGON 只處理平面上多邊形的二元運算，而且對座標點的描述限定使用整數。

相對於 CGAL 而言，這個函式庫一開始就是針對 VLSI CAD 上的用途設計出來的，或許也就是因為這個原因才使得這個函式庫選擇 int 做為基底。此外這個函式庫的集合運算速度也比 CGAL 快上許多。要是之後要建立電路中可觀察區域列表的時候可以先考慮用這個函式庫產生。



第三章 問題構成及實作

第三章 第1節 簡化條件

在這個研究中，為了簡化問題，所以設定了一些條件：

給定一個已經完成 P&R 的電路(在這個實驗中，總共用了 iscas89 benchmark 的 s38584 以及 itc99 benchmark 的 b22 與 b17)，加入 dummy metal 之後，系統性的加入新的 via 以增加可以用 FIB 觀察到的訊號數量。

觀察到的繞線數量的計算是從最高層的繞線層開始往下算的。在最高繞線層以上的訊號都假設成當 FIB 由晶片最頂層往下鑽孔時可以忽略的部分，而且這個假設在非繞線層中應該也算是合理的，畢竟在繞線層之上的金屬層中所傳導的訊號多半使用網狀結構 (Mesh) 以利能量的傳遞，如果只是切開一個點照理而言不應該會對電路的功能有破壞性的影響。

另一個假設是不考慮同一時間觀察兩個訊號可能造成的問題。這個假設只是用來單純化問題，避免考慮挖下一個洞的同時破壞了別條線接到可觀察區的可能性。

還有一個影響了 layout 的假設：在加入 Dummy Metal Fill 的時候加入了 constraint，使該層的 Dummy Metal Fill 寬度的最大值和最小值等同於該層的訊號線寬度。這是為了要讓 Dummy Metal Fill 在補上 via 之後就更有可能滿足訊號線的 DRC 條件。此外在轉寫入 DEF 檔中時也容易許多；SOC Encounter 在 APR 後產生的 DEF 檔中的訊號線只記錄該段訊號線兩端的中心點位置，寬度則是在 LEF 檔中就已經設定好的，在同一層裡每條訊號線都可以當作是一樣粗，要是原本有各種不同寬度的 Dummy Metal Fill 從底下一層的訊號接上了 via，就算不管能不能通過 DRC 檢查，DEF 格式是否支援描述不符合該層繞線寬度的訊號也不確定。因此，為了簡化問題，在產生測試用電路 layout 的時候就只加入固定寬度的 Dummy Metal Fill，以利之後的資料處理。

此外，為了方便判斷，Dummy Metal Fill 設定成不與 VDD 或 GND 相連。這樣一來才有機會把訊號接到 Dummy Metal Fill 上以供之後觀察。

最後，我們把無法觀察到的訊號尋找可能接上的 Dummy Metal Fill 線段的搜尋範圍限制在一層。理論上一開始無法用 FIB 技術觀察到的訊號，如果向外接上了某個 Dummy Metal Fill 線段，就有很大的機會可以再利用剛接上的 Dummy Metal Fill 線段再向更上一層的 Dummy Metal Fill 線段相連。但相對之下不止搜尋時間變長，而且在電路特性的層面上，多出來的 via 以及多接上的訊號線長度都會使得阻抗變的更強，進而產生 timing violation。由於這份研究中沒有把修改過的電路重新交給 PrimeTime 之類的軟體分析，所以決定使用保守一點的方法，限定搜尋範圍為一層，希望能夠降低對於電路時間特性的影響。

第三章 第2節 實作

在 SOC encounter 作完 P&R 之後便可存成 Def 檔。Def 檔中包含了繞線及各元件擺放位置的資訊。這些資訊雖然勉強可以用看的看懂，但是在處理上還是不太容易，所以在一開始會先用之前由張琮偉寫的 defrw 將 1 個 def 檔分成各 layer 檔及一個 via 檔。

各 layer 檔所記載的是在那一層所出現的訊號，以及構成該訊號的線段 (segment)，而 Via 檔則是記錄了所有有用到 via 的訊號，以及每個訊號所使用的 via 列表，包含了 via 所在的 layer 以及 via 出現的位置，還有 via 的種類。此外還需要用一個 script 把 Def 檔中記錄的 Dummy Metal Fill 的位置抽出，Dummy Metal Fill 沒有名字，只有出現的層數，以及開始點與結束點。

第三章 2-1節 讀入 NET 資料

程式在讀取各層檔案時，會把資料存到 net 中。net 是個由 string 查 NET 資料結構的 map。NET 的資料成員有：該訊號的名稱 (以 string 儲存)、記錄線段資訊的 line (資料結構為 map<int,LINE>) 和記錄 via 資訊的 via (資料結構為 map<int,VIA>)。

line 和 via 這兩個 map 的 key 為什麼要用 int 而不是其他複合型的資料結構？照理說要搜尋 1 個訊號在某個 layer 的某一點開的線段應該要使用 2 個 int (標記在哪個 layer 以及在所在的 x 軸或 y 軸座標) 或 3 個 int (標記所在的 layer 以及其中一點的 x 座標及 y 座標) 的資料結構作為鍵值才合理。

但是在 LINE 的資料結構中取巧的利用了 layer 的數量不會太多的性質 (我們處理的情形中只有 metal 1 到 metal 9，甚至有的設計中會出現只有到 metal 8 的情形) 以及各訊號在各層都不會有太多線段的條件 (在每層都有非常多線段的 clock 訊號在讀取輸入的時候會被特意忽略)，把 key 定成所在的 layer*OFFSET+線段被加入該訊號的順序。OFFSET 原本設為 100000，不計 Dummy Metal Fill 的話應該是夠用的。目前在程式中則是改設定為 2000000，就一定沒有訊號有那麼多線段了，即使是把 Dummy Metal Fill 一起計入也不至於超過。

NET 還有 2 個值得說明的成員變數：lineMap 及 viaMap。它們的資料型態都是 map<TRI,int>。TRI 是個比較單純的資料結構，內容為 3 個 int。viaMap 用 3 個 int 作為鍵值可以理解，就是把 layer、via 所在的點的 x 座標、via 所在的點的 y 座標記下，就可以查到想知道的 via 在該 NET 物件內的 via index。而 lineMap 則是比 viaMap 麻煩一些，畢竟是多了一個維度的概念，所以在 lineMap 中每個線段都會用兩個 TRI 記起來，第 1 個 TRI 中記錄 layer、開始點的 x 座標、開始點的 y 座標，第 2 個 TRI 中記錄 layer、結束點的 x 座標、結束點的 y 座標。

第三章 2-2節 設定 DETERMINE

就上面提到的架構看來，應該是足以把電路中每個訊號的線段以及 via 加入資料結構中了，但是在為了要能夠更方便的判斷某個訊號現在是否可以被觀察到，所以在加入線段或 via 時還會用到 determine。determine 的資料型態是 DETERMINE，成員變數只有一個 layerMap。layerMap 的資料型態為 LAYERMAP 陣列，長度就是總共的 layer 數。和上面幾段提到建立電路資料比較有關的部分是它的 add_line(string, Axis, Axis, int) 函數，這個函數會把線段的訊號名稱、線段起點、線段終點及所在 layer 傳入對應的 layerMap 中。DETERMINE::add_layer(string, Axis, Axis, int)這個函數在 NET 物件每次加入 1 個新線段時都會被呼叫，這使得 determine 可以追蹤讀入的 Def 檔中的所有訊號。

每個訊號的 via 及線段資訊會記錄在 NET 物件中，而各個 layer 中出現的線段及 via 由 LAYERMAP 物件負責保管。LAYERMAP 主要的成員變數有 3 個：hMap、vMap、viaMap，資料型態都是 multimap< int, SEG>。hMap 記錄該 layer 中的垂直線段，vMap 記錄該 layer 中的水平線段，viaMap 記錄 via 出現的位置。先解釋 SEG 的內容：SEG 共有 4 個成員變數，起始點 sp、結束點 ep 及訊號名稱 name；SEG 完全可以當作是給 LAYERMAP 專用的 LINE。了解了 SEG 的功能後，LAYERMAP 存入線段資料的方法就比較能理解了：如果線段是水平的，就以 x 座標為鍵值，把線段以 SEG 型別存入 hMap；如果線段是垂直的，就以 y 座標為鍵值把線段以 SEG 型別存入 vMap。

LAYERMAP 存入 via 的情形比較詭異，會由所在的 LAYERMAP 的 mainDirection 變數是 HORIZONTAL 或 VERTICAL 決定要以 y 座標為鍵值或是以 x 座標為鍵值。而每個 LAYERMAP 的 mainDirection 是在 DETERMINE 初始化時，視該 LAYERMAP 物件是奇數層還是偶數層決定的：metal (2n+1) 都設為 HORIZONTAL，metal (2n) 都設為 VERTICAL。不過只要前後一致的話應該不會有功能上的影響，至多就是在加 via 以及刪 via 的時候多判斷一次。

LAYERMAP 在這份研究中還有個比較重要的功能是配合 DETERMINE 判定各個訊號是否算是可觀察到的。這件事也是在 DETERMINE 初始化時，呼叫 LAYERMAP::set_visibleSpace(int) 完成的。目前的設定是從 routing layer 的最高層開始，越往下 visible space 越大；visible space 的命名其實不太明確，因為比較接近程式中運作方式的解釋其實是”要在這一個 layer 看到一個訊號所需要的觀察框的大小”。關於觀察框大小的影響，會在之後提到計算可觀察訊號的部分時再詳細說明。

在之後處理 Dummy Metal Fill 時，會再度利用 DETERMINE 與 LAYERMAP 資料結構中的定位、搜尋功能，使程式能更方便的搜尋到有可能可以用來接出訊號的 Dummy Metal Fill。

第三章 2-3節 確定訊號的可觀察性

再來要說明判定有多少訊號是可被觀察的函數：`void recalculate_obrate()`。因為是重新計算，所以會先把各個訊號的 `flag` 都清掉，其中包含了各 NET 物件的 `bool visible`，也會一起清成 `false`。接著就是由最高繞線層開始一個 layer 一個 layer 往下看。在各個 layer 的迴圈中會再把 net 中各個元素檢查一遍，遇到了還不是 `visible` 的 net，再進入檢查這個 net 各個線段（型別為 `LINE`）的下一層迴圈。在這層迴圈中，會先找出在目前的 layer 中的 `LINE` 物件（在這裡是用之前設定 `index` 時所加入的巨大 `offset` 配合 `map` 的找上界下界功能把該 layer 中正在處理的訊號所屬線段的範圍標出來），再每個線段試著是否能夠在該 layer 的條件下變的 `visible`。

判斷一個線段是否 `visible` 的部分分成幾層：首先要呼叫用來協助判定該 layer 向上是否有重疊的線給擋住的 `DETERMINE::check_up(int, string, Axis, Axis)`，把要檢查的線段的所在 layer、名稱、起點及終點傳入。在 `check_up` 函數中，先判斷線段的水平或垂直：如果垂直就把要檢查的空間在垂直方向上下推一個延伸段（延伸段的大小是由線段所在 layer 的深度決定，越底層的 layer 所要延伸的距離越長），再傳入 `LAYERMAP::cover_space_v(string, Axis, Axis, int, int, LRNODE&)`，要給的參數分別是要檢查可觀察性的線段的訊號名稱、線段起點（含之前的延伸段）、線段終點、半線寬、延伸段長度，以及用來記錄可觀察線段的 `LRNODE`。

在 `LAYERMAP::cover_space_v()` 中，會先檢查在該 layer 中的 `vMap` 中的線段，搜尋範圍是用線段所在的 `x` 座標向左右各推 1 個該 layer 標定線寬決定的。在符合搜尋範圍中的線段都有可能遮住想要觀察的線段，在排除了和要觀察線段同名的線段之後，把會遮住的部分以 `LRNODE::push_interval(int, int)` 記入 `LRNODE`。解決了 `vMap` 之後改搜尋以 `y` 軸範圍上下各推一個延伸段的 `hMap`，也是把遮住了觀察線段的範圍記入 `LRNODE`。最後再把同樣的檢查對 `viaMap` 跑一次。

上一段說到，在 `LAYERMAP::cover_space_v()` 中，凡是被遮住的範圍都是記入 `LRNODE`。`LRNODE` 的目的是維持一張表示數線上間距（`interval`）的表。每個 `interval` 都是由 2 個數字組成，小的數字在左，大的數字在右。每次呼叫 `LRNODE::push_interval(int, int)` 的時候，`LRNODE` 就開始檢查自己已記錄的 `interval` 表中是要新增一段 `interval`，或是要修改一個 `interval`，以及同時是否要刪去幾個被剛新增的 `interval` 所包含的舊 `interval`。

以上大約說明了 `LAYERMAP::cover_space_v()` 的運作，同理可以套用在 `LAYERMAP::cover_space_h()` 上，只是把許多地方的水平、垂直交換而已。值得一提的是在 `LAYERMAP::cover_space_v()` 及 `LAYERMAP::cover_space_h()` 執行的過程中，只有正好遇到把整個線段給遮住的上層線段才會直接回傳 `false` 停止往後的搜尋，就算是在 `LRNODE` 內部已經把整段空間都標示成被遮蔽也一

樣。話說回來，不管那兩兩個 LAYERMAP 內部的函數有沒有發現正好有條線段完全把要觀察的範圍完全遮住了，DETERMINE::check_up()最後都會把已經將所有被遮蓋間距加總起來的 LRNODE 傳回給要觀察的線段。

正在檢查是否能夠被觀察到的線段得到了回傳的 LRNODE 後，便執行 LINE::Visible()檢查這個訊號是否真的能從觀察不到變成可觀察到。基本上就是檢查是否存在有兩個 interval 之間有比規定距離更大的間隔，這個規定距離定義的和一開始提到 LAYERMAP 時講到的 visible space，都是越往下層所需要的空間越大。而且下層所需空間大的同時，每條線的規定寬度同時也縮小了。這使得在 metal 2、metal 1 的訊號幾乎無法觀察到。

第三章 2-4節 Dummy Metal Fill 的儲存與管理

為了保持原本功能運作正常，我們盡量使 Dummy Metal Fill 也能嵌入原有的資料結構。即便我們已經把加入 Dummy Metal Fill 的區塊縮小到 layout 核心地區的 9 倍大小而不是原本該選擇的整片 layout，SOC Encounter 產生出來的 Dummy Metal Fill 數量還是十分多，再加上設定 dummy metal 擺放條件時要求了寬度要固定為該層規定訊號線寬，所以最後每個電路中的 Dummy Metal Fill 總數大約都在百萬上下。加入的 Dummy Metal Fill 最多的電路是 b17，各層加起來超過了 150 萬個，每一層大約是 6 萬到 7 萬個左右。

一來 Dummy Metal Fill 不應該和普通訊號放在一起，因為在之後的程式運行中有可能會干擾計算 FIB 可觀察度的機制；二來 Dummy Metal Fill 的資料量十分巨大，就算不做其他事情，光是把這上百萬個 SEG 物件加入原本的各 LAYERMAP 結構中就得花一些時間，要是特意只在計算 FIB 可觀察度的函數中將 Dummy Metal Fill 排除，以原來的程式架構執行起來也大約會增加半分鐘的執行時間。但是 Dummy Metal Fill 的資料總是要載入的，為了避免干擾之前的運作，並且利用已經有的功能，我們在原本以 NET 記錄訊號，DETERMINE 管理各層線段資訊並計算各線段間遮蔽效果的框架中，另外加入了一個專門記錄 Dummy Metal Fill 的 NET 物件，另外再建了一個獨立的 DETERMINE 以記錄各層的 Dummy Metal Fill 線段。

第三章 2-5節 搜尋可能相連線段

在確定訊號的可觀察性一節中，有提到在判定一個訊號是否能夠以 FIB 技術觀察到時，會將這個訊號分層分線段的設定觀察框，再將這個觀察框向上移，每移一層就把會被該層中訊號遮蔽的部分刪去，要是遇到正好把整個觀察框都遮住的線段便立即回應：觀察不到；要是沒有，就要等到最高繞線層中對觀察框有影響的線段都標記完後才開始看還有多少空間可以觀察。這個原理在稍加修改之後，也能應用在對一個線段尋找有可能接出的 Dummy Metal Fill 上。

在經過一次 recalculate_obrate()後，每個訊號都會分別被標上可觀察到或無法觀察到。對於各個無法觀察到的訊號，分別由其所在的最高 layer（要是最高 layer 高過 metal 5，就直接設成 metal 5），逐個線段將有可能相連區域的參數：目前所在的 layer 向上推一層，目前線段起點，目前線段終點。這三個參數便可作為尋找有機會相連的 Dummy Metal Fill 的搜尋框。將這個搜尋框交給管理 Dummy Metal Fill 的 DETERMINE 物件查詢範圍內是否有機會接入，DETERMINE 會回傳第一個有機會接上的 Dummy Metal Fill 線段，或者回應找不到。

第三章 2-6節 連線策略

能夠知道有哪些更高一層的 Dummy Metal Fill 線段可以連入之前無法被觀察到的訊號後，選擇要連上哪些 Dummy Metal Fill，要連上多少 Dummy Metal Fill，Dummy Metal Fill 要給哪個之前無法被觀察的訊號連接…就需要進一步的討論了，而且每一步的選擇都會影響之後的訊號是否能接上 Dummy Metal Fill 的機會。

訊號選擇的先後

為了這個問題，我們曾經以各個訊號曾經出現的最高 layer 分群，再以該訊號是否能用 FIB 觀察到分組，作成一張表：



表格 1 分層可觀測訊號及總訊號數比較

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6+
s38417	2	91	1215	1106	646	44
	361	1558	3980	308	30	0
	363	1649	5195	1414	676	44
b17	0	50	446	1299	2201	198
	137	2921	7158	1470	348	0
	137	2971	7604	2769	2549	198
b22	0	39	494	757	712	335
	95	1612	4562	588	50	0
	95	1651	5056	1345	762	335

表中每個電路右邊都有三列，上列的數字代表最高 layer 在此的可觀察訊號數量，中列的數字代表最高 layer 在此的無法觀察到的訊號數量，下列的數字代表最高 layer 在此的總訊號數。最右邊的一欄則是包含了從 layer 6 一直到最高 layer 在內的訊號。在表中可以明顯看出來的兩個極端資訊：要是有個訊號的最高 layer 在 layer 6 或更高層，那麼就算不能在 layer 6 就被觀察到，也保證能在下

面幾個 layer 被觀察到；若是有個訊號的最高 layer 只有到達 layer 1，那它就幾乎不可能被觀察到了。

這個結果是符合直覺的：每個訊號其實都是從最底下的 Poly 層一段段接上來的。若是有個訊號最高爬到了 layer 7，它在 layer 6、layer 5…也一定會留下屬於它的的線段。此外，越高層的訊號以 FIB 觀察時所需要的觀察框也越小，上層訊號線的數量相較於下層也不算多，基本上不容易被其他訊號給遮住，這兩點使得高層訊號相對於低層訊號擁有非常大的優勢。

因此，對於一個原本無法觀察到的訊號來說，要是可以把訊號向上推到 layer 6，被觀察到的可能性便可大幅提昇。但這並不是沒有代價的，一個訊號連到 layer 6 的 Dummy Metal Fill 有可能直接影響到另外四個 layer 4 的訊號中的線段，使它們由原本可以經由那些線段被觀察變成需要尋找其他的線段作為觀察點，甚至變成無法觀察到。關於這種“可觀察到訊號的替代效應”之後會再討論，這裡先談在這次實驗中所使用的訊號選擇標準。

在選擇訊號時，為了能讓成功接上 Dummy Metal Fill 的機會提昇，目前的作法是按訊號所在最高 layer 排序。由表中可以知道，凡是訊號所在最高 layer 在 layer 6 以上的幾乎都一定看的到，所以這就等於是按 layer 5、layer 4…這樣排序下來了。

一個訊號可接上多少 Dummy Metal Fill

每個有可能被接上的 Dummy Metal Fill 對處理中的訊號而言都是個機會。但是對其他待處理的訊號（甚至一些原本可觀察到的訊號）而言，卻有可能使那些訊號無法再被觀察到。在判斷訊號的可觀察到一節中，說明了我們目前的做法是減法：把要觀察的線段列出來，接下來往上一層一層找，凡是有不行的線段就裁去，最後再來看餘下的空間是否能夠塞的下觀察點大小。這個方法就算在修改之後也只能知道訊號在某個點上可以被觀察到，但卻不能確定某個訊號到底還有沒有其他點可以被觀察到，抑或僅僅只有那個點可以觀察了。

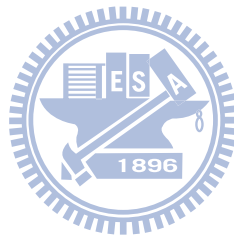
在進一步修改程式使我們可以判斷究竟各個訊號要再被觀察到的難度有多大之前，就只能設法避免一開始先選擇 Dummy Metal Fill 的訊號把太多 Dummy Metal Fill 接在自己的訊號上，還不論是否能夠真的在修改之後被觀察到，有可能就這樣把下層原本可被觀察到的訊號遮住了，或者搶去原本可以給同層的訊號接出的 Dummy Metal Fill。一開始曾經設定一個訊號一次最多可加入 50 個 Dummy Metal Fill，但是卻使得 layer 3 以下可觀察到的訊號大量減少，目前的設定則是一個訊號最多可以接上 5 個 Dummy Metal Fill。

演算法

總的說來，我們的做法是這麼進行的：

1. 計算 FIB 可觀察的訊號數
2. 將未觀察到的訊號依所達到的最高 layer 排序
3. 對每個未觀察到的訊號進行第 4 步，全部執行完則進行第 5 步
4. 由最高 layer 開始逐個線段尋找可連接的 Dummy Metal Fill，找到一個就連接一個。若找到 5 個可連接線段或全部找完跳回第 3 步執行下一個訊號
5. 重新計算 FIB 可觀察到的訊號數

目前的做法還是十分粗略的，像是從第 2 步到第 5 步就可以再多做幾輪。不過在之前嘗試時，發現在第二輪之後對於 layer 3 以下的訊號的可觀察性的破壞有時候還高過 layer 4、layer 5 中訊號可觀察性的提昇，所以這邊就沒有把對電路多次修改的作法寫出來，只留下接入一層 Dummy Metal Fill 的作法和結果了。



第四章 數據

在這一份研究中所用到的測試電路分別是 iscas85 的 s38417 以及 ITC99 的 b22 和 b17。這三個電路都是在 SOC Encounter 中 P&R 之後，以該繞線層所要求的訊號寬度加入 Dummy Metal Fill。

表格 2 修改後訊號可觀察度比較

	s38417			b17			b22		
Layer	#net	obsv	ob.f	#net	obsv	ob.f	#net	obsv	ob.f
6	44	44	65	187	187	193	334	330	337
5	676	601	620	2560	1972	1987	771	666	694
4	1414	1045	1214	2769	1463	1479	1348	742	733
3	5195	1228	1215	7604	459	461	5056	516	512
2	1649	158	148	2971	109	102	1651	72	66
1	363	28	19	137	4	2	95	11	3
total	9341	3104	3281	16228	4194	4224	9255	2337	2345

這張表格的 #net 欄表示在該層出現的新訊號數量，obsv 欄表示在修改電路前聚焦離子束在該層所能新觀察到的訊號數量，ob.f 欄則是以我們的演算法修改電路後所能觀察到的訊號數。

ob.f 欄的數字和 obsv 相同，都是取在該層新發現有辦法觀察到的訊號數量。因為有些原本在下層無法被觀察到的訊號在修改後最高出現層數向上推了一層，所以會出現有些地方 ob.f 的數字比 #net 的數字還大的情形。比較明顯的變化在 layer 6 到 layer 4 的部分，幾乎可觀察到訊號數量都是有增加的，但是在更下層的部分就可以看出可觀察訊號數量減少的情形。

這樣的結果是可以預期的，畢竟在目前的架構之下，每次加入的 Dummy Metal Fill 要是能發揮作用，最好的情形是增加一個可觀察訊號；但要是無法發揮作用，反而會讓加入 Dummy Metal Fill 的訊號在所屬 layer 增加有可能遮住其他訊號的線段，最好的情形是減少零個可觀察訊號。

在表上還有個令人注意的地方：在每個測試電路的 layer 3 通常都是最多新訊號出現的 layer，同時也是無法觀察的訊號的數量比可以被觀察到訊號的兩倍還高的一層。造成這種現象的原因有可能是：一來有太多訊號的最高 layer 都只有達到這一層，所以相對於上面幾個 layer 訊號的擁擠程度提高了；同時 FIB 在這一層所需的觀察框又持續放大，到了這一層終於使兩種訊號的比例出現明顯的改變。對於這種情形的出現，在之後的改進中必然要有針對這一層訊號數量及密度上的對應方法，才能更有效的處理這個問題。關於這部分在結論中會立一段說

明可能的對策。

而且這一大批無法被觀察到的訊號又會再遮住 layer 2、layer 1 的訊號。而 layer 2 的訊號量通常也和 layer 4 同個數量級，但可觀察訊號的比例卻往往不如 layer 4。甚至在修改過的電路中，layer 2 的可觀察訊號數有可能因為上層訊號試圖加入 Dummy Metal Fill 提高可觀察性，然而在更加擴大的 FIB 觀察框的影響之下進而使得 layer 2 的可觀察訊號數量下降了。



第五章 結論

在本研究中，我們試圖利用電路中已加入的 Dummy Metal Fill 與訊號連接，提高以 FIB 技術觀察晶片中訊號的可觀察性。在目前所訂定的限制條件下，確實可以提高所測試電路中訊號的可觀察性。此外在文中還探討了朝這個方向進一步發展需要處理的問題，以及未來可能的改進空間。

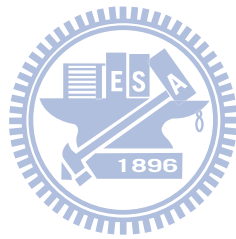
首先，要處理的訊號應該要分群。就像在 ATPG 中不是所有的錯誤都是可以檢測出來的，在電路中有些訊號就是陷在其他訊號的包圍之中，不止找不到可以接上的 Dummy Metal Fill，想要把原本的線路修改都有困難。從另一個角度思考，所在最高 layer 越下層的訊號，通常這種訊號連接點較少，距離也比較短，或許它們的影響層面不會比接點多、連線距離長的訊號更大。至於要分成幾群，以及要以哪些條件分群便需要再思考了。反過來想，給定一個已經 P&R 完成並加入 Dummy Metal Fill 的電路，有沒有辦法有效率的判斷其中某個訊號是不是有救的？再說，就算有些訊號無法被判定為無法觀察到，但是否有辦法估計要把某個訊號用改線路或是接上 Dummy Metal Fill 傳出會比另一個訊號困難？

此外，有沒有辦法知道在這樣的條件下所能達到最高的可觀察訊號比例的上限？在實驗數據一章中可以看到，擁有接近（甚至超過）一半訊號數量的 layer 3 通常都無法將該 layer 的可觀察訊號比例達到一半，這樣算來整個電路理想中的可觀察訊號比例上限或許可以估成 75%。75% 當然不算是什麼好估計，因為這估計的基礎太不實在了；但是在給定了一個電路之後，是否能有效率的算出可觀察訊號比例的上限。

接下來提的就不是問題了，而是進一步改進結果所需要的功能。目前的架構下，檢查是一個訊號是否能被 FIB 觀察到的做法頂多只能稱為負面列表，但是在決定是否該加入某個 Dummy Metal Fill 線段時，卻很需要正面列表。正面列表不一定只是標示某一個區塊看的見，某一個區塊看不見，要是能把選上某個區塊的代價也標示在表上，應該對於選擇 Dummy Metal Fill 有更好的效果。除了更明確的標示每個 Dummy Metal Fill 加入某個訊號中的代價之外，若是想要在之後的實驗中將訊號嘗試接到多層 Dummy Metal Fill，就要設法估計訊號接上新的線段及 via 之後對於電路時間特性的影響。此外，為了更符合 Dummy Metal Fill 加入時的情形，將目前表示訊號線段的架構修改成可以處理寬度超過該 layer 規定訊號線寬度也會有幫助的。

針對 metal 3 的數量及密度問題，目前有可能利用方法是分而治之 (divide and conquer) 在未來的改進中的必要性。我們現在所處理的測試電路其實都算是相對不大的，要是之後要處理更大的電路以及更多的 Dummy Metal Fill，以目前的架構而言很可能執行到一半就無法找到另夠的記憶體裝下需要的資料。再者，如 metal 5、metal 4 的訊號數量少，可觀察度也高，執行一次之後幾乎就不用再多處理，而從 metal 3 開始問題變的複雜，不止要處理的訊號大量增加，而且線

段數也大大超越了前面幾層。為此，以後的程式架構在兩個維度間都應該要可以切割才能更方便的調整。第一個維度是各層間的分離，要是可以各層分離，則比較不好處理的 layer 便可獨立出來修正或切換參數；第二個維度是層內的分區處理，同一層內的各區域其實若沒有共同訊號，也不相鄰的話，其實關聯性還不如不同層但上下相連的分區強。如此分開處理應該可以對不同的分區、分層間的修正更加順利。



第六章 參考書目

- [1] A. L. D'Souza, and M. S. Hsiao, "Error Diagnosis of Sequential Circuits Using Region-Based Model." *Journal of Electronic Testing* , 21 (2), pp. 115-162, 2005
- [2] M. L. Agrawal, V. D. Bushnell, "Essentials of Electronic Testing." Boston, Kluwer, 2000
- [3] FEI Company, "Focused ion beam technology, capabilities and applications." 2006
- [4] L. M. Huisman, "Diagnosis Arbitrary Defect in Logic Designs Using Single Location at A Time (SLAT)." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , 23 (1), pp. 91-101, 2004
- [5] Kai-hui Chang, Igor L. Markov, Valeria Bertacco, "Automating Post-Silicon Debugging and Repair." *International Conference on Computer-Aided Design* (pp. 91-98). 2007
- [6] N. Mukherjee, A. Pogiel, J. Rajski, and J. Tyszer, "High Volumn Diagnosis in Memory BIST Based on Compressed Failure Data." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29 (3), pp. 441-453. 2010
- [7] P. Bernardi, M. Grosso, M. Rebaudengo, and M. Sonza Reorda, "A Pattern Ordering Algorithm for Reducing The Size of Fault Dictionaries." *VLSI Test Symposium* (pp. 389-391). 2006
- [8] H. T. Park, "Characterization and Modeling of Pattern Dependencies in Copper Interconnects for Integrated Circuits." Ph. D. Dissertation. Department of Electrical Engineering and Computer Science, MIT, May 2002
- [9] R. Desineni, and R. Blanton, "Diagnosis of Arbitrary Defecs Using

- Neighborhood Function Extraction." *VLSI Test Symposium*, (pp. 366-373), 2005
- [10] R. Desineni, O. Poku, and R. Blanton, "A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior." *International Test Conference*, (pp. 1-10), 2006
- [11] V. Boppana, R. Mukherjee, J. Jain, M. Fujita, and P. Bollineni, "Multiple Error Diagnosis Based on XLISTS." *Design Automation Conference*, (pp. 660-665), 1999
- [12] X. Wen, S. Kajihara, K. Miyase, Y. Yamato, K. K. Saluja, L.-T. Wang, K. Kinoshita, "A Per-Test Fault Diagnosis Method Based on The X-Fault Model." *IEICE Transactions on Information and Systems* , *E89-D* (11), 2756-2765, 2006
- [13] Y.-C. Lin, F. Liu, and K.-T. Cheng, "Multiple-Fault Diagnosis Based on Adaptive Diagnostic Test Pattern Generation." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , *26* (5), pp. 932-942, 2007

