

ICCAD 2018 CAD Contest

Color-aware Routing for Double Patterning

本題目由對EDA領域充滿熱忱之不具名人士提供

Contents

0. Announcements.....	P2
1. Introduction	P3
2. Problem Description.....	P4
3. Example of Input/Output Files.....	P6
4. Language.....	P6
5. Platform.....	P6
6. Testcases.....	P6
7. Evaluation.....	P6
8. Reference.....	P8
9. FAQ.....	P9

0. Announcements

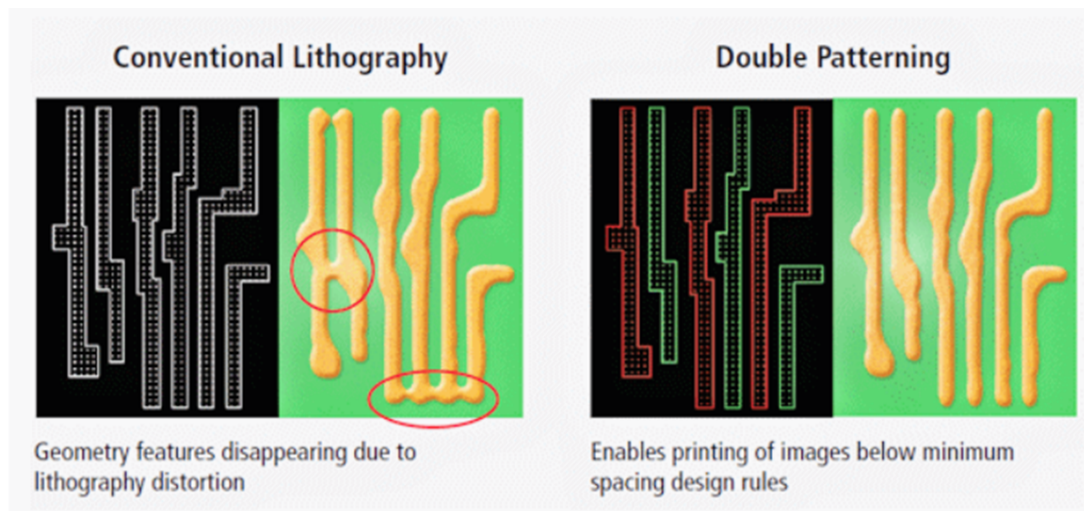
- 2018-05-25- Problem E FAQ is updated.
- 2018-05-20- Problem E FAQ is updated.
- 2018-05-16- Problem E FAQ is updated.
- 2018-05-04- Problem E FAQ is updated.
- 2018-05-09- Problem E Testcase: case1_pin is updated.
- 2018-05-04- Problem E FAQ is updated.
- 2018-05-04- Problem E Testcase: case1_pin is updated.
- 2018-05-03- Problem E FAQ is updated.
- 2018-04-12- FAQ is updated.
- 2018-02-05- Problem is released.

Color-aware Routing for Double Patterning

(本題目由對 EDA 領域充滿熱忱之不俱名人士提供)

Introduction

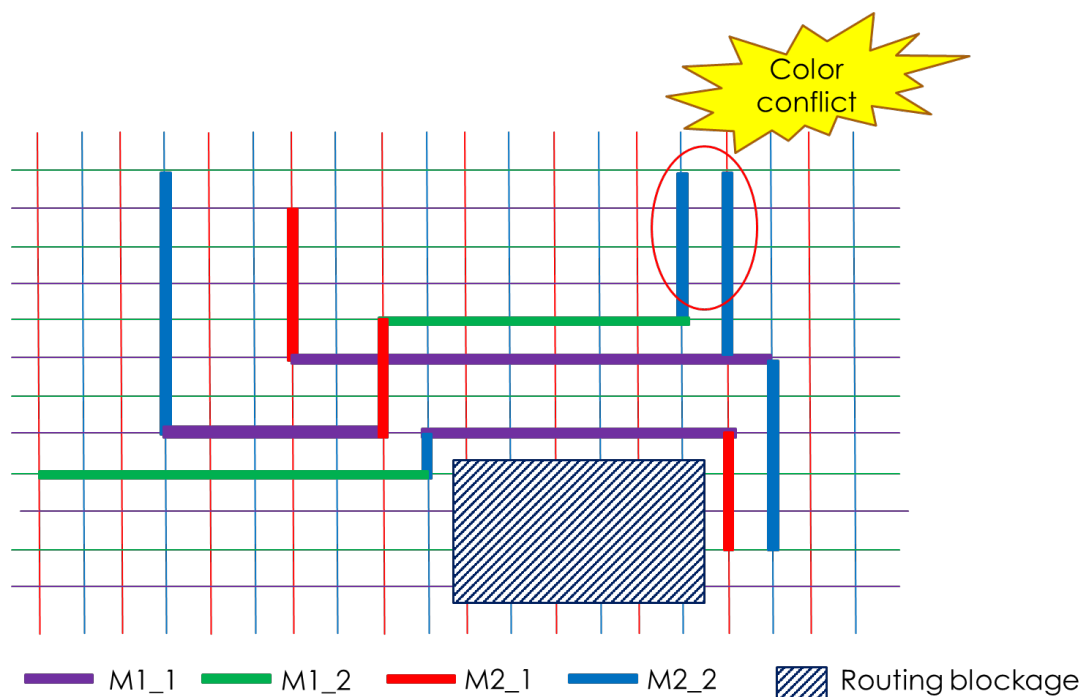
對於 45nm 或以上的積體電路製程，傳統的 193nm 波長浸潤式顯影仍可以滿足最小線寬的需求，但隨著製程進步到 32nm 或以下，單次曝光已無法應付最小線寬的需求，如圖一左邊所示。因此發展出新的雙重曝光(Double Patterning)技術，將同一金屬層曝光分為兩次，如圖一右的紅色與綠色。



圖一：傳統曝光(左)與雙重曝光(右)的比較

因應雙重曝光的技術，繞線時也將每一金屬層分為兩種顏色。圖二是雙重曝光繞線的一個例子，Metal 1 分為綠色與紫色兩種；Metal 2 則分為藍色與綠色兩種。在雙重曝光中，相鄰的相同金屬層必須分別屬於兩種不同的曝光序(以不同顏色代表)，否則會發生 Color Conflict，也就是曝光後會有短路的情況，就像圖一左一樣。在圖二的右上角，由於相鄰兩條 Metal 2 被 assign 到同一曝光層(藍色)，因此就發生了 Color Conflict。

此外在不同曝光層的使用上，平均的使用可以提升良率，也就是對同一金屬層而言，我們盡量平均地指定到兩個曝光層(顏色)上，而不是偏向使用單一曝光層。



圖二：雙重曝光的繞線

Problem Description

給予一個技術規格與 netlist，參賽者必須完成所有 net 的繞線與每個 wire segment 的 metal/color assignment，繞線結果必須避開所有障礙物(blockage)並且確保沒有 open/short violation 與 color conflict，metal 方向必須遵從 preferred direction，詳細的技術規格、輸入與輸出敘述如下：

技術規格部分：

技術規格包含四層 Metal，各層 Metal 的 Layer Number，Routing Pitch 與 Preferred Direction 如下

Layer	Layer No.	Pitch	Preferred Direction
=====			
Metal 1	1	0.5	Horizontal
Metal 2	2	0.5	Vertical
Metal 3	3	0.5	Horizontal
Metal 4	4	0.5	Vertical

每層 Metal 均包含兩種顏色的 mask，分別以數字 1 與 2 代表之，所謂的 Routing Pitch 是指兩條平行的同一層 Metal 在製程上所允許的最小距離，也就是設計規則(Design Rule)裡的最小空間(minimal space)，若小於這個距離就可能會造成相鄰兩條金屬線短路。以圖二為例，Metal 1 的 Routing Pitch 就是相鄰紫色

與綠色 Track 的距離，Metal 2 的則是相鄰紅色與藍色 Track 的距離。所有 Metal 的 Routing Track 都是以 0 為起點，以 Metal 1 為例，各 Track 的 y 座標分別是 0, 0.5, 1.0, 1.5, ... 依此類推。

輸入部分：

1. Pin Definition File (檔名 caseN_pin.in):

包含每個 Pin 的序號，metal layer (layer number)與位置，下列是一個定義檔的範例：

```
1 2 20.5 101.0 ## pin 1, metal 2, 位置 x:20.5 y:101.0
2 1 95.0 33.5 ## pin 2, metal 1, 位置 x:95.0 y:33.5
3 1 6.5 509.0 ## pin 3, metal 1, 位置 x:6.5 y:509.0
```

2. Net Connection File (檔名: caseN_net.in):

包含每條 Net 的序號，所連接的 Pin 序號，以及是否為 critical net，下列是一個範例：

```
1 5 6 7 N ## net 1, 連接 pin 5, 6, 7, 非 critical (N)
2 1 3 4 Y ## net 2, 連接 pin 1, 3, 4, 是 critical (Y)
3 2 9 N ## net 3, 連接 pin 2, 9, 非 critical (N)
```

3. Blockage Definition File (檔名: caseN_blockage.in):

定義所有的 blockage，全部是矩形，以 bounding box 形式表示 (lower left x, lower left y, top right x, top right y)，下列是一個範例：

```
10.0 10.0 30.0 60.0 (左下 x=10 y=10，右上 x=30 y=60)
20.5 30.5 100.0 53.0 (左下 x=20.5 y=30.5，右上 x=100 y=53)
```

輸出部分：

Routing Result File (檔名: caseN_result.out):

包含每條 net 的名稱，每段 wire segment 的座標，使用金屬層與顏色，格式如下：

Net 1

```
# from 20.5, 101.0 to 20.5 33.5 use M2 (2), color 1 (1) 註解，實際輸出不須寫入
+ 20.5 101.0 20.5 33.5 2 1
# from 20.5 33.5 to 95.0 33.5 use M3 (3), color 2(2)
+ 20.5 33.5 95.0 33.5 3 2
# from 95.0 33.5 to 101.5 33.5 use M1 (1), color 1(1)
+ 95.0 33.5 101.5 33.5 1 1
```

繞線結果允許 **stack via** 跳層的使用，例如上面例子裡 **M3** 跳 **M1**。所謂的 **Stack via** 是指當繞線換層的時候，如果換超過一層（一般是相鄰金屬層互換，如 **M1** 接 **M2**），所用的 **via** 就必須以堆疊的形式存在。舉例來說，如果 **M2** 接到 **M5**，就必須用三層 **via**，包括 **Via23**、**Via34** 與 **Via45**，其中 **Via23** 是指 **M2** 到 **M3** 的 **Via**；**Via34** 是指 **M3** 到 **M4** 的 **Via**；**Via45** 是指 **M4** 到 **M5** 的 **Via**。

Example of Input/Output Files

Input:

case1_pin.in:

```
1  2  20.0 20.0
2  1  50.0 50.0
3  1  20.0 50.0
4  2  40.0 50.0
```

case1_net.in

```
1  1  2  N
2  3  4  Y
```

case1_blockage.in

```
30.0 30.0 40.0 40.0
```

Output:

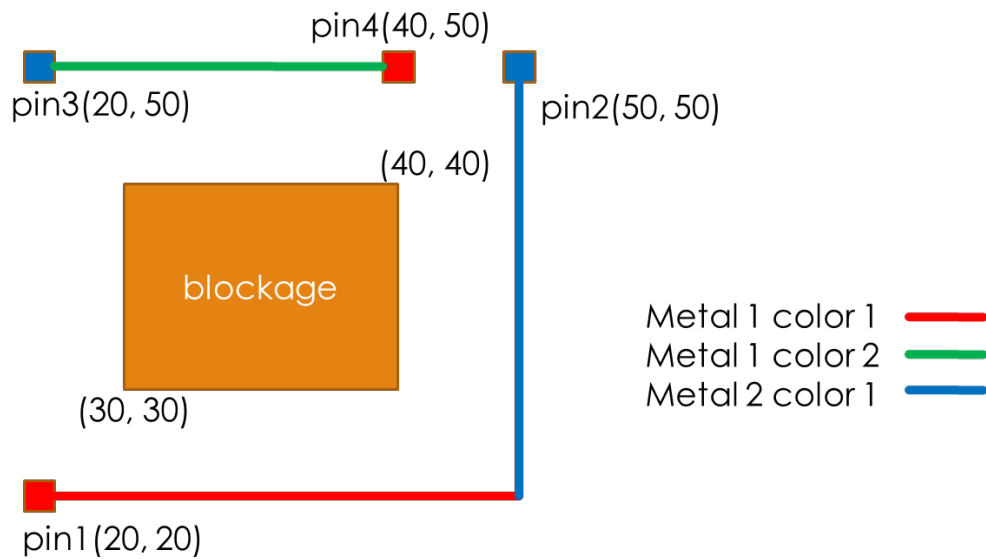
case1_result.out

Net 1

```
+ 20.0 20.0 50.0 20.0  1  1
+ 50.0 20.0 50.0 50.0  2  1
```

Net 2

```
+ 20.0 50.0 40.0 50.0  1  2
```



Language

Please implement your program in C or C++. The binary file should be called as `color_route`. Please follow the following format to get the output file.

```
./color_route case1_pin.in case1_net.in case1_blockage.in case1_result.out
```

Platform

OS: Linux

Compiler: gcc/g++

Testcases

Two open testcase can be downloaded

One non-disclosure testcase

Evaluation

The score is 100 for each testcase. If your program encounters compilation errors, crash (coredump), runs more than 2 hours, the result contains open/short/metal direction/blockage violations or color conflict, the score for this test case will be 0.

The final score is the average of the scores of all the test cases. The score is combined by the following parts:

1. Overall wire length (35%)
2. Number of vias (20%)
3. Detour ratio of critical nets (35%)
4. Mask usage balancing (10%)

For 1, the overall wire length of all participants are ranked. The top rank (minimal overall wire length) will get score 35 and 2nd place get 34, 3rd place get 33, and so on.

For 2, number of vias are also ranked. Stack via will calculate by the number of changed layers. For example, via number of stack via from metal 1 to metal 4 is 3 (4-1=3). The first place will get score 20, 2nd get 19, and so on.

For 3, the detour ratio is the overall wire length of all critical nets divided by half perimeter wire length of all critical nets. All participants are ranked by their detour ratio. The 1st place will get score 35, 2nd place will get 34, and so on.

For 4, the mask usage balancing is calculated by the following formula: For metal N, mask usage balance ratio = $\text{abs}(50\% - (\text{overall wire length of metal N color 1}) / (\text{overall wire length of metal N}))$. Overall mask usage balancing ratio is the average of mask usage balancing ratio of all metals. All participants are ranked by their overall mask usage balancing ratio. The 1st to 3rd place will get score 10, 4th to 6th place will get score 9, 7th to 9th place will get score 8, and so on.

For example, if one player get 1st place in 1, 3rd place in 2, 2nd place in 3, and 5th place in 4 for one testcase, his score for this testcase is $35 + 18 + 34 + 9 = 96$.

Reference

1. https://en.wikipedia.org/wiki/Multiple_patterning
2. K. Yuan, J.S. Yang, D.Z. Pan, "Double Patterning Layout Decomposition for Simultaneous Conflict and Stitch Minimization", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (Volume: 29, Issue: 2, Feb. 2010)
3. M. Cho, Y. Ban, D.Z. Pan, "Double patterning technology friendly detailed routing", Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, Pages 506-511
4. S.I. Lei, Chris Chu, W.K. Mak, "Double patterning-aware detailed routing with mask usage balancing", 15th International Symposium on Quality Electronic Design, 2014
(<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.707.2241&rep=rep1&type=pdf>)

9. FAQ

A1. 提供的測資中，Net 的連接順序是需要按照測資由 1 至 33 嗎？

還是可以用最佳化的方式自由選擇第一條連接的 Net？

Q1. 順序可自由決定，不需按照編號。

A2. 在同一層 Metal 層中只要有兩條線小於 0.5，就會造成 congestion。

那 Color Conflict 的部分，也是只要同層的兩條線小於 0.5 就會發生嗎？

Q2. 是的，小於 0.5 不管是不是同色都會發生。

A3. 請問我們使用的 gcc 有規定大家是用何種版本嗎？

Q3. 版本沒有特別規定，能產生可執行碼即可。

Q4 依題目中有敘述，兩條平行的同一層 Metal 若小於 0.5 這個距離就可能會造成相鄰兩條金屬線短路。但關於 Color Conflict 只有提到相鄰就會發生，請問相鄰的距離也是一樣小於 0.5 嗎？如果是的話請問有包含 0.5 嗎？

A4. 是的，有包含 0.5，也就是相鄰的同 Color 距離必須大於 0.5，在本題 (Grid-based Router) 即是必須相距兩個 Routing Track。

Q5 請問貴單位的 g++ / gcc 是什麼版本？

A5. GCC-4.4.7。

Q6. Metal 層最多只有 4 層嗎？還是會依題目要求而超過 4 層？

Q6: Yes, 只有四層。

Q7 請問 Pin 序號會不會以不同座標的複數方式存在？

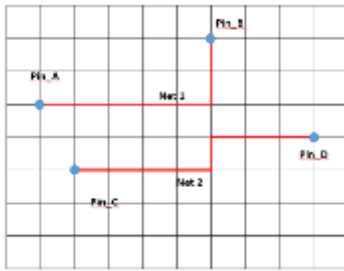
ex;

1 2 20.5 101.0 ## pin 1, metal 2, 位置 x:20.5 y:101.0

1 1 95.0 33.5 ## pin 1, metal 1, 位置 x:95.0 y:33.5

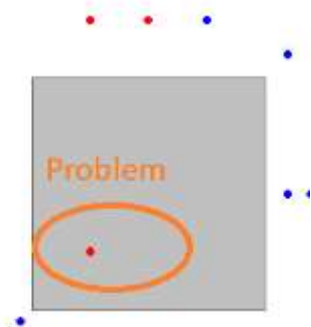
A7. 不會，目前沒有 equivalent pin 的設計，所以 Pin 序號都是唯一座標。

Q8. 請問下圖的顏色佈置會造成 color conflict 嗎?



A8. 會

Q9. 請問 case1 及 case2 的 pin 點出現在 block 中，如下圖，是 testcase 錯了，還是 block 中可以繞線呢?



A9. 我自己看 pin 的位置並沒有在 blockage 裡面，請問是第幾根 pin? blockage 裡面應該完全不能繞線。

Q10. 承上，Case1 中 Pin40 座標為(15, 15)，第一個 block 則是(10, 10)到(30, 30)，這樣是否是在 block 中?

A10. 這個 Case 座標有誤，請將 Pin40 座標 (15, 15) 改為 (15, 5)，修改後的檔案已公告在網路，請重新下載並更新，<http://iccad-contest.org/2018/tw/problems.html>。

Q11. 請問 block 沒有給 layer 層，代表是 1~4 層都會擺一樣位置的 blocks 嗎?

A11. blockage 是對所有層，也就是完全不能繞線。

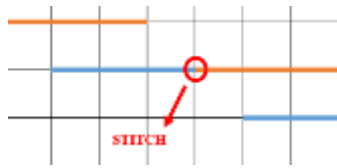
Q12. 請問其他 case 及驗證軟體什麼時候會釋出呢?

A12. 另一個 case 是 blind test 不會釋出，另外請自行驗證結果，我們不會提供驗證軟體。

Q13. 在 6/22 日為 alpha test 時，blind case 會加入測試嗎?還是只會測試兩個公開的 case?

A13. 只會測試兩個公開的 case，最後評分時才會加入 blind test。

Q14. 請問在這次比賽中允許 stitch 嗎？如下圖所示。



A14. 本題不允許 stitch。

Q15. 關於釋出的測資中，有提供一個 testbecnch.v 供 verify 這個檔案的測試，是不是少了「如果 128 無法整除電路的輸入加輸出數，則必須補 0」這項規定呢？

A15. 因為 verilog 讀取檔案資料時，沒有設定值得部分，會自動補”0”，我們檢查了測試的程式，仍會驗證補”0”的測資，故有考量這項規定。

Q16. 在 Problem E 中，layout 的座標有限制嗎？(x, y 的範圍)

A16. layout 的座標並無有限制。