

雙重曝光下的平衡拆解

Color Balancing for Double Patterning

指導教授：林英超

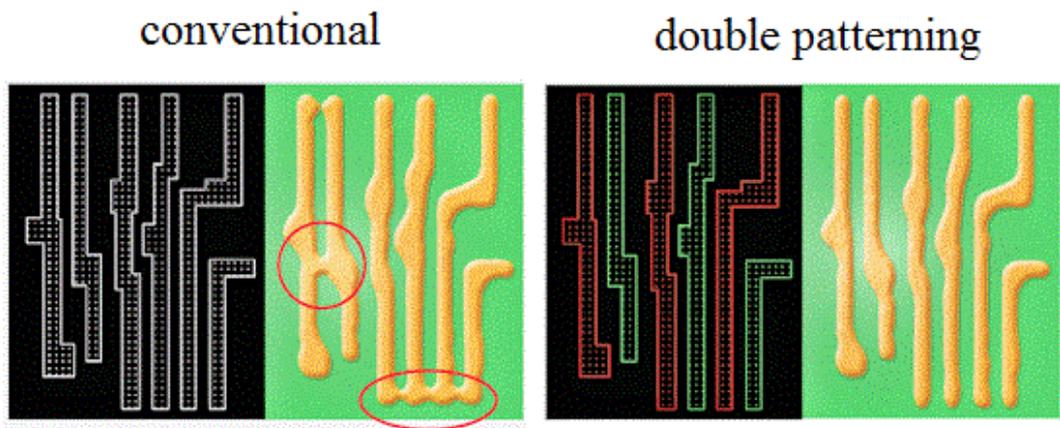
專題成員：陳昱文、林建宏

開發工具：G++ (C++11)

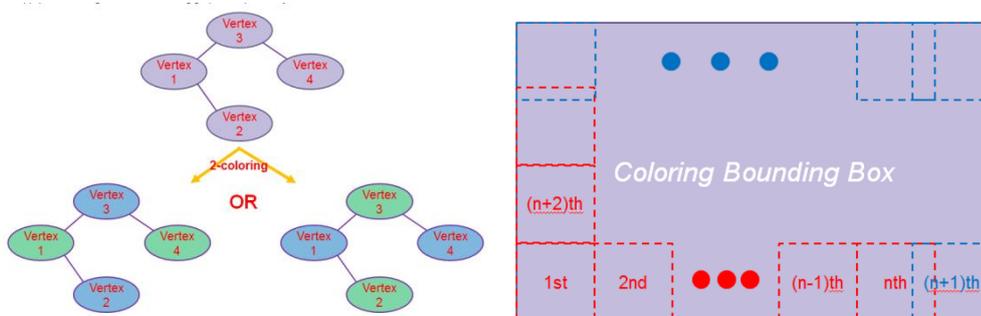
測試環境：Linux RedHat 5.9

一、簡介

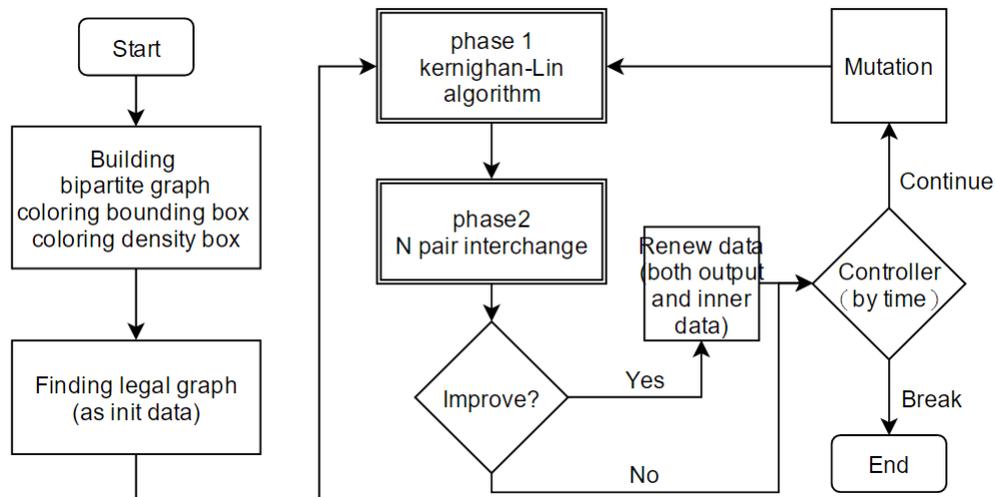
當半導體光刻技術使用波長 193nm 光線，並且在約 45nm 奈米製程曝光時，由於光的特性導致光罩產生出來的圖形互相干擾，產生非預期的成品，因此 multiple patterning 應運而生，multiple patterning 藉由將光罩拆分兩個(或是以上)交替進行曝光避免過於相近的蝕刻處互相影響。



為了進行圖形拆解，首先我們必須將互相在一定距離內的圖形拆分，並且以此關係尋找所有可以產生二分圖的圖形，所有圖形應該在 coloring bounding box 內的 coloring density windows 中決定所屬的顏色，最終目標是使所有 coloring density windows 內的兩種顏色面積差總和最小化。



我們實做的演算法受到 Kernighan - Lin algorithm 與基因演算法啟發，以下是我們的基本程式流程圖。



Kernighan - Lin algorithm 原本是用在圖形分割的演算法，它進行分割的條件可以經過部分修改後套用到我們的題目上，並具有極高的效率進行最佳化。

Phase2 進行二對（或是以上，但考慮複雜度只使用到二對）圖形顏色的交換，由於 KL algorithm 內只進行一對圖形顏色的轉換，程式執後將會到達 KL algorithm 所能到達的瓶頸，因此我們需要利用增加交換組數來增加進步可能性，藉此改變解的狀態，並且希望可以打破 KL 的瓶頸。

由於 Phase2 是以增加單次圖形轉換對數提升解題能力，但其複雜度將隨著轉換對數以指數成長，因此我們不能單單依靠增加轉換對數來增強解題能力，所以我們將部分圖形（我們採取 10% 隨機選取）轉置，重新進行 Phase1&2 來產生新的初始解，因此這樣的方法雖然不能確保每次都找到更佳解，但是由於我們每次進行 Phase1&2 耗時很短，因此可以藉由多次重複解題來改進解的分數。

二、測試結果

