

# FPGA Final Project Report

111062676 張品謙

## Problem

設計一個程式來分割電路節點到多個 FPGA, 最小化所有 FPGA 的 external degree 總和, 並同時滿足以下三個限制條件:

1. 固定節點限制(Fixed node constraints): 某些節點已指定對應的 FPGA, 不可變更。
2. 容量限制(Partition size constraints): 每個 FPGA 可容納的節點數 不得超過其容量。
3. 拓撲限制(Topology constraints): 若  $u$  是某 net 的 source、 $v$  是 sink, 則
  - $u$  和  $v$  要嘛在同一個 FPGA (即  $FPGA(u) = FPGA(v)$ )
  - 或者  $u$  和  $v$  所在的 FPGA 要相鄰 (即  $(FPGA(u), FPGA(v)) \in E'$ )

(FPGA  $F_i$  的 external degree 被定義為: 若 net 的 source 在此  $F_i$  但是某 sink 不在, 則此  $F_i$  的 external degree 加一)

## Idea

1. 初始化候選 **FPGA (candidates)**:
  - 針對與固定節點相連的鄰居節點(sink), 縮小其可選的候選 FPGA 範圍。
  - 每個鄰居節點的候選 FPGA 集合為:  
鄰居節點的候選 FPGA 交集 與他相連的固定節點所屬的 FPGA 與此 FPGA 相鄰的 FPGA。
2. 節點放置優先順序策略 (使用 **Priority Queue Q**):
  - 第一優先: candidates 數量越少, 越優先處理 (避免未來無法放置)。  
若某節點無法放置, 則視為重大違規, external degree 增加兩倍作為懲罰。
  - 第二優先: net 的 sink 數量越少者越優先放置。這樣比較可能使 external degree 為 0。因為 nets 是只要一個 sink 不在同一個 FPGA 則增加 1, 就算多個 sink 不在也是增加 1, 因此優先幫助 sink 少的 nets, 盡可能避免增加 degree。
3. 選擇放置在哪個 **FPGA** (使用 **Priority Queue R**):
  - 第一優先: 此放置所造成的 external degree 增加量最小。

- 第二優先:選擇當前 `fpga_capacity_predict` 值較小的 FPGA。  
`fpga_capacity_predict[i]` 表示目前有幾個節點能放在 `FPGAi`。越多人能放進該 FPGA, 代表這個FPGA越有價值, 因此優先選擇對其他節點需求較少的 FPGA放, 這樣更有機會放完全部。
- 目前這個方法只能用在node數量為20000以下, 20000以上我是用, 先固定fixed node, 再照順序的方式將node放進FPGA(從FPGA 0開始放, 滿了就換FPGA 1, 以此類推), 但盡量讓nets中的node放在 同一個FPGA(舉例來說, 若node 1放在FPGA 0, 那我就讓nets[1]中的node在不超過當前FPGA capacity的前提下放在FPGA 0), 來盡量減少external\_degree, 這方式比完全照順序放可以減少快一倍。