

# Physical Design PA1 FM partitioning

我的程式的主要動作是，初始化cell位置後，依序做Gain initialization、Bucket list initialization、FM partition。

FM的過程是，持續挑Gain最大的node、計算Gain、執行node的移動、reset。一直重複直到max accumulative gain不大於零。

## FM

---

```
void Partitioner::FM() {
    do {
        while(_unlockNum != 0) {
            pickFM();
        }
        execFM();
        reset();
    } while (_maxAccGain > 0);
    return;
}
```

---

## pickFM

---

```
inline void Partitioner::pickFM() {
    // pick cell has max gain
    pickMaxGainCell(_maxPinNum);
    auto candiCell = _cellArray[_maxGainCell->getId()];
    // if unbalanced, pick the other part
    if(!balance(candiCell)) {
        pickPartMaxGainCell(!candiCell->getPart(), _maxPinNum);
        candiCell = _cellArray[_maxGainCell->getId()];
    }
    // record candidate cell for update bucket list
    _moveStack.emplace_back(candiCell->getNode()->getId());
    _gainStack.emplace_back(candiCell->getGain());
    updateCellGain(candiCell);
    moveCell(candiCell);
    lockCell(candiCell);
    updateBList();
    return;
}
```

---

我的Bucket List的結構是

MaxPinNum	head node ->	head node	// no cell here		
	head node ->	c1304 ->	c3308 ->	head node	
	head node ->	unlocked node ->	—	locked node ->	headnode
	head node ->	head node			
-MaxPinNum	head node ->	head node			

我的bucket會建成Circular Linked List, unlocked的node會insert在head node的後面, locked的node會insert在head node的前面。  
因此每次iteration不需要再重新建立一次Bucket list。  
以及insert、delete皆是O(1)。

結果

	Iteration	Cutsizes	Cell	Net	Part A	Part B	5500U	G5218
input_0	28	2267	150750	166998	67982	82768	32.4 s	31 s
input_1	9	1241	3000	5000	1487	1513	0.032 s	0.024 s
input_2	8	2226	7000	10000	3484	3516	0.074 s	0.064 s
input_3	23	26893	66666	88888	30000	36666	4.75 s	2.5 s
input_4	27	44758	150750	166998	74622	76128	11.9 s	7.5 s
input_5	21	142608	382489	483599	189333	193156	29.5 s	21 s

結語

- 以下是我覺得結果可能會變好的幾種方法：
- 1. 嘗試先找到比較好的初始位置, 我試過依照net先集中放一邊, 再放剩餘的cell, 不過結果是沒有比較好。
  - 2. 做完FM後, 擾動0.5%~20%的cell, 再做一次FM, 但我試了整體的結果沒有比較好, 還是會卡在 local minimum。
  - 3. 相同gain的cell, 想辦法取最好的那一個。
  - 4. 先把cell cluster成比較大的cell, 整體node的數量可以大幅減少, 再做FM, 應該可以大幅加速FM的執行時間, 但cutsizes的結果可能會比較差一點。