

EE6094
CAD for VLSI Design
Programming Assignment 4 Report

Student Name:魏子翔

Student ID:107501019

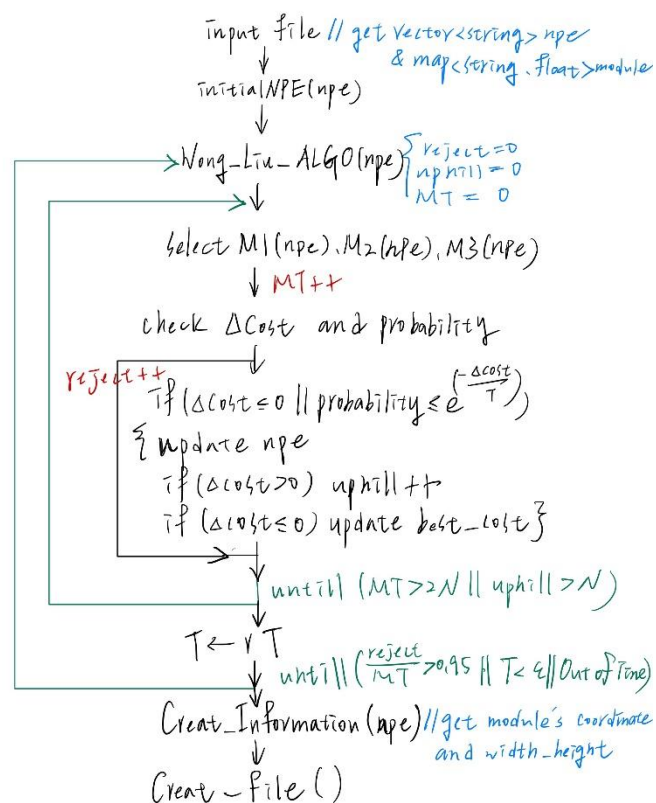
Abstract

為了縮小 chip 擺設的面積，所以我們將要寫出一套演算法去將所有的 block 可以擺進最小的 bounding box，及盡可能的減少 dead space。

I. Problem Description

本次 PA 主要是要將給定面積的 soft block module，藉由 Wong Liu Algorithm 中的 SA 及 NPE 去將整體的面積給予最小化，並且不會發生 overlapping。

II. Program Structure



III. Data Structures / Algorithms Used

```
map<string, float> module;
```

首先會利用上方的 *map* 去儲存 input file 所得到的資訊，*string* 為 key，用來存 module 的名字，*float* 則存對應的面積。

```
vector<string> npe;
```

再來會利用 *vector* 把 NPE 建立好。

```
class block{    //NPE 中的 block
    string name;
    vector<pair<float, float> > WidthHeight;
    friend class FloorPlanning;

public:
    block(string a,vector<pair<float,float> >b):name(a),WidthHeight
(b){}    //constructor
};

class FloorPlanning{    //floorplanning
    map<string, float> module;

public:
    map<string, pair<float, float> > F_WidthHeight; //最終的長寬
    map<string, pair<float, float> > F_coordinate; //最終的座標

    FloorPlanning(map<string, float>m):module(m){} //constructor
    void InitialNPE(vector<string>&npe);
    vector<block *>createNPE_Block(vector<string>&npe);
    void Combination(block*operand1, block*operand2, block*operator
1);

    float Cost(vector<string>&npe);
    bool Check_consecutive(vector<string>&npe);
    bool Check_ballot(vector<string>&npe);
    void M1(vector<string>&npe);
    void M2(vector<string>&npe);
    void M3(vector<string>&npe);
```

```

void Wong_Liu_ALGO(vector<string>&npe);
void Create_Information(vector<string>&npe);
void printNPE(vector<string>&npe);
};

```

讀取玩 input file 及建立好 NPE 後，就會進入到上方的資料結構及演算法，那首先我們會先將一開始創建好的 *map<string, float>module* 作為 *class FloorPlanning* 的結構子，再來就會進入演算法的部分。

1. *vector<block*>createNPE_Block(vector<string>&npe)* 主要是根據 NPE 來建立 block，block 的資料結構如上方程式碼最一開始所示，當中會儲存 block 的名字及 comperand block 的所有寬高組合。
2. *void Combination(block*operand1, block*operand2, block*operator1)* 則是根據 NPE 將建立好的 block 去做結合，結合出每個 operand1 block 及 opeand2 block 所有的寬高組合，來生成對應 operator block 的寬高組合。
3. *void Cost(vector<string>&npe)* 是將目前的 NPE 所花費的 area cost 算出來，

```
vector<block*>NPE_block = createNPE_Block(npe); //將 NPE 創建 block
```

首先會利用 *createNPE_Block* 來將 NPE 建立 Block

```
stack<block*>block_stack; //用 stack 的方法去做 combination
```

接著會皆由 *stack* 去輔助 *block* 去做結合，因為 NPE 是以 postorder 的形式表示，所以從 *NPE_block* 的頭開始尋訪，當遇到 operand 就將其 *push* 進 *stack* 裡，若遇到 operator 則從 *stack* 裡 *pop* 兩個出來 進行 *Combination(block*operand1, block*operand2,*

*block*operator1*)，結合好後再 *push* 進 *stack* 裡，不斷重複此步驟直到尋訪完。最後利用 *NPE_block* 最後一個 *block* 的寬高組合來求出最小的 *area cost*。

4. *bool Check_consecutive(vector<string>&npe)* 和 *bool Check_ballot(vector<string>&npe)*則是用來檢驗經過 M3 的交換之後 NPE 是否還符合規定。
5. *void M1(vector<string>&npe)*、*void M2(vector<string>&npe)*、*void M3(vector<string>&npe)*則是 Wong Liu Algorithm 中的 3 中交換方式。
6. *void Wong_Liu_ALGO(vector<string>&npe)*則是這次 floorplanning 主要的優化演算法，將 NPE 丟入後可以得到優畫好的 NPE。
7. *void Create_Information(vector<string>&npe)*是將優畫好的 NPE 的座標及寬高資訊建立好。

IV. Difficulty Encountered

這次的 PA 主要都還是照著老師的上課講義慢慢打的，最一開始主要是程式總是會再執行 M2 的交換時當掉，一開始不太知道是甚麼原因，不斷的將程式資訊輸出，發現所有的 case 都會跑一陣子後再 M2 時當掉，再更深入去探究就發現都是在將最後一對 *operator chain inverse* 時停掉

```
while((npe[i]=="H" || npe[i]=="V") && i < npe.size())
{
    if(npe[i] == "H")
        npe[i] = "V";
    else if(npe[i] == "V")
        npe[i] = "H";
    i++;
}
```

原本再執行 operator chain inverse 的迴圈是這樣寫，想說 i 只要超過 `npe.size()` 就會跳出運算，但研究了好一陣子後才研究出 `npe[i]` 取不出值程式才當掉

```
while(npe[i]=="H" || npe[i]=="V")
{
    if(npe[i] == "H")
        npe[i] = "V";
    else if(npe[i] == "V")
        npe[i] = "H";
    i++;
    if(i >= npe.size())
        break;
}
```

最後將程式改成這樣後程式就不會當在 M2 了。

之後在 t10.txt、t20.txt 都很順利的可以執行完，但是到了 t30.txt、t200.txt 及 t300.txt 時都會跑出 bad_alloc(core dumped) 記憶體崩潰，程式執行時都會把記憶體佔到 1500 多 MB，想說是不是 vector 的關係，後來在每個 function 結束時都把 vector 釋放，但最後仍然沒解決問題，上網查了一陣子，查到了似乎是跟 new 有關，因為我在 `Combination()` 中會使用 new

去建立 **block**，原本想說函式結束時，這個宣告也會跟著消滅，但事實並不會，所以在那函式中就經過不知道多少次 **for** 迴圈的創建，導致記憶體被佔滿，最後又額外在另一個函式中將其釋放後，全部的 **case** 都能很順利的完成，**t300.txt** 最高記憶體用量也只用到了 **2.4MB** 和原本相差了 **60** 多倍。

V. Experimental Results

t10.txt

```
[107501019@eda359_forclass ~/hw4]$ make run INPUT=t10.txt OUTPUT=t10out.txt
./go t10.txt t10out.txt

Initial NPE is 0 1 V 2 H 3 V 4 H 5 V 6 H 7 V 8 H 9 V
Initial cost is 51554.012

Optimized NPE is 7 1 H 9 6 H V 2 8 0 4 5 V H V H V 3 V
Optimized cost is 31156.906

Spend 29.530 s
```

```
[107501019@eda359_forclass ~/hw4]$ ./PA4_checker.out t10.txt t10out.txt
Expression is correct!!
Module size is correct.
Overlapping check is correct.
All correct! Will done.
```

t20.txt

```
[107501019@eda359_forclass ~/hw4]$ make run INPUT=t20.txt OUTPUT=t20out.txt
./go t20.txt t20out.txt

Initial NPE is 0 1 V 10 H 11 V 12 H 13 V 14 H 15 V 16 H 17 V 18 H 19 V 2 H 3 V 4 H 5 V 6 H 7 V 8 H 9 V
Initial cost is 6158192.500

Optimized NPE is 0 11 V 1 V 13 V 15 V 2 V 4 12 H V 19 10 H V 3 14 H V 16 V 6 V 9 V 7 V 18 V 17 V 8 V 5 V
Optimized cost is 2294548.250

Spend 317.320 s
```

```
[107501019@eda359_forclass ~/hw4]$ ./PA4_checker.out t20.txt t20out.txt
Expression is correct!!
Module size is correct.
Overlapping check is correct.
All correct! Will done.
```

t30.txt

```
[107501019@eda359_forclass ~/hw4]$ make run INPUT=t30.txt OUTPUT=t30out.txt
./go t30.txt t30out.txt

Initial NPE is 0 1 V 10 H 11 V 12 H 13 V 14 H 15 V 16 H 17 V 18 H 19 V 2 H 20 V 21 H 22 V 23 H 24 V 25 H 26 V 27 H
28 V 29 H 3 V 4 H 5 V 6 H 7 V 8 H 9 V

Initial cost is 429831.969

Optimized NPE is 13 1 3 V H 26 27 V H 12 14 V H 11 21 V H 7 22 V H 20 23 V H 8 10 V H 4 H 15 19 2 H V H 16 H 5 H 17
0 V H 6 H 18 9 25 H V H 24 29 V H 28 H

Optimized cost is 110605.031

Spend 1376.080 s
```

```
[107501019@eda359_forclass ~/hw4]$ ./PA4_checker.out t30.txt t30out.txt
Expression is correct!!
Module size is correct.
Overlapping check is correct.
All correct! Will done.
```

t200.txt

```
[107501019@eda359_forclass ~/hw4]$ ./hw4.out t200.txt t200test.txt

Initial NPE is 0 1 V 10 V 100 V 101 V 102 V 103 V 104 V 105 V 106 V 107 V 108 V 109 V 11 V 1
10 V 111 V 112 V 113 V 114 V 115 V 116 V 117 V 118 V 119 V 12 V 120 V 121 V 122 V 123 V 124
V 125 V 126 V 127 V 128 V 129 V 13 V 130 V 131 V 132 V 133 V 134 V 135 V 136 V 137 V 138 V 1
39 V 14 V 140 V 141 V 142 V 143 V 144 V 145 V 146 V 147 V 148 V 149 V 15 V 150 V 151 V 152 V
153 V 154 V 155 V 156 V 157 V 158 V 159 V 16 V 160 V 161 V 162 V 163 V 164 V 165 V 166 V 16
7 V 168 V 169 V 17 V 170 V 171 V 172 V 173 V 174 V 175 V 176 V 177 V 178 V 179 V 18 V 180 V
181 V 182 V 183 V 184 V 185 V 186 V 187 V 188 V 189 V 19 V 190 V 191 V 192 V 193 V 194 V 195
V 196 V 197 V 198 V 199 V 2 V 20 V 21 V 22 V 23 V 24 V 25 V 26 V 27 V 28 V 29 V 3 V 30 V 31
V 32 V 33 V 34 V 35 V 36 V 37 V 38 V 39 V 4 V 40 V 41 V 42 V 43 V 44 V 45 V 46 V 47 V 48 V
49 V 5 V 50 V 51 V 52 V 53 V 54 V 55 V 56 V 57 V 58 V 59 V 6 V 60 V 61 V 62 V 63 V 64 V 65 V
66 V 67 V 68 V 69 V 7 V 70 V 71 V 72 V 73 V 74 V 75 V 76 V 77 V 78 V 79 V 8 V 80 V 81 V 82
V 83 V 84 V 85 V 86 V 87 V 88 V 89 V 9 V 90 V 91 V 92 V 93 V 94 V 95 V 96 V 97 V 98 V 99 V

Initial cost is 611770.938

Optimized NPE is 1 0 H 100 V 10 V 101 V 102 V 103 V 104 V 105 V 107 V 106 V 108 V 11 V 109 V
110 V 111 V 113 V 112 V 114 V 116 V 115 V 117 V 118 V 119 V 12 V 120 V 121 V 122 V 123 V 12
5 V 124 V 126 V 127 V 129 V 128 V 13 V 131 V 130 V 132 V 133 V 134 V 135 V 136 V 137 V 138 V
139 V 14 V 140 V 142 V 141 V 143 V 145 V 144 V 146 V 147 V 148 V 149 V 15 V 150 V 151 V 152
V 153 V 154 V 155 V 156 V 157 V 158 V 159 V 16 V 160 V 161 V 162 V 163 V 164 V 165 V 166 V
168 V 167 V 169 V 170 V 17 V 171 V 172 V 173 V 174 V 176 V 175 V 178 V 177 V 179 V 180 V 18
V 181 V 183 V 182 V 184 V 186 V 185 V 188 V 187 V 19 V 189 V 190 V 191 V 192 V 193 V 194 V 1
95 V 196 V 197 V 198 V 199 V 2 V 20 V 21 V 22 V 23 V 24 V 25 V 26 V 27 V 28 V 3 V 29 V 30 V
32 V 31 V 33 V 34 V 35 V 36 V 37 V 4 V 38 V 39 V 40 V 42 V 41 V 43 V 44 V 46 V 45 V 47 V 48
V 5 V 49 V 51 V 50 V 52 V 53 V 54 V 55 V 56 V 57 V 58 V 6 V 59 V 60 V 61 V 62 V 63 V 64 V 65
V 66 V 67 V 69 V 68 V 7 V 70 V 73 V 71 V 72 V 74 V 75 V 76 V 77 V 79 V 80 V 8 V 78 V 81 V 8
2 V 83 V 84 V 85 V 87 V 86 V 88 V 89 V 9 V 90 V 91 V 92 V 93 V 94 V 95 V 98 V 97 V 96 V 99 V

Optimized cost is 610651.312

Spend 3428.480 s
```

```
[107501019@eda359_forclass ~/hw4]$ ./PA4_checker.out t200.txt t200test.txt
Expression is correct!!
Module size is correct.
Overlapping check is correct.
All correct! Will done.
```


t300.txt

```
[107501019@eda359_forclass ~/hw4]$ ./hw4.out t300.txt t300test.txt

Initial NPE is 0 1 V 10 V 100 V 101 V 102 V 103 V 104 V 105 V 106 V 107 V 108 V 109 V 11 V 1
10 V 111 V 112 V 113 V 114 V 115 V 116 V 117 V 118 V 119 V 12 V 120 V 121 V 122 V 123 V 124
V 125 V 126 V 127 V 128 V 129 V 13 V 130 V 131 V 132 V 133 V 134 V 135 V 136 V 137 V 138 V 1
39 V 14 V 140 V 141 V 142 V 143 V 144 V 145 V 146 V 147 V 148 V 149 V 15 V 150 V 151 V 152 V
153 V 154 V 155 V 156 V 157 V 158 V 159 V 16 V 160 V 161 V 162 V 163 V 164 V 165 V 166 V 16
7 V 168 V 169 V 17 V 170 V 171 V 172 V 173 V 174 V 175 V 176 V 177 V 178 V 179 V 18 V 180 V
181 V 182 V 183 V 184 V 185 V 186 V 187 V 188 V 189 V 19 V 190 V 191 V 192 V 193 V 194 V 195
V 196 V 197 V 198 V 199 V 2 V 20 V 200 V 201 V 202 V 203 V 204 V 205 V 206 V 207 V 208 V 20
9 V 21 V 210 V 211 V 212 V 213 V 214 V 215 V 216 V 217 V 218 V 219 V 22 V 220 V 221 V 222 V
223 V 224 V 225 V 226 V 227 V 228 V 229 V 23 V 230 V 231 V 232 V 233 V 234 V 235 V 236 V 237
V 238 V 239 V 24 V 240 V 241 V 242 V 243 V 244 V 245 V 246 V 247 V 248 V 249 V 25 V 250 V 2
51 V 252 V 253 V 254 V 255 V 256 V 257 V 258 V 259 V 26 V 260 V 261 V 262 V 263 V 264 V 265
V 266 V 267 V 268 V 269 V 27 V 270 V 271 V 272 V 273 V 274 V 275 V 276 V 277 V 278 V 279 V 2
8 V 280 V 281 V 282 V 283 V 284 V 285 V 286 V 287 V 288 V 289 V 29 V 290 V 291 V 292 V 293 V
294 V 295 V 296 V 297 V 298 V 299 V 3 V 30 V 31 V 32 V 33 V 34 V 35 V 36 V 37 V 38 V 39 V 4
V 40 V 41 V 42 V 43 V 44 V 45 V 46 V 47 V 48 V 49 V 5 V 50 V 51 V 52 V 53 V 54 V 55 V 56 V
57 V 58 V 59 V 6 V 60 V 61 V 62 V 63 V 64 V 65 V 66 V 67 V 68 V 69 V 7 V 70 V 71 V 72 V 73 V
74 V 75 V 76 V 77 V 78 V 79 V 8 V 80 V 81 V 82 V 83 V 84 V 85 V 86 V 87 V 88 V 89 V 9 V 90
V 91 V 92 V 93 V 94 V 95 V 96 V 97 V 98 V 99 V

Initial cost is 823160.625

Optimized NPE is 1 0 V 10 H 100 V 101 V 102 V 103 V 104 V 105 V 106 V 107 V 108 V 109 V 110
V 11 V 111 V 112 V 113 V 114 V 115 V 116 V 117 V 118 V 119 V 12 V 120 V 121 V 122 V 123 V 12
4 V 125 V 126 V 127 V 128 V 129 V 13 V 130 V 131 V 132 V 133 V 134 V 136 V 135 V 137 V 138 V
139 V 14 V 140 V 141 V 142 V 143 V 144 V 145 V 146 V 148 V 147 V 149 V 15 V 150 V 151 V 152
V 153 V 154 V 155 V 157 V 156 V 158 V 159 V 16 V 160 V 161 V 162 V 163 V 164 V 165 V 166 V
167 V 168 V 169 V 17 V 170 V 171 V 172 V 173 V 174 V 175 V 176 V 177 V 179 V 178 V 18 V 180
V 181 V 182 V 183 V 184 V 185 V 186 V 187 V 188 V 189 V 19 V 191 V 190 V 192 V 193 V 194 V 1
95 V 196 V 197 V 198 V 199 V 2 V 20 V 200 V 201 V 202 V 203 V 204 V 205 V 206 V 207 V 208 V
209 V 21 V 210 V 211 V 212 V 213 V 214 V 215 V 217 V 216 V 218 V 219 V 22 V 220 V 222 V 221
V 223 V 224 V 225 V 226 V 227 V 228 V 229 V 23 V 230 V 231 V 232 V 234 V 233 V 235 V 236 V 2
37 V 238 V 24 V 239 V 240 V 241 V 242 V 243 V 244 V 245 V 247 V 246 V 248 V 249 V 25 V 250 V
251 V 252 V 253 V 254 V 255 V 256 V 257 V 258 V 259 V 26 V 260 V 261 V 262 V 263 V 264 V 26
5 V 266 V 267 V 268 V 269 V 27 V 270 V 271 V 272 V 273 V 274 V 275 V 276 V 277 V 278 V 279 V
28 V 280 V 281 V 282 V 283 V 284 V 285 V 286 V 287 V 288 V 289 V 290 V 29 V 291 V 292 V 293
V 294 V 295 V 296 V 297 V 298 V 299 V 3 V 30 V 31 V 32 V 33 V 34 V 36 V 35 V 37 V 38 V 39 V
4 V 40 V 41 V 42 V 43 V 44 V 45 V 46 V 47 V 48 V 49 V 5 V 50 V 51 V 52 V 53 V 54 V 55 V 56
V 58 V 57 V 59 V 6 V 60 V 61 V 62 V 63 V 64 V 65 V 66 V 67 V 68 V 69 V 70 V 7 V 71 V 72 V 73
V 74 V 75 V 76 V 77 V 78 V 79 V 8 V 80 V 81 V 82 V 83 V 84 V 85 V 86 V 87 V 88 V 89 V 9 V 9
1 V 90 V 92 V 93 V 95 V 94 V 96 V 97 V 98 V 99 V

Optimized cost is 822164.625

Spend 3447.280 s

[107501019@eda359_forclass ~/hw4]$ ./PA4_checker.out t300.txt t300test.txt
Expression is correct!!
Module size is correct.
Overlapping check is correct.
All correct! Will done.
```

實際 output file 的輸出因為太長就沒有貼了，所以只在這裡貼出執行時間及驗證結果。

VI. Reference

陳聿廣教授上課講義