Part 2 Project Report:

Minimum-Cut Placement Algorithm

Introduction

In integrated circuit design and hardware implementation, there are many developed algorithms that deals with netlist/design components placement onto the physical chip. Placement algorithms can be implemented by many different techniques such as Partitioning-based algorithms, analytic techniques and stochastic algorithms.

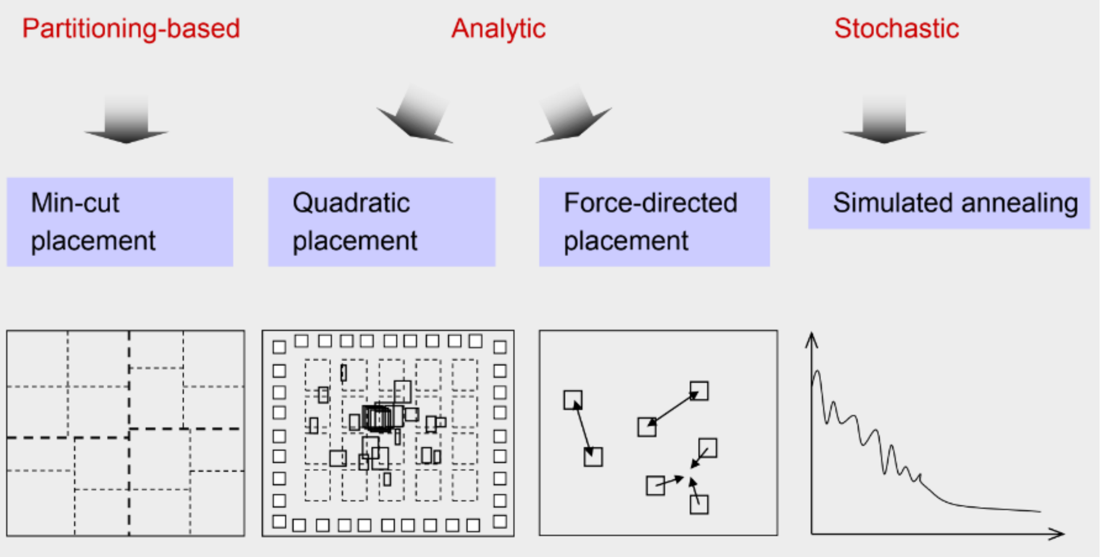


Figure 1 Partitioning Schemes

We are demonstrating the Min-cut placement algorithm which is a partitioning-based algorithm that incorporates the KL Algorithm in each partitioning step.

Implementation:

- The header and program files are submitted with commentary explaining all functions used
- The Min-Cut placement function is generic and is not specific to some kind or number of netlist, it can do as many cuts up to 15 (4h).
- There is an extra work done to optimize the estimated wire length but it is specific only to 2x4 grids.
- The results below shows the placement of the netlist file named: *netlist_small_no_hypernets.txt*

Algorithm

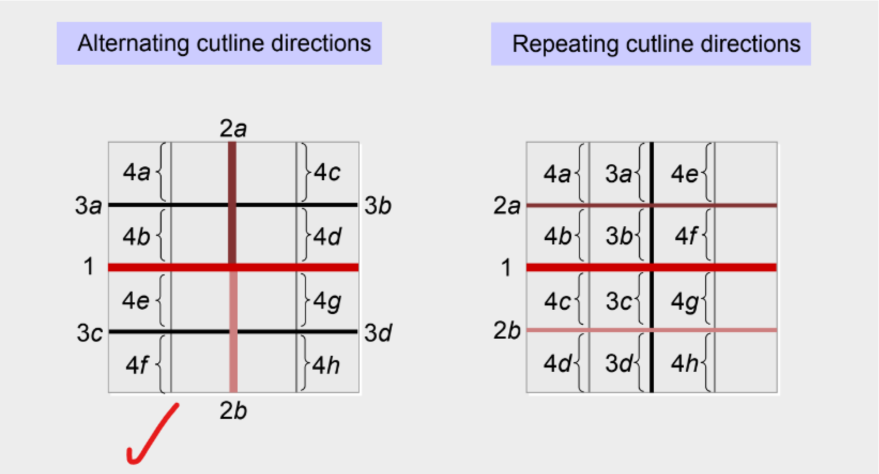


Figure 2 We use Alternating cutline direction

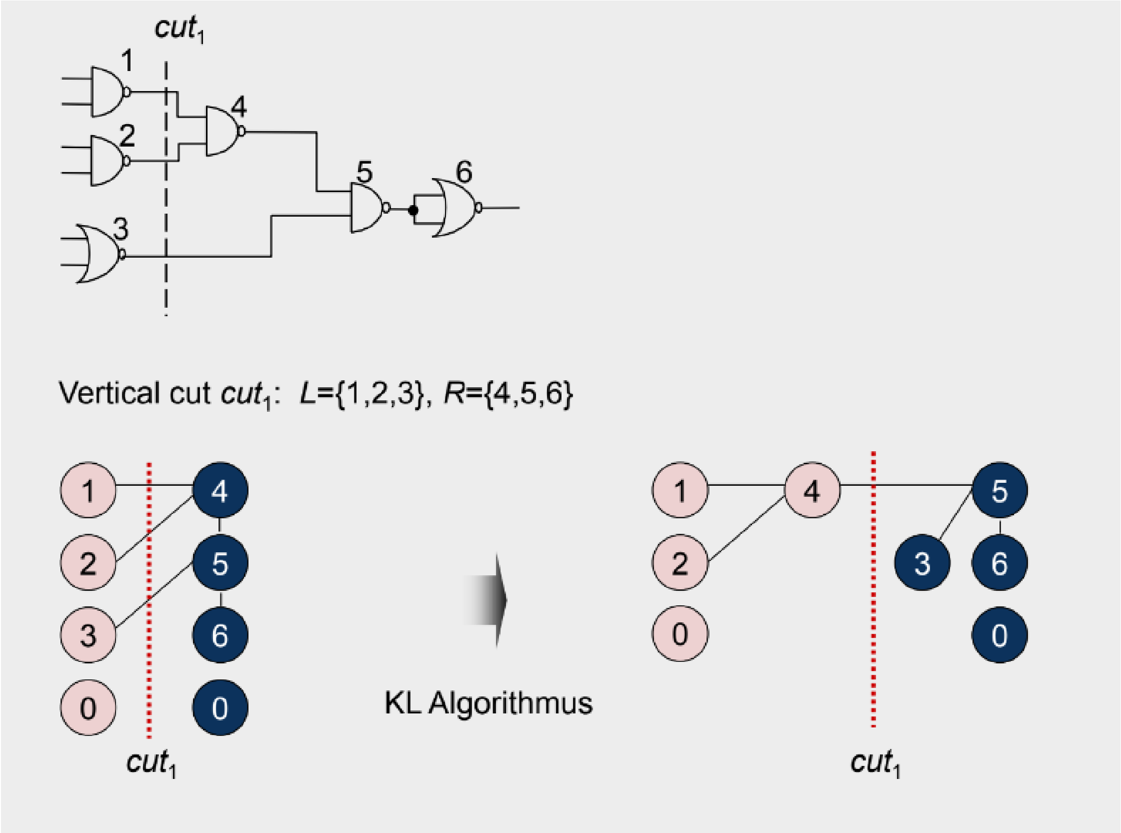


Figure 3 Step1

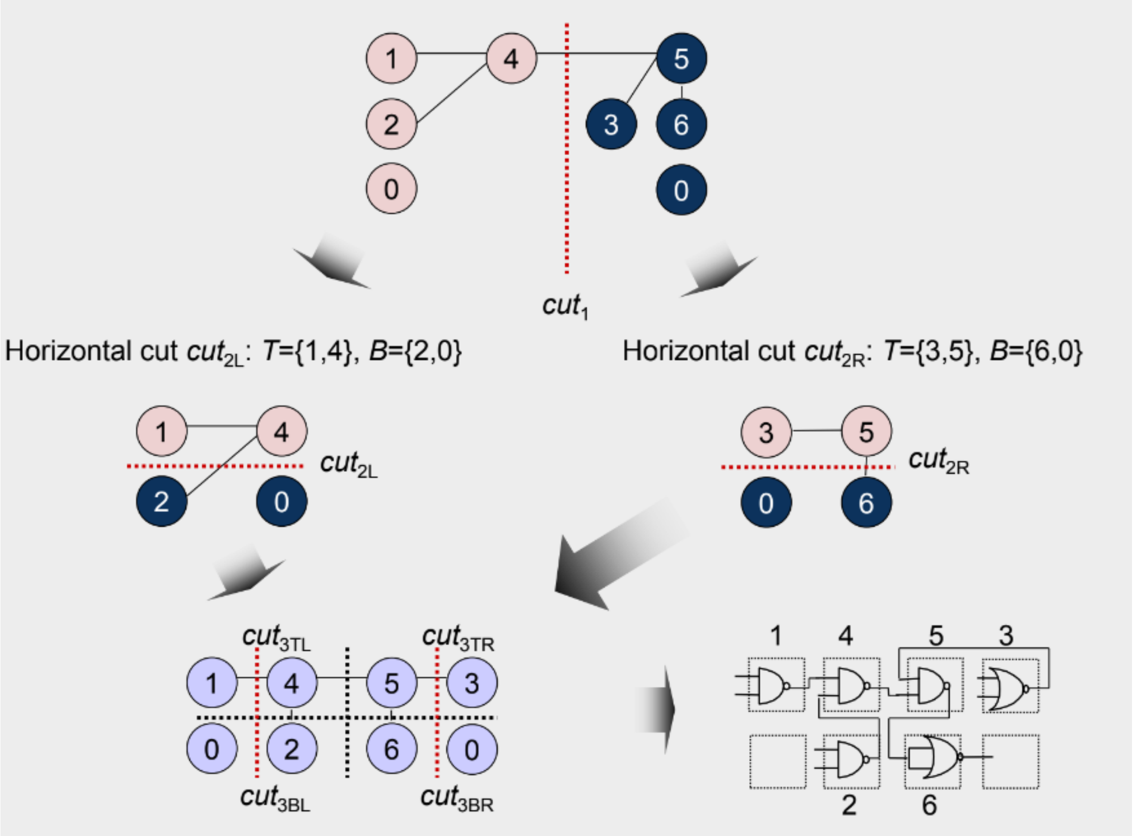
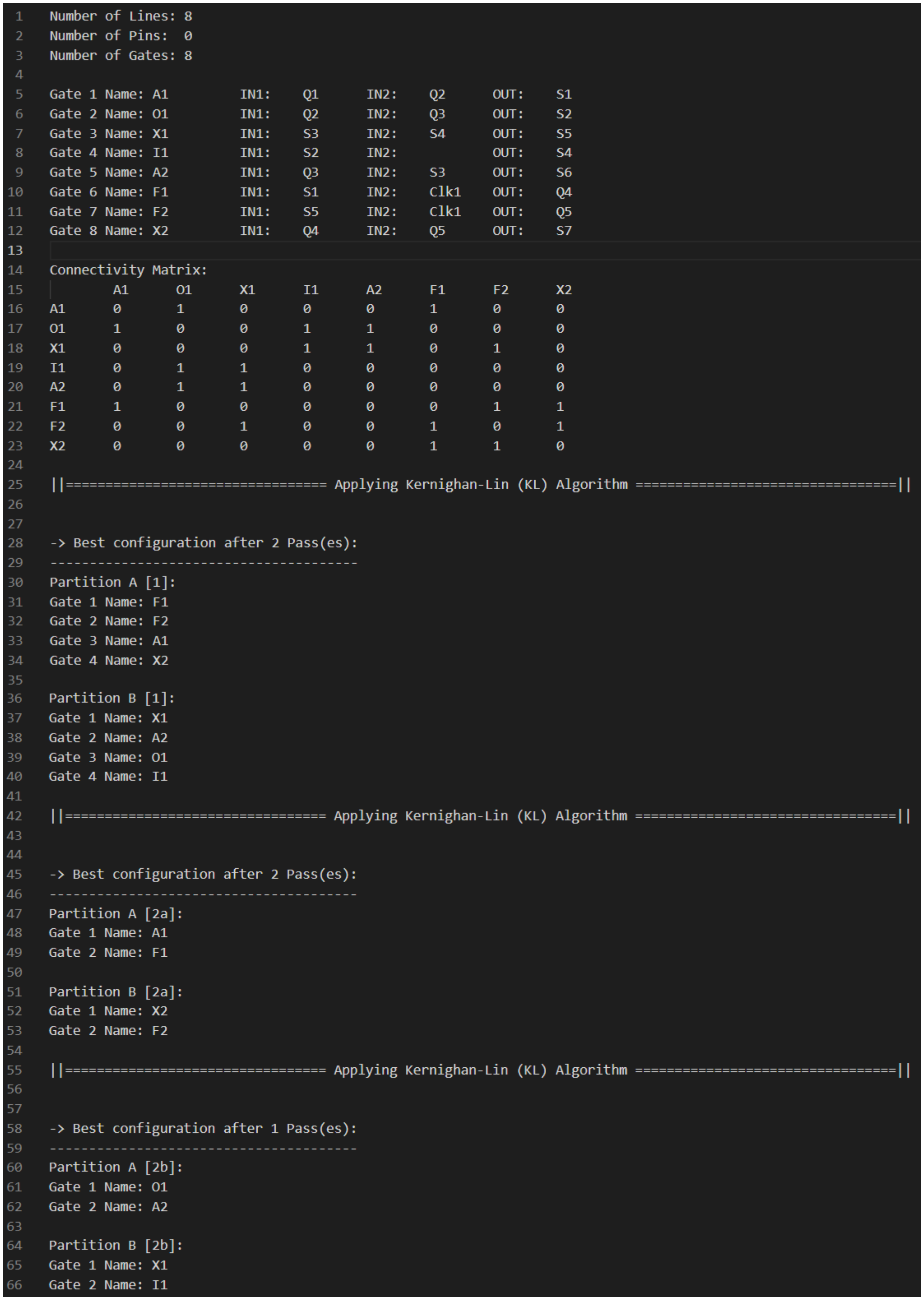
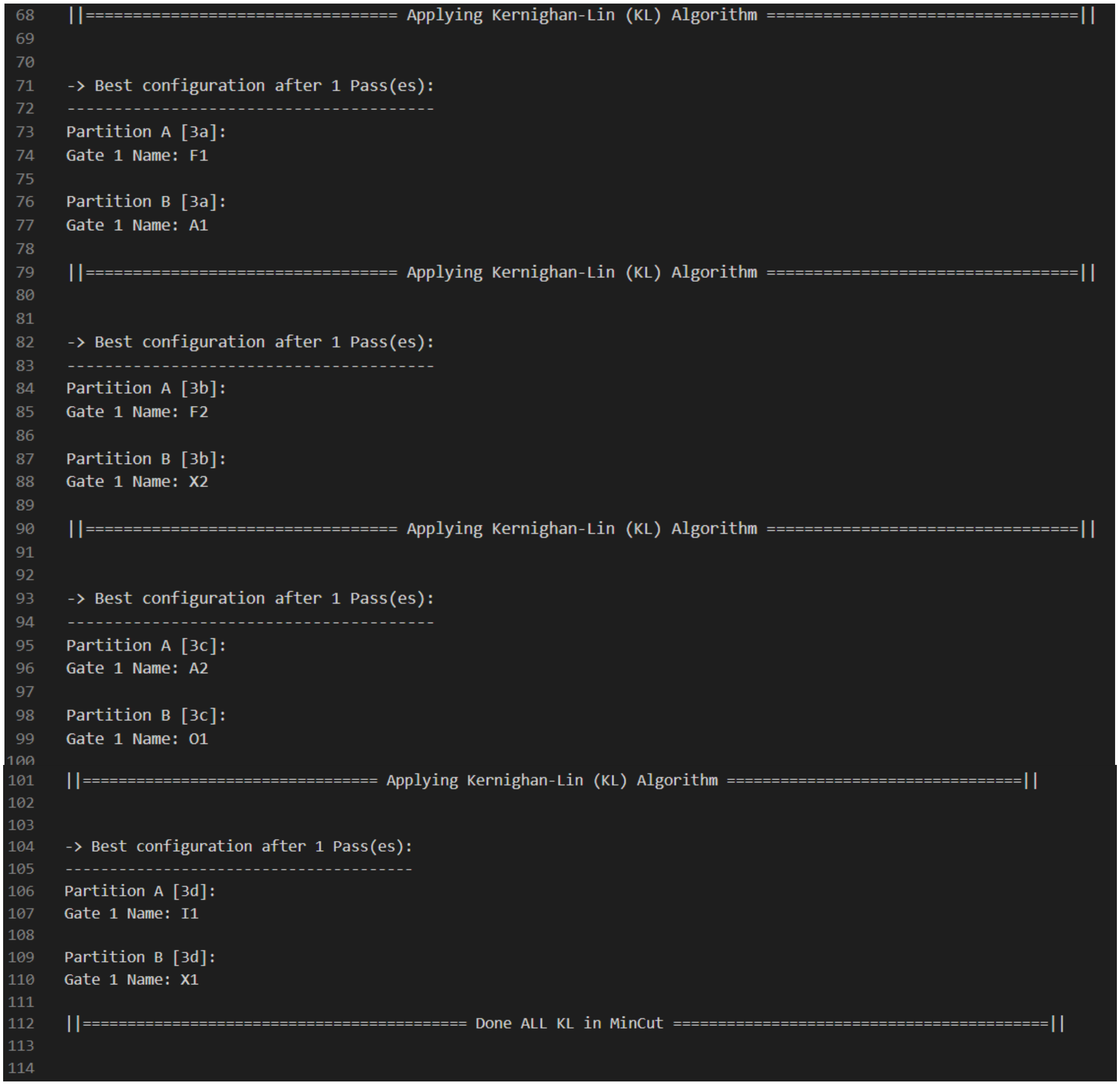


Figure 4 Step2

Results

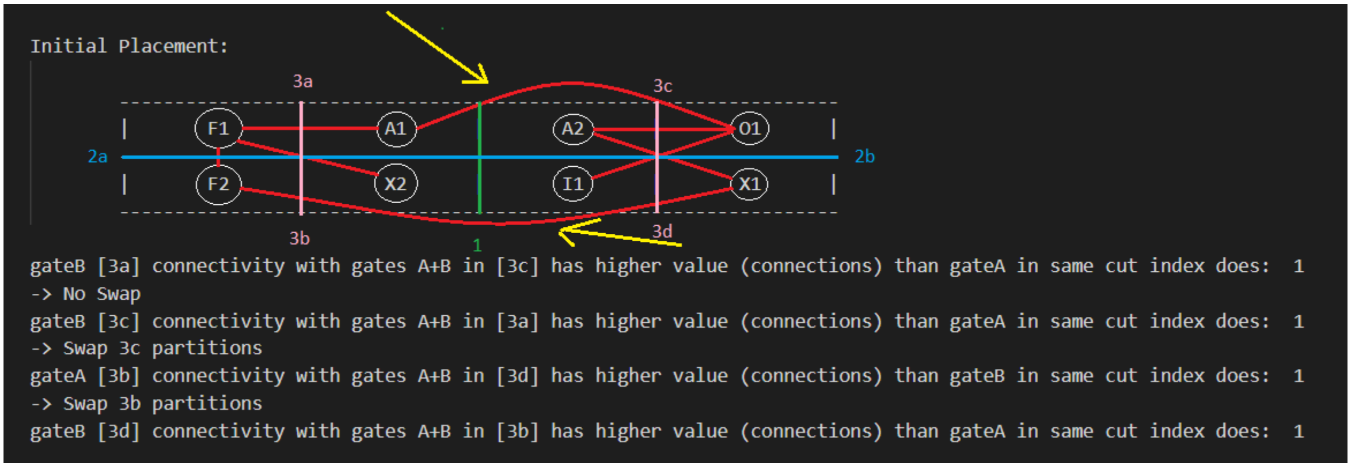


```
Number of Lines: 8
Number of Pins:  0
Number of Gates: 8

Gate 1 Name: A1        IN1:    Q1      IN2:    Q2      OUT:    S1
Gate 2 Name: O1        IN1:    Q2      IN2:    Q3      OUT:    S2
Gate 3 Name: X1        IN1:    S3      IN2:    S4      OUT:    S5
Gate 4 Name: I1        IN1:    S2      IN2:            OUT:    S4
Gate 5 Name: A2        IN1:    Q3      IN2:    S3      OUT:    S6
Gate 6 Name: F1        IN1:    S1      IN2:    Clk1    OUT:    Q4
Gate 7 Name: F2        IN1:    S5      IN2:    Clk1    OUT:    Q5
Gate 8 Name: X2        IN1:    Q4      IN2:    Q5      OUT:    S7


Connectivity Matrix:
        A1      O1      X1      I1      A2      F1      F2      X2
A1      0       1       0       0       0       1       0       0
O1      1       0       0       1       1       0       0       0
X1      0       0       0       1       1       0       1       0
I1      0       1       1       0       0       0       0       0
A2      0       1       1       0       0       0       0       0
F1      1       0       0       0       0       0       1       1
F2      0       0       1       0       0       1       0       1
X2      0       0       0       0       0       1       1       0

||================================= Applying Kernighan-Lin (KL) Algorithm =================================||


-> Best configuration after 2 Pass(es):
---------------------------------------
Partition A [1]:
Gate 1 Name: F1
Gate 2 Name: F2
Gate 3 Name: A1
Gate 4 Name: X2

Partition B [1]:
Gate 1 Name: X1
Gate 2 Name: A2
Gate 3 Name: O1
Gate 4 Name: I1

||================================= Applying Kernighan-Lin (KL) Algorithm =================================||


-> Best configuration after 2 Pass(es):
---------------------------------------
Partition A [2a]:
Gate 1 Name: A1
Gate 2 Name: F1

Partition B [2a]:
Gate 1 Name: X2
Gate 2 Name: F2

||================================= Applying Kernighan-Lin (KL) Algorithm =================================||


-> Best configuration after 1 Pass(es):
---------------------------------------
Partition A [2b]:
Gate 1 Name: O1
Gate 2 Name: A2

Partition B [2b]:
Gate 1 Name: X1
Gate 2 Name: I1
```



```
||================================ Applying Kernighan-Lin (KL) Algorithm ================================||


-> Best configuration after 1 Pass(es):
--------------------------------------
Partition A [3a]:
Gate 1 Name: F1

Partition B [3a]:
Gate 1 Name: A1

||================================ Applying Kernighan-Lin (KL) Algorithm ================================||


-> Best configuration after 1 Pass(es):
--------------------------------------
Partition A [3b]:
Gate 1 Name: F2

Partition B [3b]:
Gate 1 Name: X2

||================================ Applying Kernighan-Lin (KL) Algorithm ================================||


-> Best configuration after 1 Pass(es):
--------------------------------------
Partition A [3c]:
Gate 1 Name: A2

Partition B [3c]:
Gate 1 Name: O1

||================================ Applying Kernighan-Lin (KL) Algorithm ================================||


-> Best configuration after 1 Pass(es):
--------------------------------------
Partition A [3d]:
Gate 1 Name: I1

Partition B [3d]:
Gate 1 Name: X1

||======================================== Done ALL KL in MinCut ========================================||
```

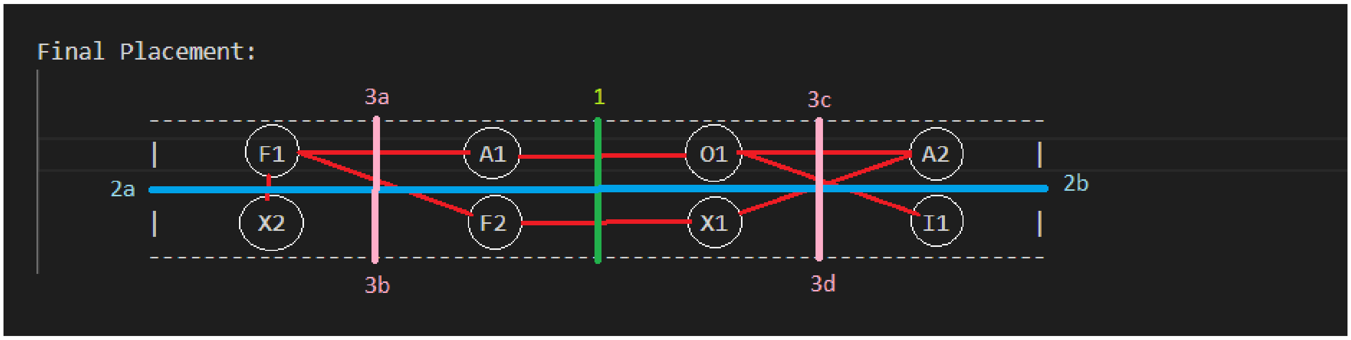
Figure 5 Min Cut results



```
Initial Placement:
```

```
gateB [3a] connectivity with gates A+B in [3c] has higher value (connections) than gateA in same cut index does:  1
-> No Swap
gateB [3c] connectivity with gates A+B in [3a] has higher value (connections) than gateA in same cut index does:  1
-> Swap 3c partitions
gateA [3b] connectivity with gates A+B in [3d] has higher value (connections) than gateB in same cut index does:  1
-> Swap 3b partitions
gateB [3d] connectivity with gates A+B in [3b] has higher value (connections) than gateA in same cut index does:  1
```

Figure 7 Initial placement



```
Final Placement:
```

Figure 6 Final refinement