**EDA HW4**

江若綾 113062640

1. **How to compile and execute program**

**Compile program:**

Step 1: Enter "HW4/src/"

Step 2: Enter command: $ make

**Remove compile program:**

Step 1: Enter "HW4/src/"

Step 2: Enter command: $ make clean

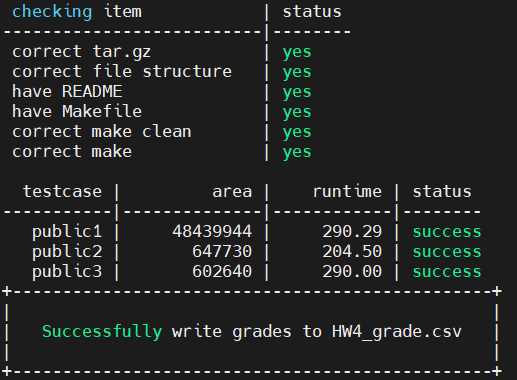
**Execute program:**

Step 1: Enter "HW4/bin/"

Step 2: Enter command: $ ./hw4 <.txt file> <.out file>

Execution example: $ ./hw4 ../testcase/public1.txt ../output/public1.out

1. **Area and runtime of each testcase**

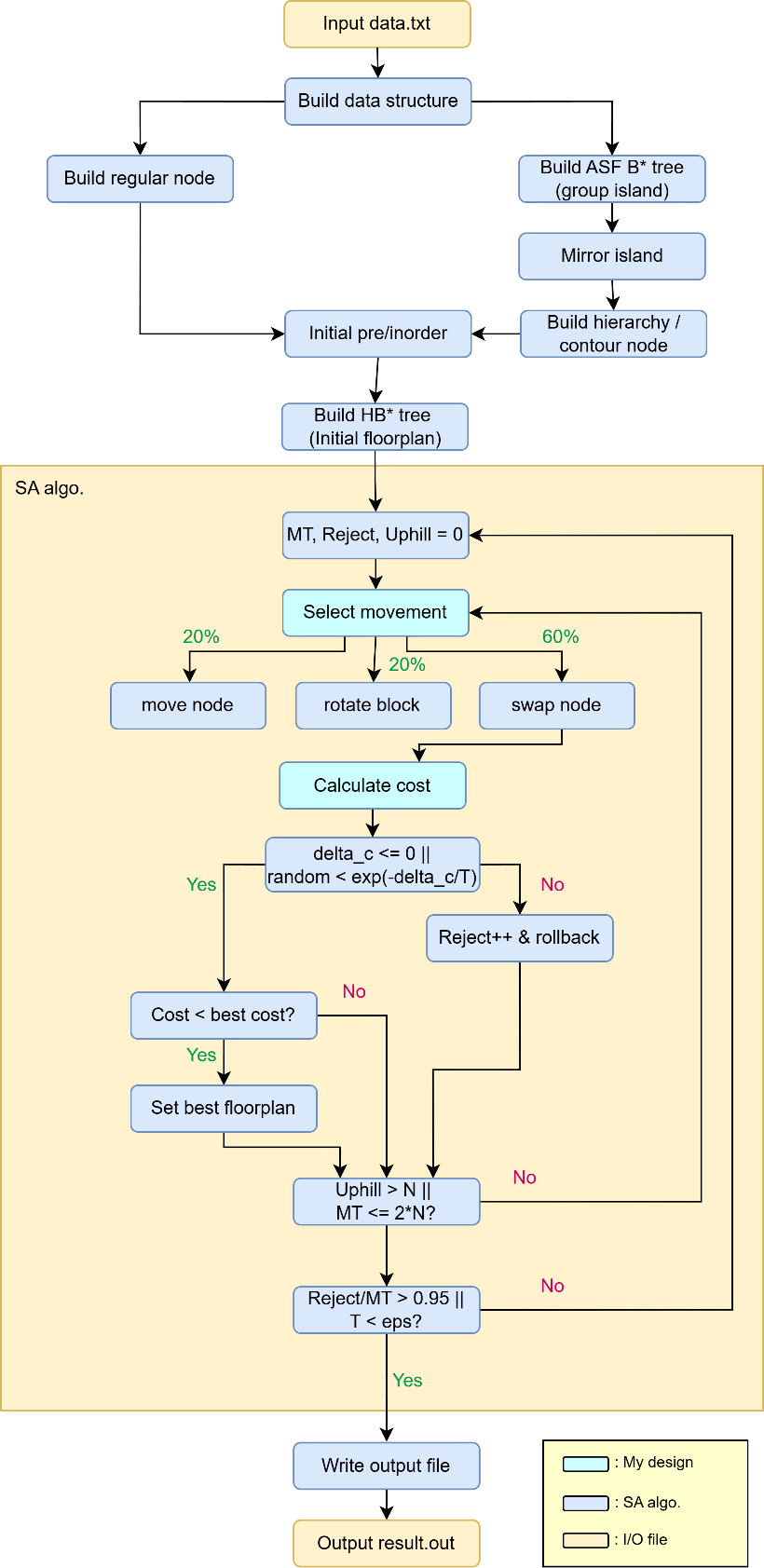


|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **public1** | **public2** | **public3** |
| **Area** | 48439944 | 647730 | 602640 |
| **Run time** | 290.29 | 204.50 | 290.00 |

**[Floor plan result]**

|  |  |  |
| --- | --- | --- |
| **Public1** | **Public2** | **Public3** |
|  |  |  |

1. **Algorithm Implementation**

****

My design is similar to **Simulated\_Annealing\_floorplanning (SA) algorithm**[[1]](#footnote-1) **and Analog Placement Based on Symmetry-Island Formulation**[[2]](#footnote-2). I modified some parts of the algorithm to speed up the execution time and make the final Area become lower.

**Technique to minimize the Area:**

The final Area is heavily affected by its perturbation during SA. Since perturbation is important to the final area, I implemented three of the major perturbations in the original paper of **Analog Placement Based on Symmetry-Island Formulation**.

1. **M1 perturbations (move node)**

In this perturbation, I will choose one regular node(A) and another one node(B) from either regular node or hierarchy node. I will then attach node A to node B, which makes B the parent of A. I will randomly choose to make node A the right or left child of node B. But if node B is a hierarchy node, node A will only be the left child of node B. If B has left(right) child before A attaches to its left(right), the left(right) child of B will be the A’s left(right) child after the attachment.

1. **M2 perturbations (rotate block)**

In this perturbation, I will choose one node from either a regular node or a hierarchy node. If it’s a regular node, I will rotate its block in which its width and height will be swapped. If it’s a hierarchy node, I will then pick a block inside the hierarchy node. If the block is not self-symmetric, I will again swap its width and height. If the block is self-symmetric and its vertical (horizontal) symmetric, I will swap its current width and height (half block) and multiply(divide) its new width by 2 and divide(multiply) its new height by 2 to match the correct self-symmetric node. And pack the corresponding ASF B\*tree and the HB\* tree.

1. **M3 perturbations (swap node)**

In this perturbation, I will choose one node from either a regular node or a hierarchy node. If it’s a regular node, I will then choose another regular node and swap their position inside the HB\* tree. If it’s a hierarchy node, I will then choose two nodes from the corresponding ASF B\*tree, and swap their positions inside the ASF B\*tree. And pack the corresponding ASF B\*tree and the HB\* tree.

1. **Floorplan initialization**

The initial floorplan initialization can deeply affect the area of the final results

I make the initial HB\* tree left skew if it starts with vertical symmetric for all the symmetric group and right skew otherwise. If the initial floorplan is far sway from the real optimal floorplan, it might have a better chance to find better solutions compared to the initial floorplan that is already inside the local optima.

1. **Trick to speed up my program or enhance solution quality**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **[Original] (only M3)** | **[Trick 1] (M3 + M2)** | **[Trick 2] (M3 + M2 +M1)**  **[Final result]** |
| Floorplan |  |  |  |
| Area | 1572636 | 944694 | 602640 |

At first, I follow the original SA algorithm’s method and implement M3 movements into my algorithm.

**[Original] Use M3 movements to modify the HB\* tree**

I soon found out that M3 is not enough for the SA to find the best solution for the floorplan. Simply swapping node is not enough, especially if my initial floorplan is left or right skew. No matter how I perturb, the result remains bad. So, I try to implement another perturbation.

**[Trick 1] Use M2 and M3 movements to modify the HB\*tree**

As the above result shows, we can see that the M3 movement is not quite enough to efficiently lower the area. The reason might be the inside of the M3 movement. M3 movement only allows nodes to swap. In which can be too insignificant. To enhance the solution quality, I try implementing M2 perturbation in which to rotate the blocks. But this perturbation is still not good enough changes for the HB\*tree to transform it into a better solution. So, I started to implement perturbation in which can move node around.

**[Trick 2] Use M1 and M2 and M3 movements to modify the HB\*tree**

As the above result shows, we can see that after the adjustment, the area becomes lower.

1. **Learned & encountered problems**

**Learned:**

In this course work, I’ve encountered a lot of problems related to node manipulation inside either B\*tree or HB\*tree. In order to move nodes, I must prevent nodes from looping and keep the data structure correct for every perturbation. This course work is almost like forcing me to review all my experience about tree. I also learned how to use GDB to debug when having a segmentation fault or other error.

**Encounter problems:**

Because of the lack of consideration about the data structure. I step on my own foot when implementing perturbation. I made two types of nodes for ASF B\*tree node and HB\*tree node, this can be good if I know how to keep the data only in one place. But I carelessly copy multiple data inside all the places I can store in which make the real data very hard to find later. Somehow I successfully find the correct data but it is still a lesson to remind myself to not do it again.

1. D. F. Wong and C. L. Liu, "A New Algorithm for Floorplan Design," *23rd ACM/IEEE Design Automation Conference*, Las Vegas, NV, USA, 1986, pp. 101-107, doi: 10.1109/DAC.1986.1586075.

   2 P. -H. Lin, Y. -W. Chang and S. -C. Lin, "Analog Placement Based on Symmetry-Island Formulation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 6, pp. 791-804, June 2009, doi: 10.1109/TCAD.2009.2017433. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)