مدينة زويل للعلوم والتكنولوجيا

Zewail City of Science and Technology

Nano Technology Engineering Program

Spring 2019

Physical Design & EDA Algorithms

NANENG 503

Final Project report

Team Members:

Mohamed Faris	201600657
Eslam Attia	201600661
Islam M.Adam	201601026

Supervised by

Dr.Amr Helmy

Eng. Mostafa Hafez

Eng. Islam Elsadek

- # Updated from Mini project 1:

  The nets names were taken to start only by “N”. Updated to handle any name for the net with an infinite number of inputs.

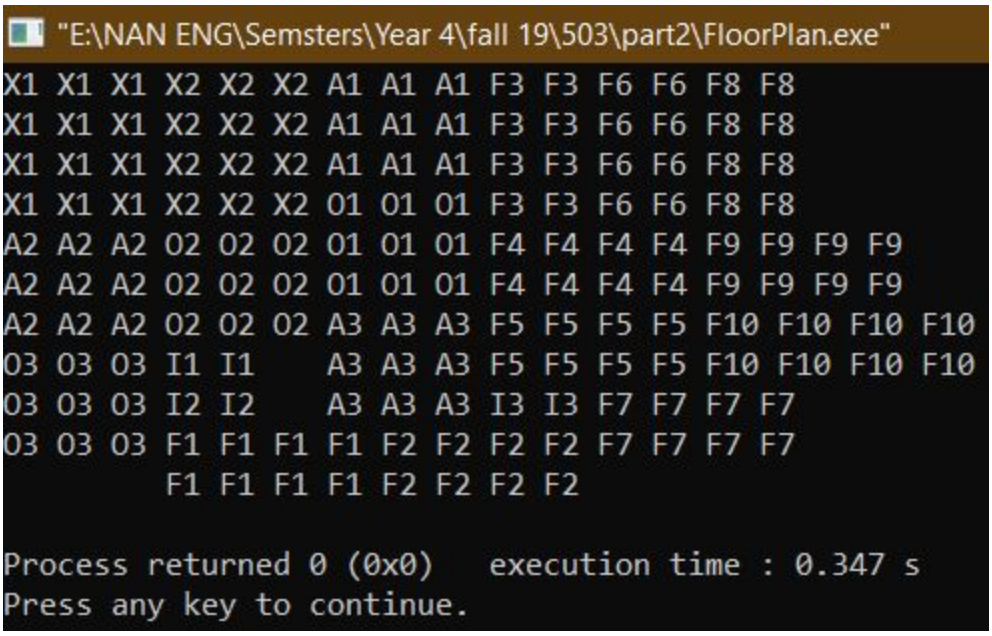
- # Floorplanning:
  - Algorithm:

    Our algorithm is based on cluster growth algorithm. Cluster growth algorithm starts by placing block by block with respect to aspect ratio. So, to achieve this algorithm we followed the following steps:

    1. We started by generating a big Matrix of strings to place all blocks in it its initial value of all cells is “0”.
    2. Constructing a string containing block type, name, no. of inputs, area, length and width.
    3. Sorting the blocks from the biggest to smallest to be able to generate a better floorplan by placing the smallest to biggest objects.
    4. Placing the first block manually.
    5. Two for loops to place each block
    6. In these for loops, we place each block by its four possible ways.
    7. Placing was done vertically and horizontally in nearest point able to have that block.
    8. The block is placed vertically and horizontally in the matrix and the block itself is placed vertically and horizontally in each direction, so we have four cases.
    9. We start to check for the four cases which have the minimum area and best aspect ratio.
    10. The minimum in the area and an aspect ratio close to unity is used for the next block plot.
    11. The output is plotted in the matrix for each block if it is 4*2 it is plotted as that

        X X X X
        X X X X

- Output:



- struct block

  This struct is used to get the name of each block and its length and width. It is used in floorplan function.

- Functions used:
  - void getSize(string currFloor[n][n], float *floorL, float *floorW);

    This function is used to indicate the length and width of the floor plan to calculate the AR and area of the floorplan. It takes the matrix and indicates the maximum point in X and y != "0" and choose it to be maximum length and in y to be maximum width.

  - void reset(string currFloor[n][n], string prevFloor[n][n]);

    As we have four cases for placing the block so we have four currFloor matrices in our algorithm we place the current floor to equal previous floor state and vice versa when we decide which case is chosen. So the main functionality of this function is to return the needed matrix to the other one.

- void floorPlan(string currFloor[n][n], string prevFloor[n][n], string netlist, int n);

  This is the main function that other functions operate inside it.

  It starts by mapping the netlist and get the name, type and n then by knowing n and type we get the length and the width for each cell and place it in its string. Then it sorts them from the bigger to smaller. Then it places the biggest block first and then iterates to put the rest of the blocks.

  It takes the previous string from the previous iteration and then add the block vertically and take this case as the optimum solution and then check for the rest of the cases if it is more optimum or not by checking for area and AR. After that, it returns the currFloor to be the prevFloor to start the new iteration. Then at the end, it plots the results as in the figure above.

- void placing (string matrix[n][n], string ori, int lentgth, int width,string name);

  We used this function to place each block in the four cases mentioned above in the algorithm. This function starts by placing the matrix to add the block to it and choose whether work in V or H. We start by indicating all available points and then choose the lowest point so that we can get the minimum area. Then, by using two for loops from that point to that point + width and from the reference point I to I + length.

  The chosen point is taken by indicating that it is the lowest point and it can take this block without overlapping by the bool availableL. If true, we take this point as reference(m, min_point)

- int areaCalc (string word, int n, int *length, int *width) ;

  This function is used to get the length and width of each block by knowing the number of inputs for each cell.

Note: we couldn't implement the three codes in one file as we used a lot of variables with the same name in the three codes.