

ECE 1387 - CAD for Digital Circuit Synthesis and Layout
Assignment #1 – FPGA Maze Router – Logic Density and Impact on Routability

September 2023

J. Anderson

Assignment Date: September 22

Due Date: October 6 (by 23:59 Toronto time via electronic submission)

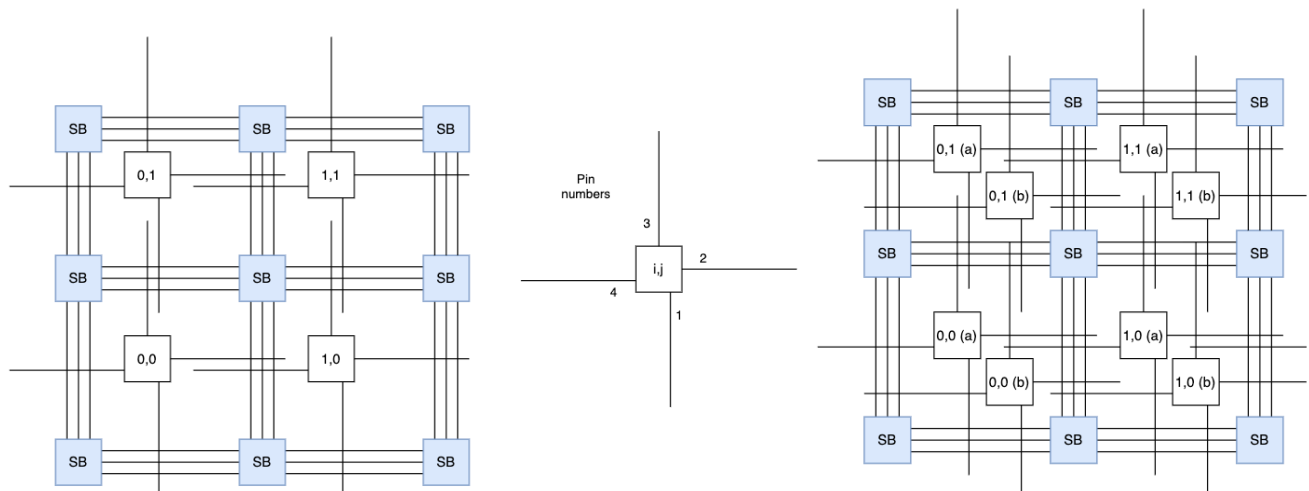
Late Penalty: -2 marks per day late, with total marks available = 20

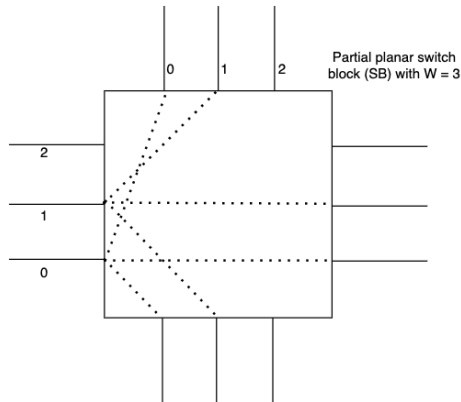
You are to write an implementation of the FPGA maze router described in class, and study the impact of logic-block architectures on routability. You must have your program display the routing with graphics. There are two options for the graphics: 1) A graphics package is provided on the course web page (courtesy of Prof. V. Betz). 2) You can have your router output a text file representation of the graphics display compatible with the NEATO format (<https://www.graphviz.org/pdf/neatoguide.pdf>). The graphics can then be drawn (using your text file) in a web browser using <https://dreampuf.github.io/GraphvizOnline>. You are to develop your router so that it executes on Linux and you will hand it in electronically on the ECF network (i.e. your router must compile/execute on ECF machines).

You should use an FPGA architecture *similar* to that described in class, illustrated in figures below. The middle figure shows the pin numbering scheme. Notice that there are **four** pins per logic block (one on each side of the block), labelled as pins 1,2,3 and 4. Note the side of the logic block on which each pin resides – your implementation must follow this. Each pin can connect to **all** tracks in the neighbouring channel ($F_c = W$). The routing segments span one logic block tile. Routing switches are *bidirectional*. The switch block is planar, where each wire segment attached to the block can connect to 3 other wire segments (on the other three sides of the block). A figure illustrating the switch block is on the next page. You are to investigate **two** FPGA architectures and study the impact of architecture on routability:

1) One logic block at each x,y position. This corresponds to the left figure below. This is referred to as the **baseline** architecture. The x,y position is shown inside each logic block.

2) Two logic blocks at each x,y position – i.e., double the *logic density*. This corresponds to the right figure below. The two logic blocks are labelled (a) and (b). This is referred to as the **dense** architecture.





For the baseline architecture, your program should take input from a file that has the following format:

The first line consists of one integer, n , where n gives the $n \times n$ dimensions of the chip in logic blocks. The grid cells are numbered from 0 to $n-1$ in each dimension.

The second line indicates the number of tracks per channel to use, W .

The next set of lines has the form “ $X1\ Y1\ P1\ X2\ Y2\ P2$ ”. Each of these lines gives a pair of pins to be connected. The first pin is attached to the block at location $X1, Y1$, and uses pin number $P1$. The second pin is specified in the same manner by $X2, Y2$ and $P2$. $P1$ is thus the **source** pin (the driver); $P2$ is the **sink** pin (the load). This list is terminated by the line: -1 -1 -1 -1 -1 -1. A source pin may have *multiple* load pins; however, each load is driven by at most one source pin. Your router may share wiring among the loads driven by a source pin.

Example input file:

| | |
|-------------------|--|
| 10 | (10 x 10) grid |
| 4 | (4 tracks per channel ($W = 4$)) |
| 1 2 4 2 3 2 | Pin 4 on block at (1,2) connects to pin 2 at (2,3) |
| 0 0 4 1 2 3 | Pin 4 on block at (0,0) connects to pin 3 at (1,2) |
| -1 -1 -1 -1 -1 -1 | (end of pin pair list) |

For the dense architecture, the file format is slightly adjusted to reflect the two logic blocks at each x, y position:

Example input file:

| | |
|-------------------|--|
| 10 | (10 x 10) grid |
| 4 | (4 tracks per channel ($W = 4$)) |
| 1 2 a 4 2 3 b 2 | Pin 4 on block at (1,2)(a) connects to pin 2 at (2,3)(b) |
| 0 0 b 4 1 2 a 3 | Pin 4 on block at (0,0)(b) connects to pin 3 at (1,2)(a) |
| -1 -1 -1 -1 -1 -1 | (end of pin pair list) |

Your program must be able to display the routing solution for all connections in the test file using graphics. For debugging purposes, you may find it helpful to write your program so that it can display the progress of your algorithm as it routes each step for each connection; that is, you may wish to display each step of the router maze expansion (though this is not mandatory for this assignment). You should test your program on the following test files located on the course Piazza web page. There are two versions of each test file: one for the baseline architecture; one for the dense architecture.

cct1, cct2, cct3, cct4

Note that in addition to routing each circuit with the value of W given in the input file, you will need to find the smallest value of W for which the test circuits will route successfully, for the both the baseline and dense architectures.

What to do and what to hand in?

1. Submit your source code, executable and PDF report on the ECF network. Details on how to do this will be announced on the course Piazza website.
 2. A paper plot of the results from the four test files (using the value of W given in the test files) for the two architectures (baseline and dense). The total number of routing segments used to route a design is a key metric in routing, as it is closely tied to performance and power consumption. Report the total number of routing segments used for each test file, for both architectures.
 3. *Report two items for each circuit, in table format, for both architectures (baseline and dense):* the smallest number of tracks/channel (W) that your program could successfully route each test circuit in; and, the total number of used routing wire segments (when W is minimum). That is, your program should be capable of varying the number of tracks per channel, and you are required to find the smallest number of tracks per channel that your program will successfully route the circuits in. Innovate to reduce W and minimize the number of routing segments used. Explain any optimizations you applied that were successful.
 4. In one paragraph, discuss the impact of the logic block architectures on W and the number of used routing segments.
 5. Hand in a two-page description of the flow of your software, describing the major routines and data structures, and how they interact. Where you were faced with choices in your implementation of the algorithm, indicate what choices you made, and why.
- In class, I will report on the minimum W and # of segments achieved for each test file, for the two logic block architectures.