University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

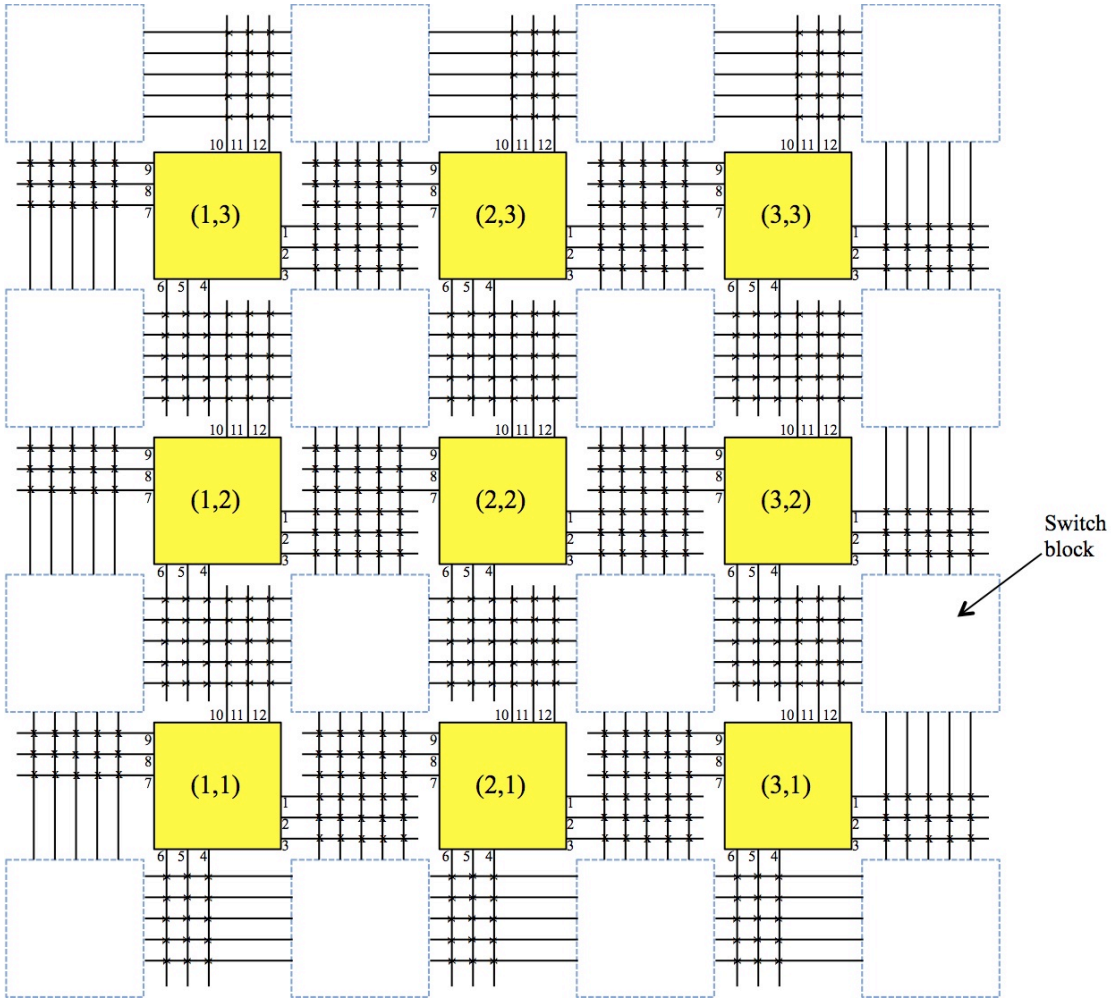
ECE 1387 - CAD for Digital Circuit Synthesis and Layout
Assignment #1 – FPGA Maze Router, Considering Input and Output Pin Equivalence

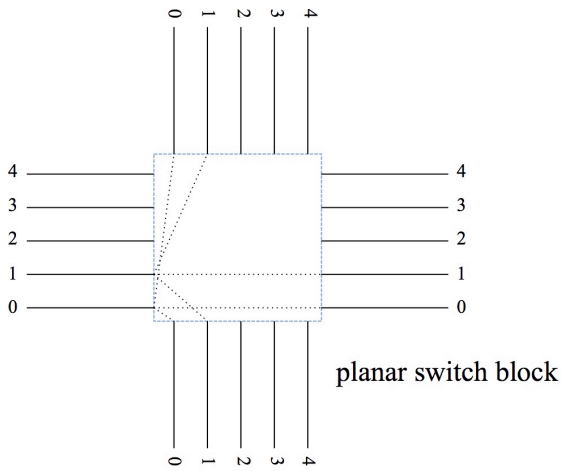
September 2013 J. Anderson

Assignment Date: September 20
Due Date: October 4 (before lecture begins)
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 input and output pin "equivalence" on circuit routability and wirelength (described below). You must have your program display its progress with graphics. A graphics packages is provided on the course web page. You are to develop your router using Linux (on EECG or ECF).

You should use an FPGA architecture *similar* to that described in class, as illustrated in the figure below. The figure shows the pin numbering scheme, and the x,y logic block positioning scheme. Notice that there are **twelve** pins per logic block (three on each side of the block). Pins 1, 4, 7 and 10 are **output** pins. Pins 2, 3, 5, 6, 8, 9, 11 and 12 are **input** pins. Each pin can connect to **all** tracks in the neighbouring channel (F_c = W). Each routing segment endpoint can connect to **three** other segments (F_s = 3). The routing segments span one logic block tile. Use the "planar" switch block, as described in class (and shown below). In the figure of a switch block below, for clarity, not all switches are shown. Make sure that you understand how the complete switch block looks, for any value of W.





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 x n dimensions of the chip in logic blocks. The grid cells are numbered from 1 to n 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, where P1 = 1,4,7, or 10. The second pin is specified in the same manner by X2,Y2 and P2. P2 = 2,3,5,6,8,9,11, or 12. 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 – these are listed on adjacent lines of the file; 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 1 4 4 5	Pin 1 on block at (1,2) connects to pin 5 at (4,4)
3 3 4 8 9 3	Pin 4 on block at (3,3) connects to pin 3 at (8,9)
-1 -1 -1 -1 -1 -1	(end of pin pair list)

Your program must be able to display the routing solution for all of the connections in the test file using the graphics display. 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 expansion (though this is not mandatory for this assignment). You should test your program on the following test files located on the course web page:

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.

What to do and what to hand in?

1. The location of the executable file and source code (on EECG or ECF), with instructions on how to run the executable. **Please** set the permissions so that I can run it and view your source.

2. A paper plot of the results from the four test files (using the value of W given in the test files). The graphics package allows you to do this. 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.

3. Report two items in a table: the smallest number of tracks/channel (W) that your program could successfully route each test circuits in, and the total number of used routing wire segments. 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. One idea is to try different connection ordering schemes to reduce W. Explain any optimizations you applied that were successful. Report the total number of routing segments used for each test file when W is minimum for that test file.

4. A PhD student in my group has recently been looking into the impact of pin equivalence on routability (i.e. the ease with which a circuit can be routed) and also route quality (e.g. speed performance and wirelength). Input-pin equivalence means that the input pins on a logic block are "swappable". With input-pin equivalence, the router has the freedom to route to *any **input*** pin on a load logic block and it is not restricted by the P2 listed in the input file. Input-pin equivalence can be made possible through internal connectivity within a logic block. Modify your router to support input-pin equivalence and repeat step #3 above. Discuss the results. **Note:** be sure that a unique input pin is used for each signal routed to a logic block – "shorts" between signals on one input pin are not allowed!

5. Out-pin equivalence means that output pins are "swappable" and the router is free to route from *any output* pin on a logic block (the router is not restricted by the P1 listed in the input file). Modify your router to support output-pin equivalence and repeat step #3 above twice: with only output-pin equivalence and with both input-pin and output-pin equivalence. Discuss the results. Again, ensure that at most one signal is routed from each logic block output pin.

6. 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. Describe how you implemented pin equivalence, the different approaches you investigated and why you selected the one that you did.

In class, I will report on the minimum W achieved for each test file, as well as the different schemes proposed to handle pin equivalence.