

ECE 1387 - CAD for Digital Circuit Synthesis and Layout
Assignment #1 – FPGA Maze Router &
The Impact of Connection Block Flexibility (F_c)

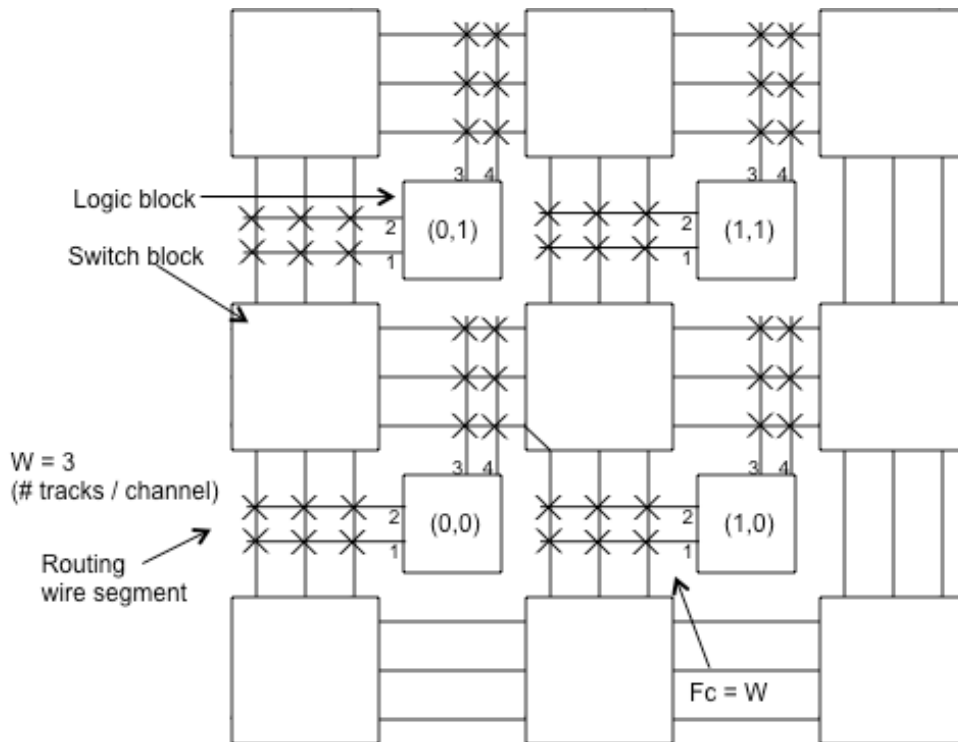
September 2016

J. Anderson

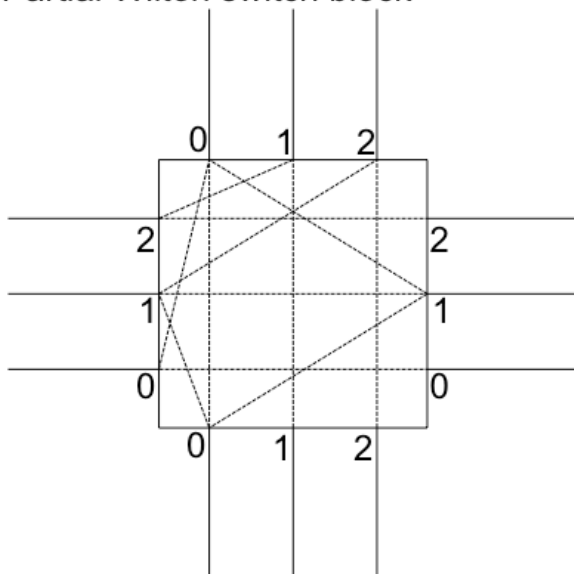
Assignment Date: September 23
Due Date: October 14 (before lecture begins)
Late Penalty: -2 marks per day late, with total marks available = 20

You are to write an implementation of the FPGA maze router described in class, and study one aspect of FPGA routing architecture: connection block flexibility. You must have your program display its progress with graphics. A graphics packages is provided on the course web page (courtesy of Prof. V. Betz). You are to develop your router using Linux and you will hand it in electronically on the ECF network (i.e. your router must compile/execute on ECF).

You should use the FPGA architecture illustrated in the figure below. The figure shows the pin numbering scheme and the x,y logic block positioning scheme. Notice that there are **four** pins per logic block (two on the left and two on the top), 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. In the first part of this assignment, each pin can connect to **all** tracks in the neighbouring channel ($F_c = W$, as shown in the figure). In the second part of this assignment, each pin can connect to $\frac{1}{2}$ of the tracks in the neighbouring channel ($F_c = W/2$). Each routing segment endpoint can connect to **three** other segments ($F_s = 3$). The routing segments span one logic block tile. Switches are bidirectional. Switch blocks use the Wilton topology. Make sure that you understand how the complete switch blocks look, for any value of W .



Partial Wilton switch block



Connectivity rules

North/South and East/West connections stay on same track index

Track i on West connects to track $W-i \bmod W$ on North

Track i on North connects to track $i+1 \bmod W$ on East

Track i on East connects to track $2W-2-i \bmod W$ on South

Track i on South connects to track $i+1 \bmod W$ on West

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)
    
```

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. You may assume W will be even.

What to do and what to hand in?

1. Submit your source code and executable on the ECF network. Details on how to do this will be announced on the course website.
2. A paper plot of the results from the four test files using the value of W given in the test files, and $F_c=W$. 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. Repeat #2 with $F_c=W/2$, where pins 1 and 3 may connect *only* to even track indices, and pins 2 and 4 may connect *only* to odd track indices.
4. Report two items for each circuit in table format for both F_c values: 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. Only consider even values for W .
5. In one paragraph, discuss the impact of $F_c=W$ and $F_c=W/2$ on W and the # of used routing segments.
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.

In class, I will report on the minimum W and # of segments achieved for each test file, for both F_c values.