EEL 5722C
Field-Programmable Gate Array Design

Lecture 9: CAD 3: FPGA Routing (Basic)*
www.eecs.ucf.edu/~mingjie/EEL5722

Prof. Mingjie Lin

* Some slides adopted from UMN EE5301 by Kia Bazargan & NWU EECS357 lectures
Overview

• Recap + Short intro. to the 2nd lab

• FPGA Routing
  – Recap: FPGA Routing Architecture
  – FPGA Routing Problem Formulation
    • Difference from conventional VLSI routing problem
  – Maze routing algorithm
  – Congestion-based negotiated routing algorithm (VPR)
FPGA Architecture - Layout

• Island FPGAs
  – Array of functional units
  – Horizontal and vertical routing channels connecting the functional units
  – Versatile switch boxes
  – Example: Xilinx, Altera

• Row-based FPGAs
  – Like standard cell design
  – Rows of logic blocks
  – Routing channels (fixed width) between rows of logic
  – Example: Actel FPGAs
FPGA Programmable Switch Elements

• Used in connecting:
  – The I/O of functional units to the wires
  – A horizontal wire to a vertical wire
  – Two wire segments to form a longer wire segment
FPGA Routing Channels Architecture

- Note: fixed channel widths (tracks)
- Should “predict” all possible connectivity requirements when designing the FPGA chip
- Channel -> track -> segment

  - Segment length?
    - Long: carry the signal longer, less “concatenation” switches, but might waste track
    - Short: local connections, slow for longer connections
FPGA Switch Boxes

- Ideally, provide switches for all possible connections

- Trade-off:
  - Too many switches:
    - Large area
    - Complex to program
  - Too few switches:
    - Cannot route signals

One possible solution

Xilinx 4000
VLSI Routing (NOT only FPGA)

• Problem
  – Given a placement, and a fixed number of metal layers, find a valid pattern of horizontal and vertical wires that connect the terminals of the nets
  – Levels of abstraction:
    • Global routing
    • Detailed routing

• Objectives
  – Cost components:
    • Area (channel width) – min congestion in prev levels helped
    • Wire delays – timing minimization in previous levels
    • Number of layers (fewer layers \(\rightarrow\) less expensive)
    • Additional cost components: number of bends, vias
Variations of Routing Problems

- **Full-custom:**
  - No constraint on routing regions

- **Standard cell:**
  - Variable channel height?
  - Feed-through cells connect channels

- **FPGA:**
  - Fixed channel height
  - Limited switchbox connections
  - Prefabricated wire segments have different weights

Figs. [©Sherwani]
FPGA Routing

• Routing resources pre-fabricated
  – 100% routability using existing channels
  – If fail to route all nets, redo placement

• FPGA architectural issues
  – Careful balance between number of logic blocks and routing resources (100% logic area utilization?)
  – Designing flexible switchboxes and channels (conflicts with high clock speeds)

• FPGA routing algorithms
  – Graph search algorithms
    • Convert the wire segments to graph nodes, and switch elements to edges
  – Bin packing heuristics (nets as objects, tracks as bins)
  – Combination of maze routing and graph search algorithms
Global vs. Detailed Routing

• Global routing
  – Input: detailed placement, with exact terminal locations
  – Determine “channel” (routing region) for each net
  – Objective: minimize area (congestion), and timing (approximate)

• Detailed routing
  – Input: channels and approximate routing from the global routing phase
  – Determine the exact route and layers for each net
  – Objective: valid routing, minimize area (congestion), meet timing constraints
  – Additional objectives: min via, power

Figs. [©Sherwani]
Maze Router

• Lee Algorithm

• Strengths
  – Guarantee to find connection between 2 terminals if it exists
  – Guarantee minimum path

• Weaknesses
  – Discussion mainly on single-layer routing
  – Requires large memory for dense layout
  – Slow

• Applications: global routing, detailed routing
  – VLSI, FPGA, PCB routing, …
Lee Algorithm

• Find a path from S to T by “wave propagation”

• Time & space complexity for an $M \times N$ grid: $O(MN)$ (huge!)
Connecting Multi-Terminal Nets

- Connecting Multi-Terminal Nets
  - Propagate wave from the source $s$ to the closest target
  - Mark ALL cells on the path as $s$
  - Propagate wave from ALL $s$ cells to the other cells
  - Continue until all cells are reached
  - Apply heuristics to further reduce the tree cost
Fast Maze Router

• Soukup, “Fast maze router,” DAC-78.
  • Combined breadth-first and depth-first search
    – Depth-first (line) search is first directed toward target T until an obstacle or T is reached
    – Breadth-first (Lee-type) search is used to “bubble” around an obstacle if an obstacle is reached.

• Time and space complexities: $O(MN)$, but 10–50 times faster than Lee’s algorithm.

• Find a path between S and T, but may not be the shortest!
Pathfinder negotiated congestion algorithm

• Initially routes each net by the shortest path, regardless of any overuse of wiring segments or logic block pins
• One iteration of the router consists of sequentially ripping-up and re-routing (by the lowest cost path found) every net in the circuit
• Cost of using a routing resource is a function of the current overuse of that resource and any overuse that occurred in prior routing iterations
• Gradually increasing the cost of oversubscribed routing resources, the algorithm forces nets with alternative routes to avoid using oversubscribed resources, leaving only the net that most needs a given resource behind.
VPR Improvement

(a) Expansion reaches a sink

(b) Traditional method: restart wavefront

(c) VPR method: maintain wavefront and expand around new wire
Verilog Tip of the Day: 3

- How to make your circuits run faster?
  - Example: FIR filter

```verilog
module fir(
  output [7:0] Y,
  input [7:0] A, B, C, X,
  input clk,
  input validsample);
  reg [7:0] X1, X2, Y;

  always @(posedge clk)
    if (validsample) begin
      X1 <= X;
      X2 <= X1;
      Y <= A* X + B* X1 + C* X2;
    end
  endmodule
```
Verilog Tip of the Day: 3 (cont.)
Verilog Tip of the Day: 3 (cont.)

module fir(
    output [7:0] Y,
    input [7:0] A, B, C, X,
    input clk,
    input validsample);
reg [7:0] X1, X2, Y;
reg [7:0] prod1, prod2, prod3;

always @ (posedge clk) begin
    if (validsample) begin
        X1 <= X;
        X2 <= X1;
        prod1 <= A * X;
        prod2 <= B * X1;
        prod3 <= C * X2;
    end
    Y <= prod1 + prod2 + prod3;
end
endmodule
Verilog Tip of the Day: 3 (cont.)

Adding register layers improves timing by dividing the critical path into two paths of smaller delay.
Final issues

• Good suggestions:
  – HDL/Verilog tip
  – Optional exercises for study

• Come by my office hours (right after class)

• Any questions or concerns?
Multiple Terminal Nets: Steiner Tree

- **Steiner tree (aka Rectilinear Steiner Tree – RST):**
  - A tree connecting multiple terminals
    - Original points: “demand points” – set D
    - Added points: “Steiner points” – set S
  - Edges horizontal or vertical only

- **Steiner Minimum Tree (SMT)**
  - Similar to minimum spanning tree (MST)
  - But finding SMT is NP-complete
  - Many good heuristics introduced to find SMT

- **Algorithm**
  - Find MST
  - Pass horizontal and vertical lines from each terminal to get the Hannan grid (optimal solution is on this grid)
  - Convert each edge of the MST to an L-shaped route on Hannan grid (add a Steiner point at the corner of L)
Steiner Tree

- Hannan grid reduces solution space (smaller grid)
  - For min length RST, Steiner points always on Hannan grid
- Convert MST to rectilinear paths
  - Length bounded by 1.5 times optimal SMT length
- Use alternate “L” routes to find the minimum tree

MSP (length=11)  Steiner tree (len=13)
Steiner Tree Routing

- Can apply different costs to different regions (or horizontal/vertical preference)
- Order of the nets
  - Sequential
    - Use # of terminals, criticality, etc. to determine order
  - Parallel
    - Divide the chip into large regions, perform the routing in parallel
- Key to popularity
  - Fast (not theoretically, but practically)
  - Bounded solution quality
- Shortcomings
  - Difficult to predict or avoid congestion