Should we use HLS for structural design?

- Xilinx Vivado High Level Synthesis (HLS)
  - Algorithm to hardware synthesis
  - Automatic parallelization and scheduling
  - Pragma to steer mapping decisions

- Structural Design: Precise Cycle- and Bit-level Control
  - Examples: Network-on-Chip (NoC), Routers...
  - Challenge: Use untimed and sequential C code to describe cycle-accurate and concurrent hardware.

*Can we and Should we* Use Vivado-HLS for Structural Design?

- Exact structure and timing?
- Same quality as RTL?
- Improve productivity?

What worked very well

- Illustrative example here; see website for complete router source

What didn’t work so well

- Sequential C Language => Concurrent Hardware?

Can we? Yes; Should we? Depends.

- Case Study Overview
  - RTL Reference: CONNECT Parameterized NoC
  - Produced exact cycle- and bit-accurate replacements
  - Achieved comparable hardware cost and critical path

- Productivity Improvement
  - Natural C’s sequential reading is maintained. Utilize C’s facilities to capture more maintainable and scalable designs. 1126 lines of C++ VS. 2605 lines of Bluespec System Verilog
  - Still need to specify structural details to get desired hardware
  - Separation of functionality and structural details enables fast design iteration
  - Ordering discipline could be cumbersome when creating a netlist to compose submodules

- Hardware Quality (HLS VS. CONNECT)

Tech report and source code available

http://www.ece.cmu.edu/calcm/connect_hls