A Framework Testing Functional Correctness for FPGA Architecture Exploration

Ash Furrow / Kenneth B. Kent
University of New Brunswick
Faculty of Computer Science
w8320@unb.ca / ken@unb.ca

Outline

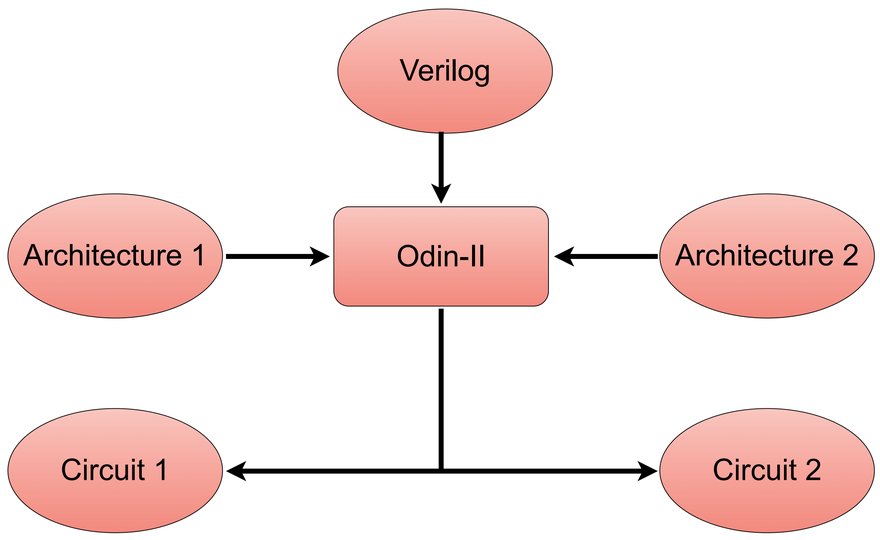
• Implementation of a simulator of Field Programmable Gate Array (FPGA) netlists

• Use of simulator to verify results of FPGA architecture exploration

• Implementation of generic hard block, memory, and multiplier simulation

Motivation

Given two or more FPGA architecture specifications, we want to compare them to make an objective decision about which is better in terms of speed, size, or other characteristics.



In order to be confident in the results of the architecture exploration, we must be certain that the two generated circuits behave as expected.

Background

Odin-II
• Used to compile verilog hardware descriptions into circuits
• Other tools in the tool flow optimize, place, and route
• Simulator was developed into Odin-II to make use of in-memory data structures

Example
Odin-II supports optimization for FPGA synthesis; it can employ multipliers, memories, and generic hard blocks. These necessitate the development of a standalone simulator instead of use of an existing solution.
Any simulator used to verifies results from Odin-II must allow researchers to test these optimizations

Development

Simulator performs a breadth-first search of the generated circuit. When dequeuing a node for computation, we need to ensure that the input values to the node have been calculated for the current cycle.

Multipliers use a standard algorithm for computing the result of arbitrary-length integers. Memories are simulated using dynamically allocated memory. Memory initialization files may be used to give initial values; otherwise, all values in a memory are zero by default.

Generic hard blocks pose a unique challenge. Researchers need to be able to specify any generic hard block. Our simulator, then, needs to be able to simulate any generic hard block, too. This is accomplished via dynamically-loaded compiled C libraries; researchers write the C which simulates their hard block and the simulator uses this in its circuit simulation

Results

The simulator processed large circuits at a modest speed; use of this tool to verify FPGA architecture exploration is completely tractable.

A few bugs in Odin-II's circuit compilation were identified during development of the simulator. Additionally, several issues with the microbenchmarks were discovered.

Issues regarding some shortfalls of Odin-II's ability to compile circuits were identified. Odin-II supported only flip flops with rising edges with a default value of zero. In reality, flip flops can be falling edge or asynchronous, and may not have a default value.

Current & Future Work

Issues still exist with simulation of sequential circuits, which are under current development. Christ Towler is currently working on reading in netlists circuits so that researchers can simulate circuits generated at any point in the tool flow.