The Improvement of VTR Project by using Carry-Chains and Power Specification

Sen Wang / Kenneth B. Kent
University of New Brunswick
Faculty of Computer Science
h3p24/ken@unb.ca

Outline

• Improving the hard-logic architectures in ODIN II

• Support power-aware CAD flow by adding power specification

Motivation

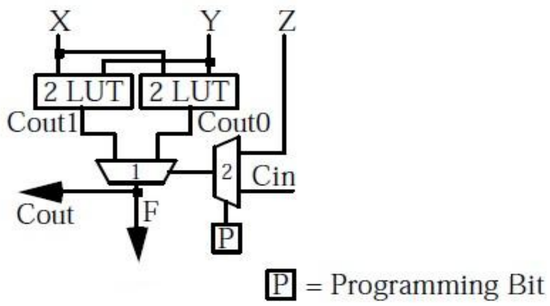
ODIN II is part of the Verilog-to-Routing (VTR) workflow, which is an open source academic FPGA CAD flow designed for exploration of architectures. In order to support the research of hypothetical FPGA architectures, ODIN II must support all the important FPGA architectural features. The goal for this project is to improve the VTR workflow by adding new functionality to ODIN II. This goal can be divided into two parts:

1. Provide the hard block carry-chain function in ODIN II.
2. Provide power specification in ODIN II

Background

The VTR workflow has three main parts: ODIN II, a Verilog hardware description language exploration tool; ABC, which handles logic synthesis, and VPR, which is responsible for packing, placing, and routing the circuits to the target FPGA architecture.

Carry-Chains are a very important method for most computations, including FPGA. Current FPGAs mainly use the basic ripple carry cell in their architectures. This element can produce all the functionality of its three inputs.



Power constraints are becoming more of a concern for hardware circuits. In the power-aware FPGA architecture, it is possible to use the power estimates to guide the placement and routing process.

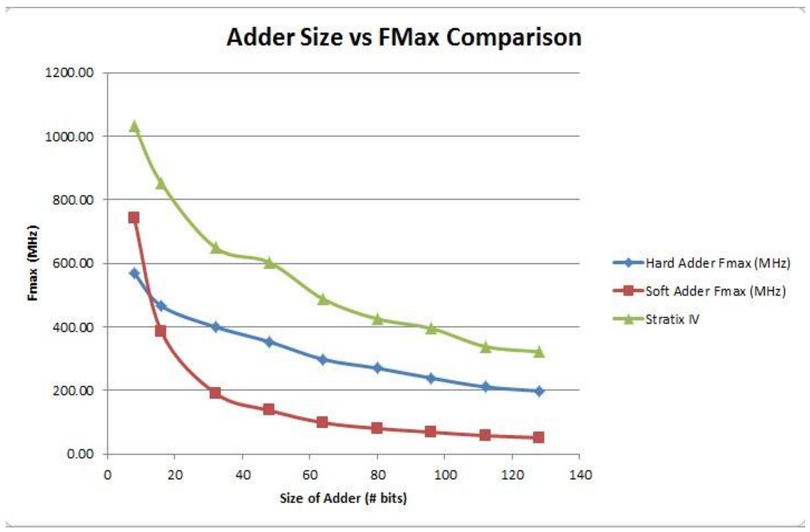
Solution

• To perform addition or subtraction, the user of ODIN II can change values in configuration file to choose different ways in order to implement it: a hard logic block, soft logic, or a combination of both hard and soft logic.

• To support the power-aware FPGA architectures, we will have a power specification for the FPGA in the architecture file. Also, a given circuit targeting the FPGA will include a power requirements file. While ODIN II reads the Verilog files and power requirements, the power specification will be considered before writing the BLIF file.

```
<module>
    <module_name>exampleInstance</module_name>
    <from>
            <port>a</port>
     </from><!--Optional-->
     <to>
            <port>sub</port>
    </to><!--Optional-->
    <power_constraint>0.03</power_constraint>
</module>
```

Results



The hard block carry-chain function in ODIN II is tested with different benchmarks. Using hard block carry-chain, ODIN II generates less nodes and connections, and uses less simulation time.