Synthesis and Improvement in Odin II

Bo Yan / Kenneth B. Kent
University of New Brunswick
Faculty of Computer Science
b.yan@unb.ca/ken@unb.ca

Introduction

Odin II has been developed for years by researchers among several universities, and it is continuously improved to be complete. The project of synthesis and improvement includes for loop support, AST simplification, and hard block reduction. The goal is to simplify the netlist and make the whole work flow efficient.

Background

The Verilog-to-Routing (VTR) project provides a complete and open-source framework for conducting FPGA architecture and CAD research and development. Odin II, as one core tool of VTR, compiles a Verilog source file and generates an abstract syntax tree (AST). The AST is then traversed, producing a netlist using a set of supported gates and logic blocks, and saves the result by using the Berkeley Logic Interchange Format (BLIF). It also includes a simulator which can be used to verify the functional correctness of the produced netlist.

For Loop Support

In order to allow an AST which includes a for loop statement to be recognized, the basic idea is to copy the branches showing the for loop's body as many times as the for loop expects. Then replace the for node by the copied branches. Intermediate variables are needed to be reduced, since Odin II cannot resolve the timing between statements. Figure 1 shows the modification of an AST including a for loop statement.

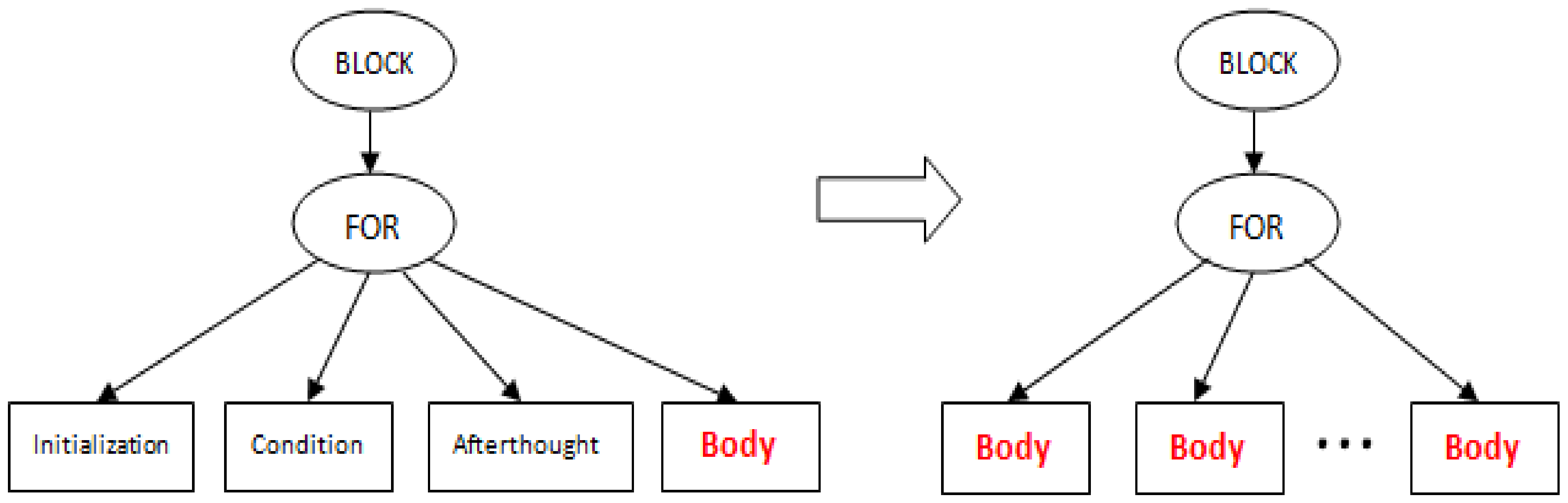

Figure 1. AST modification for a for loop statement

AST Simplification

The work is to detect the expressions stored in an AST which can be reduced, calculate and simplify these expressions, and rebuild this part of the AST. Figure 2 demonstrates a simple example of this work.

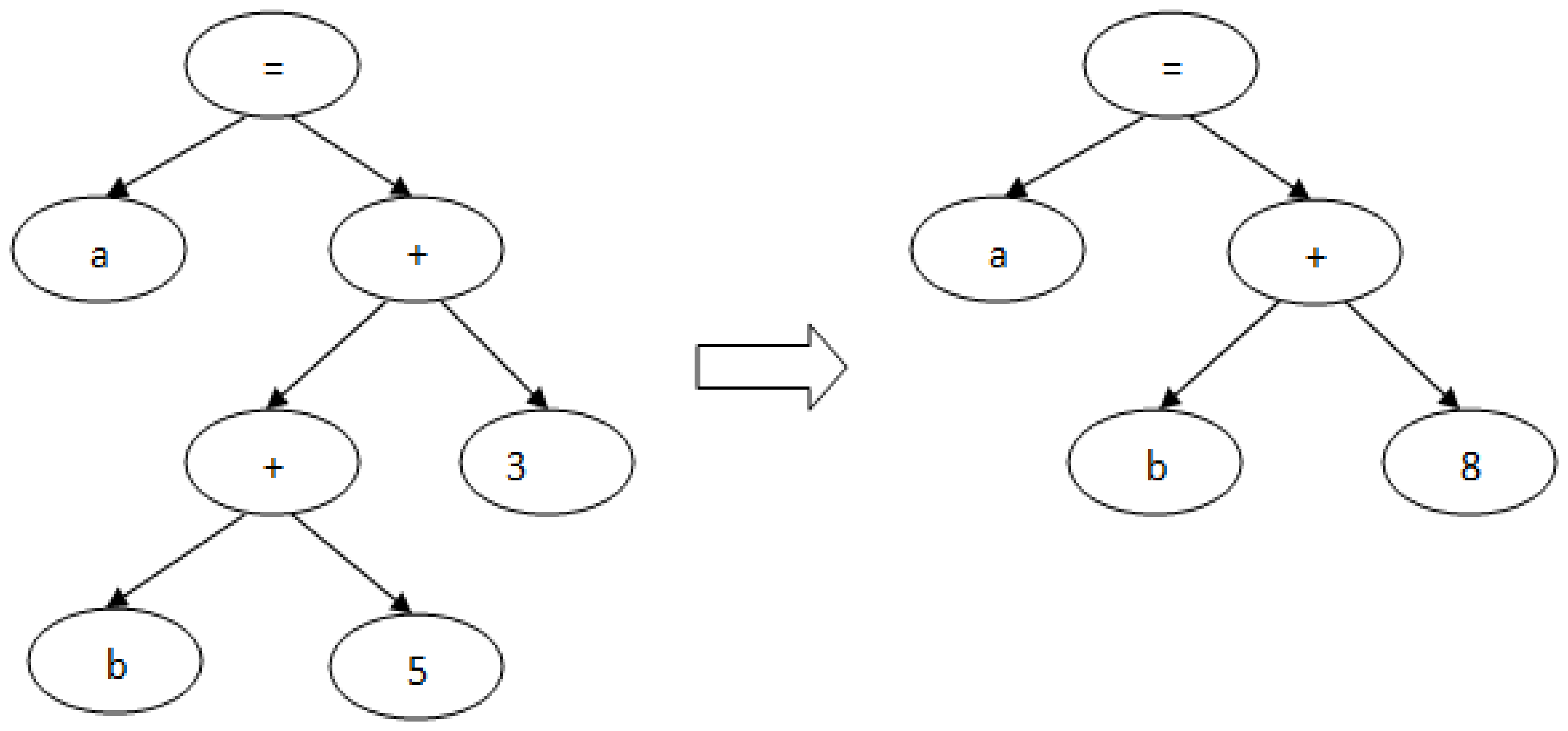

Figure 2. An example of AST Simplification

Hard Block Reduction

The work is to merge and reduce the nodes which have the high-level functions; that is, they have the same type, same number of input pins, and the same input ports. The basic principle of hard block reduction can be depicted by Figure 3.

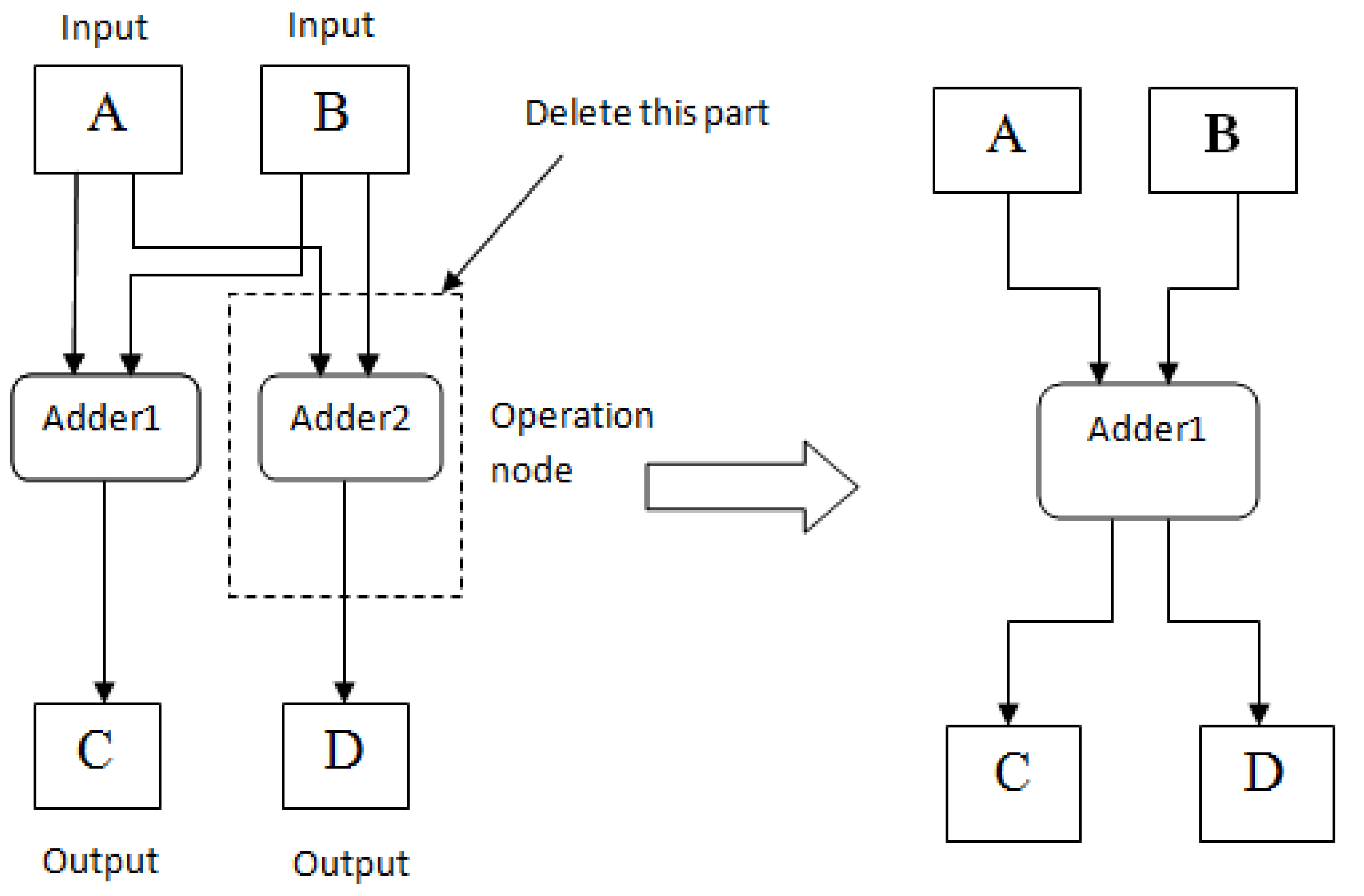

Figure 3. Basic principle of hard block reduction

Results

After the work is completed, a set of benchmarks are used for test and simulation. The functionality of Odin II is more complete once it can support for loop statement. With the synthesis and improvement, the netlist is simplified and the whole work flow can be more efficient.