Heterogeneous Programmable Logic Block Architectures

A. Koorapaty, V. Chandra, K.Y. Tong, C. Patel, L. Pileggi, H. Schmit
{aneeshk, vchandra, ktong, cpatel, pileggi, herman}@ece.cmu.edu
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Abstract

In this poster, we propose four new heterogeneous programmable logic blocks (PLBs) consisting of a combination of various sizes of look up tables (LUTs), multiplexers (MUXes), and logic gates. We demonstrate that these PLBs offer significant performance and density benefits over more homogeneous PLBs.

1. Introduction

Prior studies of PLB architectures determined the optimal LUT and cluster size for homogeneous PLBs with equivalently sized LUTs [1], and the density benefits of heterogeneous PLBs with various combinations of LUT sizes [3]. These studies excluded heterogeneous PLBs with a combination of LUTs, MUXes and logic gates, due to a lack of synthesis tools that effectively capture and exploit their features. Recently, a new fabric-specific synthesis approach [4] was presented that enables an accurate analysis of the performance and density benefits of such PLBs.

2. Analysis of Logic Gates

To determine which logic gates are well suited for heterogeneous PLBs, we map twenty standard MCNC benchmarks to 4-LUTs and 3-LUTs with Flowmap [2]. Then, we analyze the coverage of the functions in these mapped netlists provided by simple logic gates like 2 and 3-input Nand gates with programmable inversion (inputs and outputs available in both polarities). It can be shown that a 2-input Nand gate with programmable inversion (ND2WI) can implement 14 of 16 2-input functions, and a similar 3-input Nand gate (ND3WI) can implement 56 of the 256 3-input functions. Figure 1 illustrates the percentage of functions in the 3-LUT and 4-LUT mapped netlists that can be implemented by a ND3WI gate. Except for one error correction circuit, the ND3WI gate covers between 60-100% of the functions in the 3-LUT mapped netlists. For thirteen of twenty benchmarks (mostly logic circuits), the coverage is more than 90%. The ND3WI gate also achieves significant coverage of functions in 4-LUT mapped netlists.

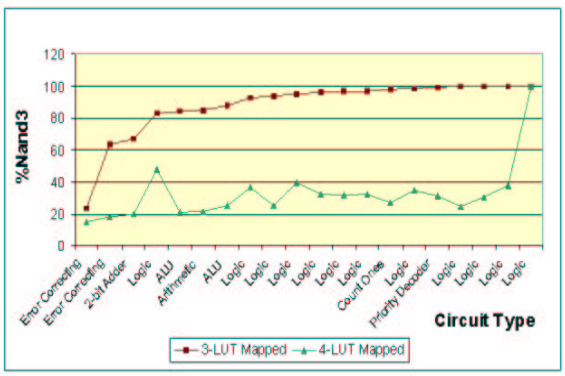

Figure 1. Utility of ND3WI Gate

Since an ND2WI gate can implement 14 of 16 two-input functions, two ND2WI gates driving a MUX can implement 196 of 256 3-input functions. We call this structure a 3-input Semi-LUT (S3 Gate). A similar analysis as above shows that an S3 gate can implement greater than 90% of the functions in the 3-LUT mapped netlists for seventeen of twenty benchmarks. Furthermore, the S3 gate consistently achieves a coverage between 20% to 40% for 4-LUT mapped netlists.

3. Heterogeneous Programmable Logic Blocks

Based on the results of the previous section, we propose four new heterogeneous PLB architectures, illustrated in Figure 2. PLB 1 can simultaneously implement a 2-input function, along with two 3-input functions of the kind implementable by the S3 gates. Also, the entire PLB can implement a 3-input LUT, or select 4-input functions. PLB 2 consists of one 3-LUT and three ND3WI gates. Each of the LUT inputs can be connected either to a primary input,

1530-1591/03 $17.00 © 2003 IEEE

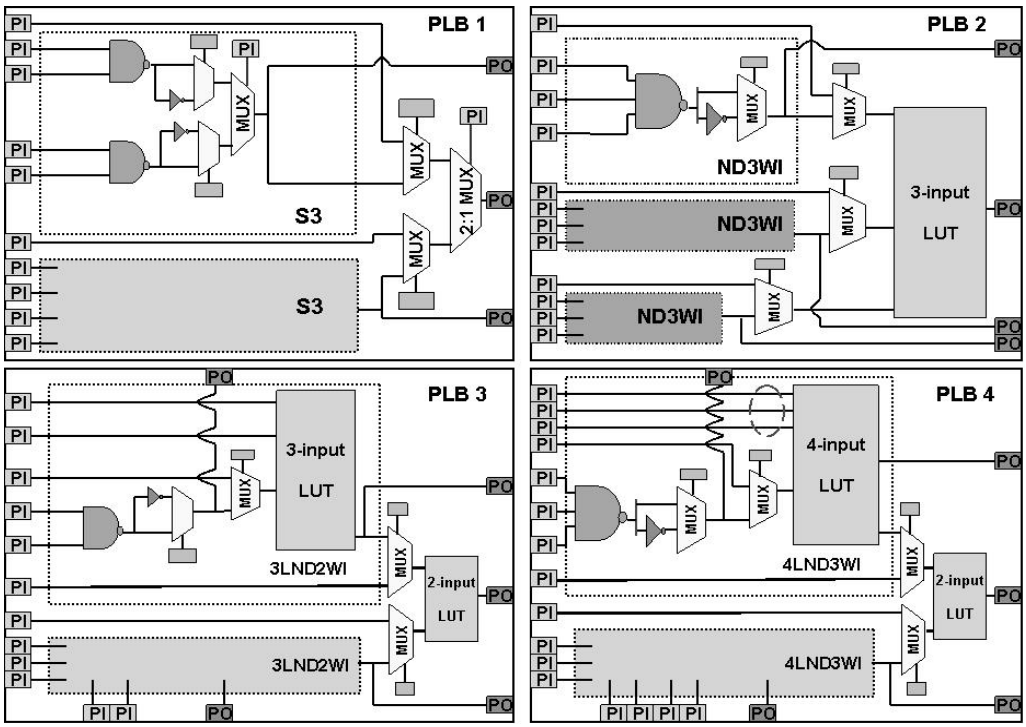

Figure 2. PLB Diagrams for the proposed Heterogeneous Architectures

or the output of one of the ND3WI gates. PLB 3 consists of one 2-LUT, two 3-LUTs, and two ND2WI gates. This PLB can simultaneously implement select 2-input functions and two 3-input functions, since the 3-LUTs and 2-LUT can be driven either by primary inputs, or the outputs of other logic elements. PLB 4 consists of one 2-LUT, two 4-LUTs, and two ND3WI gates, and can simultaneously implement select 3-input functions, two 4-input functions, and one 2-input function. The required polarity is guaranteed on all primary inputs for each of the PLBs.

4. Experimental Results

To evaluate the performance and density benefits of these PLBs, we map and pack a set of standard benchmarks for each PLB with the synthesis engine in [4], and place and route these netlists with VPR [1]. To fit the assumptions of the VPR logic block model [1], we approximate each of the logic elements in our PLBs as K-input LUTs. We also assume that each LUT input is driven by a 2:1 MUX, even though most of the inputs, like the three circled inputs on the 4-LUT in PLB 4, are hardwired to primary inputs.

To limit the performance gain due to heterogeneity, we use the lowest possible delays for the LUTs and worst case delays for the other elements. To eliminate the influence of a good initial placement, we use average values for ten VPR placements with different seeds, across a set of twenty benchmarks. Finally, we use two placement and routing strategies and three different routing architectures.

In our baseline routing architecture, all wires span 4 PLBs and 50% of the switches are buffered. For this architecture, PLB 3 achieves a 26% reduction in critical path delay over a homogeneous PLB with four 4-LUTs. PLB 1 follows PLB 3 with a reduction of 12%. When comparing the area-delay product, PLB 1 achieves a 48% reduction, followed by PLB 3, with a 26% reduction over the homogeneous PLB. These results hold for the other routing architectures as well, and are independent of all experimental parameters, including placement and routing strategies.

References

[1] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.

[2] J. Cong and Y. Ding. Flowmap: An optimal technology mapping algorithm for delay optimization in lookup-table based fpga designs. *IEEE Transactions Computer Aided Design*, 13:pp. 1–12, January 1994.

[3] J. He and J. Rose. Advantages of heterogeneous logic block architectures for fpgas. *IEEE Custom Integrated Circuits Conference*, May 1993.

[4] A. Koorapaty and L. Pileggi. Modular, fabric-specific synthesis for programmable architectures. *Field Programmable Logic and Applications*, September 2002.