A HybridLUT/Multiplexer-Based Runtime Relocation System for Circuits on Heterogeneous FPGA

S.Mohan M.E , R.Divya
[1]AP/ECE , [2]PG Scolar
Avs Engineering College

Abstract

Despite the clear potential of FPGAs to push the current power wall beyond what is possible with general-purpose processors, as well as to meet ever more exigent reliability requirements, the lack of standard tools and interfaces to develop reconfigurable applications limits FPGAs' user base and makes their programming not productive. Runtime relocation of circuits on fieldprogrammable gate arrays (FPGAs) has been proposed for achieving many desirable features including fault tolerance, defragmentation, and system load balancing. Propose a Hybrid LUT/Multiplexer FPGA Logic Architectures a six-input LE based on a 4-to-1 MUX, MUX4, that can realize a subset of six-input Boolean logic functions, and a new hybrid complex logic block (CLB) that contains a mixture of MUX4s and 6-LUTs.

I. INTRODUCTION

Digital electronics is concerned with circuits which represent information using a finite set of output states. Most of the applications use in fact just two states, which are often labelled ‘0’ and ‘1’. Behind this choice is the fact that the whole Boolean formalism then becomes available for the solution of logic problems, and also that arithmetic using binary representations of numbers is a very mature field.

Different mappings between the two states and the corresponding output voltages or currents define different logic families. For example, the Transistor–Transistor Logic (TTL) family defines an output as logic ‘1’ if its voltage is above a certain threshold (typically 2.4 V). For the same family, if we set the input threshold for logic ‘1’ as 2 V, we will have a margin of 0.4 V which will allow us to interconnect TTL chips inside a design without the risk of misinterpretation of logic states. This complete preservation of information even in the presence of moderate amounts of noise is what has driven a steady shift of paradigm from analog to digital in many applications. Here we see as well another reason for the choice of binary logic: from a purely electrical point of view, having only two different values for the voltages or currents used to represent states is the safest choice in terms of design margins.

A. Related work

FeiWangtet al[1] proposes an Exploring Architecture Parameters for Dual-Output LUT based FPGAs. Dual-output lookup tables (LUTs) are main stream in the design of commercial FPGA products. André Lalevée et al[2] studies about a Dynamic and partial reconfiguration of Field Programmable Gate Arrays (FPGA) enable to reuse logic resources for several applications which are scheduled in a sequential order or which are loaded on demand.

AdewaleAdetomi employs a novel network architecture to enable dynamic communication and thus improve the flexibility of circuit relocation. By using the clock infrastructure of the FPGA as the physical network links for tasks in a 4-node star network, we have shown that dynamic communication between relocatable circuits can be achieved without incurring any overheads of time and resources, save for only 32 slices used for the Network Interface.

MarwaHannachi et al presents a design method for relocation of variable-sized hardware task on SRAM-based FPGAs for adaptive systems using dynamic partial reconfiguration (DPR). The proposed relocation procedure takes into account the communication between different reconfigurable regions and static region.Sundar Prakash and BalajiMuthusamyet al[5] presents low-power dissipation using fpga Architecture. Power optimization is the process of generating the best design in digital VLSI circuits without violating design specifications. N.Rajagopala Krishnan and K.Sivasuparamanyanet al[6] presents a Reconfigurable Low Power FPGA Design with Autonomous Power Gating and LEDR Encoding. Field Programmable Gate Arrays (FPGAs) are widely used to implement special purpose processors.

II. EXISTING SYSTEM

Runtime relocation of circuits on field-programmable gate arrays (FPGAs) has been proposed for achieving many desirable features including fault tolerance, defragmentation, and system load balancing. However, the changes in the architectural composition of FPGAs have made relocation more challenging mainly because FPGAs have become more heterogeneous. Previous and state-of-the-art circuit relocation systems on FPGAs have relied only on direct bitstream relocation which requires the source and destination resource layouts to be the same, as well as access to the design bitstream for manipulation. Hence, their efficiency on modern heterogeneous chips greatly reduces, and mostly cannot be applied to encrypted bitstreams of intellectual property blocks. In this brief, we present a circuit relocator which augments direct bitstream relocation with a functionality-based relocation scheme.

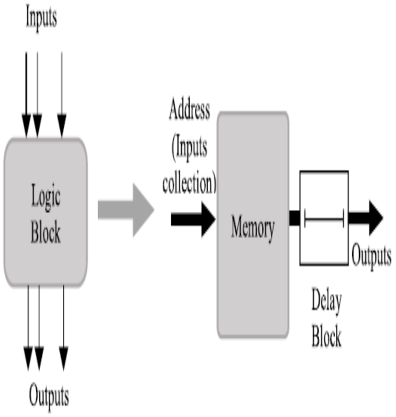


Fig. 1Transformation of logic block to memory block.

An author propose a relocation manager to improve the number of relocations for a circuit on heterogeneous FPGAs. The proposed relocator augments direct bitstream relocation with a functionality-based relocation. The functionality-based relocation presented in this brief relies on the technique of replicating the functionality of a circuit with a look-up-table (LUT) or a memory block in runtime for selected circuits.

III. PROPOSED SYSTEM

Hybrid configurable logic block architectures for field-programmable gate arrays that contain a mixture of lookup tables and hardened multiplexers are evaluated toward the goal of higher logic density and area reduction. Multiple hybrid configurable logic block architectures, both nonfracturable and fracturable with varying MUX:LUT logic element ratios are evaluated across two benchmark suites (VTR and CHStone) using a custom tool flow consisting of LegUp-HLS, Odin-II front-end synthesis, ABC logic synthesis and technology mapping, and VPR for packing, placement, routing, and architecture exploration. Technology mapping optimizations that target the proposed architectures are also implemented within ABC. With architecture-aware technology mapper optimizations in ABC, additional area is saved, post-place-and-route.

Two families of architectures were created: 1) without fracturable LEs and 2) with fracturable LEs. In this project, the fracturable LEs refer to an architectural element on which one or more logic functions can be optionally mapped. Nonfracturable LEs refer to an architectural element on which only one logic function is mapped. In the nonfracturable architectures, the MUX4 element shown in Fig is use together with nonfracturable 6-LUTs. This element shares the same number of inputs as a 6-LUT lending for fair comparison with respect to the input connectivity.

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry.All your designs for this lab must be specified in the above Verilog input format. Note that the state diagram segment does not exist for combinational logic designs.

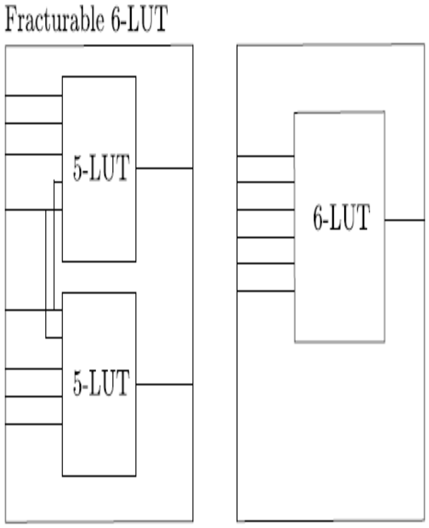
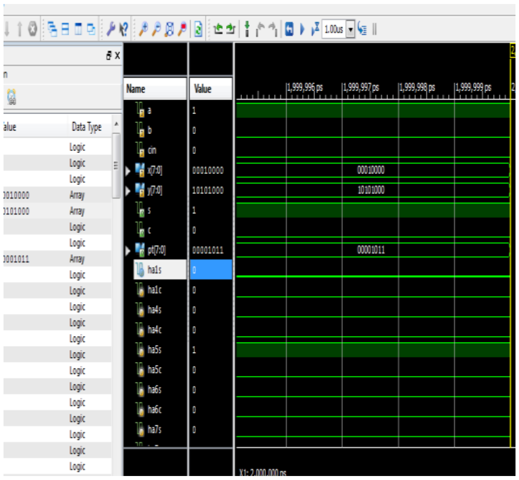


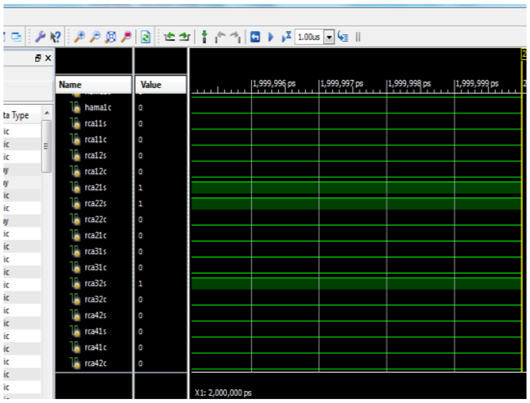
FIG 4.1: Fracturable 6-LUT that can be fractured into two 5-LUTs with two shared inputs.

IV. SIMULATION RESULTS



```
Name      Value
a         1
b         0
cin       0
x[7:0]    00010000
y[7:0]    10101000
s         1
c         0
pt[7:0]   00001011
ha1s      0
ha1c      0
ha4s      0
ha4c      0
ha5s      1
ha5c      0
ha6s      0
ha6c      0
ha7s      0
```

Fig 6.1. simulation result



```
Name      Value
hama1c    0
rca11s    0
rca11c    0
rca12s    0
rca12c    0
rca21s    1
rca22s    1
rca22c    0
rca21c    0
rca31s    0
rca31c    0
rca32s    1
rca32c    0
rca42s    0
rca41s    0
rca41c    0
rca42c    0
```

Fig.6.2 simulation result-2

CHAPTER-7

V. XILINX SYNTHESIS REPORT

The existing and proposed has been simulated and the synthesis report can be obtained by using Xilinx ISE 12.1i. The various parameters used for computing existing and proposed systems with Spartan-3 processor are given in the table 7.1.

Table1: Comparison Table

S.NO	PARAMETER	EXISTING	PROPOSED
	Slice	89	53
	LUT	158	98

CONCLUSION

We have proposed a new hybrid CLB architecture containing MUX4 hard MUX elements and shown techniques for efficiently mapping to these architectures. Weighting of MUX4-embeddable functions with our MuxMap technique combined with a select mapping strategy provided aid to circuits with low natural MUX4-embeddable ratios. From our first set of experiments with nonfracturable architectures, area reductions were seen for a 4:6 MUX4:LUTarchitecture in the CHStone suite with a 2:8 architecture most viable for the VTR suites with an area savings. Our second set of experiments with fracturable architectures showed that the flexibility of a fracturable LUT is very powerful, reducing the impact of the MUX4 LEs, yields smaller area with less aggressiveof existing architectures, respectively. Overall, the addition of MUX4s to FPGA architectures minimally impact FMax and show potential for improving logic-density in nonfracturable architectures and modest potential for improving logic density in fracturable architectures.

REFERENCES

[1]. J. Rose *et al.*, “The VTR project: Architecture and CAD for FPGAs from verilog to routing,” in *Proc. ACM/SIGDA FPGA*, 2012, pp. 77–86.

[2]. Y. Hara, H. Tomiyama, S. Honda, and H. Takada, “Proposal and quantitative analysis of the CHStone benchmark program suite for practical C-based high-level synthesis,” *J. Inf. Process.*, vol. 17, pp. 242–254, Oct. 2009.

[3]. A. Caniset *al.*, “LegUp: High-level synthesis for FPGA-based processor/accelerator systems,” in *Proc. ACM/SIGDA FPGA*, 2011, pp. 33–36.

[4]. E. Ahmed and J. Rose, “The effect of LUT and cluster size on deepsubmicron FPGA performance and density,” *IEEE Trans. Very Large Scale Integr. (VLSI)*, vol. 12, no. 3, pp. 288–298, Mar. 2004.

[5]. J. Rose, R. Francis, D. Lewis, and P. Chow, “Architecture of field programmable gate arrays: The effect of logic block functionality on area efficiency,” *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1217–1225, Oct. 1990.

[6]. H. Parandeh-Afshar, H. Benbihi, D. Novo, and P. Ienne, "Rethinking FPGAs: Elude the flexibility excess of LUTs with and-inverter cones," in *Proc. ACM/SIGDA FPGA*, 2012, pp. 119–128.

[7]. J. Anderson and Q. Wang, "Improving logic density through synthesis inspired architecture," in *Proc. IEEE FPL*, Aug./Sep. 2009, pp. 105–111.

[8]. J. Anderson and Q. Wang, "Area-efficient FPGA logic elements: Architecture and synthesis," in *Proc. ASP DAC*, 2011, pp. 369–375.

[9]. J. Cong, H. Huang, and X. Yuan, "Technology mapping and architecture evalution for k/m-macrocell-based FPGAs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 10, no. 1, pp. 3–23, Jan. 2005.

[10]. Y. Hu, S. Das, S. Trimberger, and L. He, "Design, synthesis and evaluation of heterogeneous FPGA with mixed LUTs and macro-gates," in *Proc. IEEE ICCAD*, Nov. 2007, pp. 188–193.