International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A Review on Hybrid Lut / Multiplexer Fpga Logic Architectures

Pathan Mubeena & Yendluri Kondaiah
M.Tech (VLSI) Department of ECE, Priyadarshini Institute of Technology and Management. Pulladigunta,Guntur, A.P.

Associate professor Department of ECE, Priyadarshini Institute of Technology and Management Pulladigunta, Guntur, A.P

ABSTRACT:

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 non-fracturable and factorable with varying MUX:LUT logic element ratios are evaluated across two benchmark suites (VTR and CH Stone) using a custom tool flow consisting of Leg Up-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. Experimentally, we show that for nonfracturable architectures, without any mapper optimizations, we naturally save area post place and route; both accounting for complex logic block and routing area while maintaining mapping depth. With architecture-aware technology mapper optimizations in ABC. . For fracturable architectures, the proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

KEYWORDS: FPGA, Multiplexer logic element, Complex logic block, mapping technologies

INTRODUCTION:

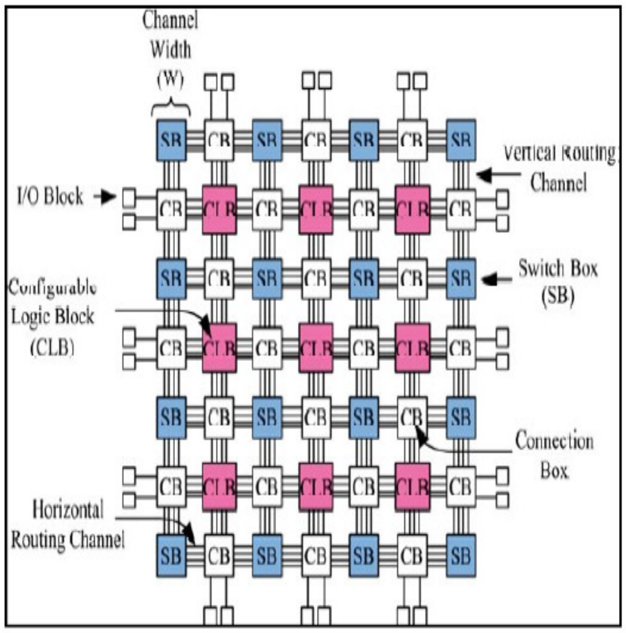
Field Programmable Gate Arrays (FPGAs) are a step in the continuum ofevolution of Integrated Circuits (IC). FPGAs are reprogrammable silicon chips and are one of

the Programmable Logic Devices (PLDs) that can be configured to implement customized hardware functionality of any digital circuit. Due to their flexibility, programmability, capacity for various applications and low end product cycle, FPGAs are highly desirable for implementation of digital circuits. The main difference between FPGAs and conventional fixed logic implementations, such as Application Specific Integrated Circuits (ASICs), is that the designer can program the FPGA on-site. Using an FPGA instead of a fixed logic implementation eliminates the non-recurring engineering (NRE) costs and significantly reduces time-to-market. FPGA chips adoption across all industries is driven by the fact that FPGAs combine the best parts of ASICs and processor-based systems. These reprogrammable silicon chips also have the same flexibility of software running on a processor-based system, but it is not limited by the number of processing cores available. The software tools provide the programming environment, whereas FPGA circuitry is truly a “hard” implementation of program execution.

FPGAs are reprogrammable silicon chips that can be configured to implement customized hardware functionality by using its prebuilt logic blocks and programmable 2 routing resources without even picking up a breadboard or soldering iron. They are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. The FPGA share a common history with most PLDs. PLDs is divided into three basic architecture types: Simple Programmable Logic Devices (SPLD), Complex Programmable Logic Devices (CPLD) and FPGA. The first of this kind of devices was the Programmable Read Only Memory (PROM). Philips invented the Field Programmable Logic Array (FPLA) in the 1970s which was driven by need of specifically implementing logic circuits. It consisted of two planes, a programmable wired AND-plane and the other as wired OR that could implement functions in the Sum of Products form. But in both the devices (PROM and FPLA), sequential logic like a flip-flop to create synchronous designs or state machine are missing.

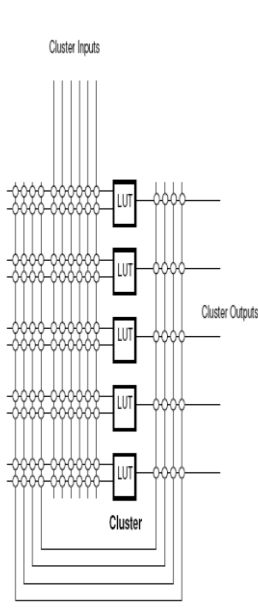
FPGA- ARCHITECTURE

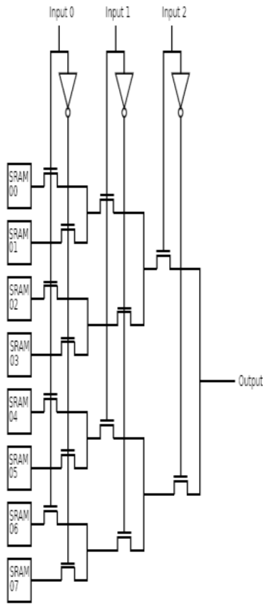
The FPGA-architecture consists of many logic modules which are placed in arraystructure and these modules are configurable at site and are therefore called as Configurable Logic Blocks (CLBs). The channels between the CLBs are used for routing. The arrays of the CLBs are surrounded by programmable I/O modules and connected via programmable interconnects. There are two subclasses of FPGA architecture depending on granularity of CLBs: Coarse-grained and Fine-grained FPGAs.



FPGA architecture

The coarse-grained FPGAs have very large logic modules/CLBs with sometimes two or more sequential logic elements, whereas the fine-grained FPGAs have very simple logic modules. A conventional, island-style FPGA can be viewed as an array of CLBs connected by programmable interconnects i.e. switchboxes (SBox) and connection boxes (CBoxes) as shown in Figure Parvez and Mehrez (2011). N programmable lookup tables (LUTs) are connected together using internal interconnect inside the CLB. The LUT is simply a circuit that selects the output of aStatic Random Access Memory(SRAM) cell based on the LUT's k inputs: by programming appropriate values in the SRAM cells the LUT can implement any k-input function. Figure 1.4b shows the architecture of 3-input LUT. All programmable interconnect is implemented using the simple, unidirectional switch that takes several inputs and selects one output. The circuit consists of input end buffers for each input that serve to isolate the switch, a multiplexer, and an output buffer that drives a longwire segment. Each of the input and output buffers can be built using a single or multiple staged inverters as described by Parvez and Mehrez (2011). Farooq, et.al (2012) shows how to assemble these switches into a switchbox and the basic 3-input switch circuit.





CLB with 5 look-up tables (LUTs)

3 input LUT

LITERATURE REVIEW

Recent works have shown that the heterogeneous architectures and synthesis methods can have a significant impact on improving logic density and delay, narrowing the ASIC–FPGA gap. Works by Anderson and Wang with "gated" LUTs, then with asymmetric LUT LEs, show that the LUT elements present in commercial FPGAs provide unnecessary flexibility. Toward improved delay and area, the macro cell-based FPGA architectures have been proposed. These studies describe significant changes to the traditional FPGA architectures, whereas the changes proposed here build on architectures used in industry and academia. Similarly, and-inverter cones have been proposed as replacements for the LUTs, inspired by and-inverter graphs (AIGs).
PURNAPRAJNA and IENNE explored the possibility of repurposing the existing MUXs contained within the Xilinx Logic Slices. Similar to this work, they use the ABC priority cut mapper as well as VPR for packing, place, and route. However, their work is primarily delay based showing an average speed up of 16% using only ten of 19 VTR7 benchmarks.
In this article, we study the technology mapping problem for a novel field programmable gate array (FPGA) architecture that is based on k-input single-output programmable logic array- (PLA-) like cells, or, k/m-macro cells. Each cell in this architecture can implement a single output function of up to k inputs and up to m product terms. We develop a very efficient technology mapping algorithm, km flow, for this new type of architecture. The experimental results show that our algorithm can achieve depth-optimality on almost all the test cases in a set of 16 Microelectronics Center of North Carolina (MCNC) benchmarks. Furthermore it is shown that on this set of benchmarks, with only a relatively small number of product terms ($m \leq k+3$), the k/m-macro cell based FPGAs can achieve the same or similar mapping depth compared with the traditional k input single-output lookup table- (k-LUT-) based FPGAs. We also investigate the total area and delay of k/m-macro cell-based FPGAs and compare them with those of the commonly used 4-LUT-based FPGAs. The experimental results show that k/m-macro cell-based FPGAs can outperform 4-LUT-based FPGAs in terms of both delay and area after placement and routing by VPR on this set of benchmarks This paper presents experimental measurements of the differences between a 90-nm CMOS field programmable gate array (FPGA) and 90-nm CMOS standard-cell application specific integrated circuits (ASICs) in terms of logic density, circuit speed, and power consumption for core logic. We are motivated to make these measurements to enable system designers to make better informed choices between these two media and to give insight to FPGA makers on the deficiencies to attack and, thereby, improve FPGAs. We describe the methodology by which the measurements were obtained and show that, for circuits containing only look-up table-based logic and flip-flops, the ratio of silicon area required to implement them in FPGAs and ASICs is on average 35. Modern FPGAs also contain "hard" blocks such as multiplier/accumulators and block memories. We find that these blocks reduce this average area gap significantly to as little as 18 for our benchmarks, and we estimate that extensive use of these hard blocks could potentially lower the gap to below five. The ratio of critical-path delay, from FPGA to ASIC, is roughly three to four with less influence from block memory and hard multipliers. The dynamic power consumption ratio is approximately 14 times and, with hard blocks, this gap generally becomes smaller.
In this paper the new architectural proposals are routinely generated in both academia and industry. For FPGA's to continue to grow, it is important that these new architectural ideas are fairly and accurately evaluated, so that those worthy ideas can be included in future chips. Typically, this evaluation is done using experimentation. However, the use of experimentation is dangerous, since it requires making assumptions regarding the tools and architecture of the device in question. If these assumptions are not accurate, the conclusions from the experiments may not be meaningful. In this paper, we investigate the sensitivity of FPGA architectural conclusions to experimental variations. To make our study concrete, we evaluate the sensitivity of four previously published and well-known FPGA architectural results: lookup-table size, switch block topology, cluster size, and memory size. It is shown that these experiments are significantly affected by the assumptions, tools, and techniques used in the experiments.

Technology mapping using ABC

ABC was used for technology mapping, with modifications that allow for MUX4- embeddable function identification and MUX2- embeddable function identification in the case of fracturable MUX4s and custom mapping. The internal data structure used within the ABC is an AIG, where the logic circuit is represented using 2-input AND gates with inverters. Priority Cuts mapping in ABC (invoked with the, if command) was modified to perform our custom technology mapping. This mapper traverses the AIG from primary inputs to primary outputs finding intermediate mappings for internal nodes and finally the primary outputs, using a dynamic programming approach. The priority cuts mapper performs multiple passes on the AIG to find the best cut per node. For depth-oriented mapping, the mapper first

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

prioritizes mapping depth then optimizes for area discarding cuts whose selection would increase the overall depth of the mapped network. Based on this standard mapper, two mapper variants were produced and evaluated. The first variant, Natural Mux, evaluates and identifies internal functions that are MUX4-embeddable, agnostic of the target architecture; i.e., this flow uses the default priority cuts mapping and performs a post processing step to identify MUX4-embeddable functions. From this mapping, we can evaluate what area savings are possible without any mapper changes. The second variant Mux Map, area-weights the MUX4-embeddable cuts relative to 6-LUT cuts, thereby establishing a preference for selection/creation of MUX4- embeddable solutions.

Modeling using VPR:
VPR was used to perform architectural evaluation. The standard ten 6-LUT CLB architecture in 40-nm included with the VPR distribution was used for baseline modeling. The hybrid CLBs shown in Figs. 3.1 and 3.2 were modeled using the XML-based VPR architectural language. The snippet from the architecture file for the physical block hardened MUX4 element, this code specifies a MUX4 as a six-input one-output black box to the VPR. In addition, since all MUX4s can also be mapped to the 6-LUTs, an additional mode was added to the 6-LUT physical block. The mode concept allows the VPR packer to pack LUTs into LUTs (as usual), but also enables MUX4s to be packed into the LUTs. The architectures with CLBs having MUX4: LUT ratios from 1:9 to 5:5 were created from the baseline 40-nm architectures with delays obtained through circuit simulations of the MUX4 variants. Importantly, we made minor modifications to the VPR packing algorithm itself, so that the MUX4 net list elements are preferred to be packed into the MUX4 Les in the architecture (while limiting packing MUX4 net list elements into LUTs). The modifications involved changing the attraction function during the CLB packing. One change was to ensure that the logic functions that were MUX4 embeddable were preferentially packed into a physical MUX4 element and not into an LUT. Another was to apply a negative weight on MUX4-embeddable functions when the current CLB's physical MUX4 elements are all occupied also preventing MUX4-embeddable functions from being placed into the LUTs. Without this, the MUX4 net list elements might needlessly consume LUTs, which should be reserved, where possible, for those net list elements that demand their flexibility. This becomes doubly important for fracturable architectures, since their packing problem is more complex. Without this modification, a significant CLB usage increase was observed across all benchmark sets.

SIMULATION RESULTS

The proposed circuits are simulated and synthesized by using modalism and xilinx12.1 which occurs low area than the existing. The experimental results are given in Table 1 and the simulation results of layout and the waveforms are shown in the fig.9.1 and fig.9.2. Then the RTL schematic of the proposed are shown in fig.9.2

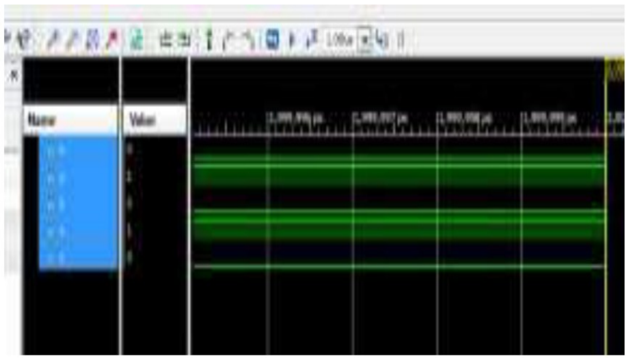


Fig 9.1 simulation results

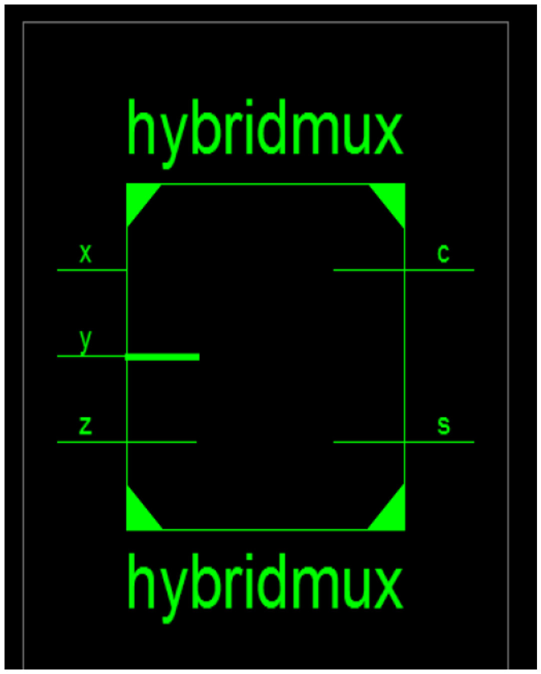
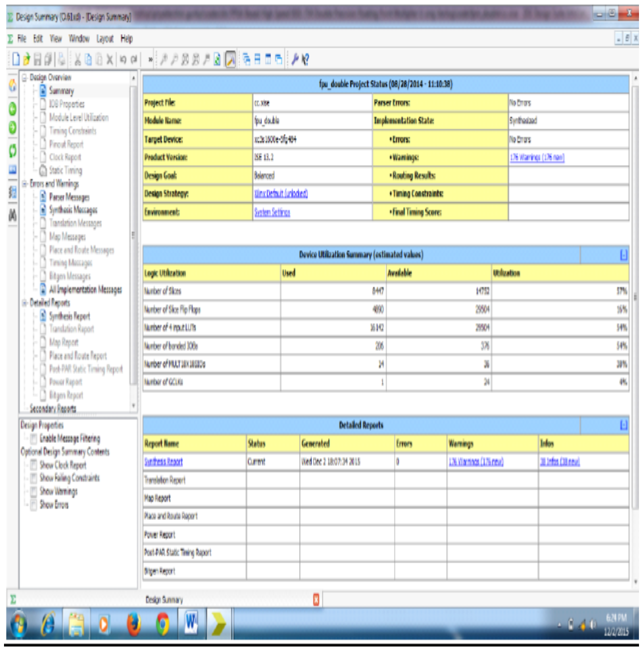


Fig.9.2 RTL schematic



fpu_double Project Status (08/28/2014 - 11:10:38)			
Project File:	cc.xise	Parser Errors:	No Errors
Module Name:	fpu_double	Implementation State:	Synthesized
Target Device:	xc3s1600e-5fg484	•Errors:	No Errors
Product Version:	ISE 13.2	•Warnings:	176 Warnings (176 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	8447	14752	57%
Number of Slice Flip Flops	4890	29504	16%
Number of 4 input LUTs	16142	29504	54%
Number of bonded IOBs	206	376	54%
Number of MULT18X18SIOs	14	36	38%
Number of GCLKs	1	24	4%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Dec 2 18:07:34 2015	0	176 Warnings (176 new)	38 Infos (38 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Figure 9.3: Synthesis report

CONCLUSION

We have proposed a new hybrid CLB architecture containing MUX4 hard MUX elements and shown techniques for efficiently mapping to these architectures. We also provided analysis of the benchmark suites post mapping, discussing the distribution of functions within each benchmark suite. The area reductions for non-fracturable architectures, is 8% and MUX4: LUT ratio is 4:6 and in the case of fracturable architecture the area reductions are 2%.The CH Stone benchmarks being high level synthesized with Leg Up-HLS also showed marginally better performance and this could be due to the way Leg Up performs HLS on the STUDIES Volume VIII /Issue 1 / DEC 2016 IJPRES CH Stone benchmarks themselves. Overall, the addition of MUX4s to FPGA architectures minimally impact F Max and show potential for improving logic-density in non-fracturable architectures and modest potential for improving logic density in fracturable architecture.

REFERECES

1. I. KUON R. Tessier and J. Rose. FPGA Architecture: Survey and Challenges. Foundations and Trends in Electronic Design Automation, 2(2):135–253, 2008.

2. Altera Corp. STRATIX III Device Handbook, Vol.1. 2006.

3. Xilinx Inc. Virtex-5 Family Overview - LX, LXT, and SXT Platforms. 2007.

4. I. KUON and J. Rose. Measuring the Gap between FPGAs and ASICs. IEEE Transactions on Computer-Aided Design (CAD) of Integrated Circuits and Systems, 26(2):203–215, 2007.

5. S.J.E.WILTON. Architecture and Algorithms for Field-Programmable Gate Arrays with Embedded Memory. PhD dissertation, University of Toronto, 1997.

6. ACTEL, ProASIC3 Flash Family FPGAs Datasheet: Device Architecture, 2007.

7. G.L. Zhang and P.H.W. Leong and C.H. Ho and K.H. TSOI and C.C.C. Cheung, D. Lee, R.C.C. Cheung and W. LUK. Reconfigurable Acceleration for Monte Carlo Based Financial Simulation. In Proc. International Conference on Field-Programmable Technology (FPT), pages 215–222, 2005.

8. J.D. Owens, M. Houston, D. LUEBKE, S. Green, J.E. Stone and J.C. Phillips. GPU Computing Proceedings of the IEEE, 96(5):879–899, May 2008.

9. M. ZECHNER and M. GRANITZER. Accelerating K-Means on the Graphics Processor via CUDA In-Proc. International Conference on Intensive Applications and Services (INTENSIVE), pages 7–15, 2009.