

Analysis of Technology Mapping Algorithm for Logic optimization of Symmetrical FPGA Architecture through Hybrid LUTs/PLAs

Sunil Kr. Singh
Ph.D, Research scholar
Uttarakhand Technical University,
Uttarakhand, INDIA
anujsunilsingh@yahoo.co.in

R. K. Singh
Professor
Uttarakhand Technical University
Dehradun, Uttarakhand, INDIA
rk Singhkec123@rediffmail.com

M. P. S. Bhatia
Professor
CSE Deptt. NSIT,
Dawarka New Delhi.,INDIA
bhatia.mps@gmail.com

ABSTRACT:

Reconfigurable computing using Field Programmable Devices (FPD) provides a method to utilize the available logic resources on the chip for various computations. The basic ability of reconfigurable computing is to perform computations in hardware to increase performance, while retaining the flexibility of application software. The purpose of this paper is to analysis the effect of technology mapping algorithms on logic density of FPGA devices for reconfigurable computing system using Hybrids of Look up tables (LUTs) from FPGA and Programmable logic arrays (PLAs) from CPLD architectures. LUTs/PLAs are both contributing particular strengths in the area of reconfigurable system design. We identified Hybrid LUTs/PLAs architectures as Hybrid Reconfigurable Computing Architectures (HRCA). The basis of the HRCA is that some parts of digital circuits are well-suited for execution with LUTs, but other for PLAs structures. The technology mapping step converts the user define gate level network into a network of LUTs. We evaluate the extensive comparison of technology mapping algorithms over 20 MCNC benchmark circuits from a common application ported on the given circuit by using the following algorithms, chortle-d, mispga-delay, FlowMap, and DAG-Map. The primary objective of this paper to find suitable optimal mapping algorithm in terms of logic density minimization with respect to reduce the number of K-LUT's used in the technology mapping solution. It also offers some improvements in computation speedup and power consumption. Initially results indicate that noteworthy logic area of symmetrical FPGA is reduced by using flow Map technology mapping algorithm on HRCA.

Research in the field of reconfigurable computing is increasing speedily in the area of computing science and electronic engineering. Reconfigurable computing utilizes hardware resources that can be tailored at run-time to give greater flexibility without compromising on performance. Reconfigurable computing devices agree to assemble both requirements i.e. flexibility and performance. Reconfigurable systems have evolved from Field Programmable devices (FPDs). FPDs have emerged as a competitive alternative to Application Specific integrated Circuits (ASICs) to implement designs. There are two main types of Field Programmable devices, Field Programmable Gate Arrays (FPGA) and Complex Logic Device (CPLD). They are both extensively used and each device contributes particular strength in the development of reconfigurable computing system. We combine the technology of FPGA and CPLD into one architecture called HRCA. The basis of HRCA is that some parts of digital circuit are well suited for implementation with LUTs, but other part parts benefit more from product term structure with PLA. User define circuit usually describe in hardware description language (HDL) like VHDL or Verilog. By using synthesis tool, this defines circuit converted into the basic logic gate. Technology mapping algorithm play an important role to convert the basic logic gates into LUTs and flip-flops according to FPGA architecture. This paper presents a academic breakthrough which shows that different technology mapping algorithm, like chortle-d[1], mispga-delay[3], FlowMap[4], and DAG-Map[2], have their own mechanism to convert gate level circuit into K-LUT (where K is no of input to LUT) and select the best one technology mapping with minimize the number of K-LUTs in LUT-based FPGA i.e. optimize the use of LUT for an application.

1. Introduction

2. Related Research Work

Field Programmable Devices (FPDs) face many challenges from lower speed-performance and less logic capacity in comparison to custom manufactured technologies, such as mask-programmed gate arrays. However, a lot of research has been devoted to improving FPD architecture. New architecture continues to emerge as main research in industry and academia with advanced total logic capacity and better speed-performance. A highlight of some recent research efforts on FPGA logic blocks is presented here.

In many reconfigurable embedded systems size, power and cost optimizations are the central goals. In those systems, the growing need of more computation power that contradict with size, power and cost optimization put a lot of pressure on researchers who must discover a good balance of all contradicting goals. In the July 2005 edition of the Altera Stratix Device Handbook[5], Stratix is an SRAM-based island-style FPGA containing many mixed computational elements. The main element is the logic array block (LAB), which contain 10 logic elements (LEs). The general architecture of the LE is much related to the structure that we try use to develop an HRCA, i.e. single 4-LUT function generator and a programmable register. In the FPGA research done by J.Rose, he focused on logic-block density. Assuming a LUT-based architecture, authors change the number of inputs to a LUT to measure the effects on implementation of a benchmark circuit set. Their conclusion is that LUTs with 4 or 5 inputs yield the best results in terms of chip area. We try to apply this result by using 4-LUTs, which are also found in commercial FPGAs such as the Altera Flex 10K, and the Xilinx Virtex. [6]. Xilinx 4000 Family was a popular first generation FPGA device family with 2,000 to 180,000 usable gates but Xilinx Virtex FPGAs, each CLB now contains four circuits similar to the earlier 4000 CLB in which interconnection network contains varying length row, column, and neighbouring CLB interconnect structures that increases the logical area utilization for large fan in circuits [7]. Multicontext programming bits, a scheme that promises some savings in area efficiency and reconfiguration time for FPGAs proposed by E. Tau in their research [8]. In the research of Altera industry, they has recently introduced the new series of field programmable devices (FPDs) known as APEX (Advanced Programmable Embedded Matrix). Their main characteristic is the combination of LUTs and PLA like blocks on the same chip. APEX architecture contains embedded system blocks that can be configured to support pterms, memory blocks and content

addressable memory cells (CAMs).The first APEX devices will offer 500,000 gates, but in future, they plan to include some more devices up to 2 million programmable gates [9]. In the study of J. He and J. Rose, called Heterogeneous FPGAs, they investigated FPGA architectures with logic blocks of two different sizes to see the effects on area efficiency of LUT-based FPGAs, in the same chip. In the result, they provide a saving of 15% in chip area by mixture of LUTs [10]. Most of the research focused on FPGAs based FPDs rather than CPLD based FPDs. There are very small effort has been published in the area of CPLDs research. Though, in the study of J. L. Kouloheris and A. El Gamal, they investigated and built FPDs using PLA based logic blocks. According to authors, an FPD based PLAs with 10 inputs, 12 Pterms, and 3 outputs achieves about the same level of logic density as FPGAs based on 4-LUTs, but this unchanging share decreases the flexibility of FPGA. We are not aware of any industrial product that is based on such FPDs based PLAs. [11]. S. Wilton et. al. explain the memory modules with variable aspect-ratio that could be incorporated as separate blocks in an FPGA. This design is not orthogonal to the Hybrid FPGA, and so memory blocks could also be included in our architecture for delay minimization [12]. Li- Guang investigated Hybrid FPGA architecture, which modify the CBs which implement the AND plane of PLA. They found that a mixture of LUTs and PLAs provide and area saving of about 46% by using flowMap but did not explain about computational delay [13]. Jason found that Flowmap technology mapping algorithm reduces the maximum no of LUT for delay optimization [4].

In the Altera data sheet, implementations of partial logic circuits with PLAs are also founded in commercial FPGAs. An Embedded System Block (ESB) of Altera APEX20K can be configured as a PLA with 32 inputs, 32 product terms(ptersms) and 16 outputs [10]. In the same way, in every Xilinx Virtex II slice has a devoted OR gate named ORCY, with which a Configurable Logic Block (CLB) can implement 2 product terms with 16 inputs [11]. These architectures benefit to the logic density of FPGA. However, they are not optimized specifically for implementing PLA. Microelectronics Center of North Carolina (MCNC) benchmark suite is used as logic synthesis and optimization benchmark. The benchmark suite has standardized libraries with representative circuit designs ranging from simple circuits to advanced circuits. MCNC benchmarks are very popular in academic research. The FPGA researcher relies a lot on benchmarks to evaluate

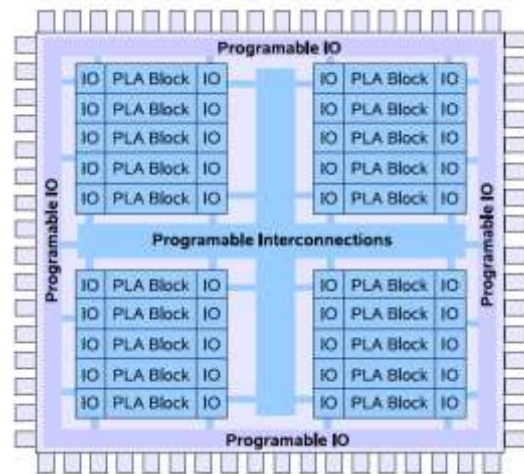
performance of their hardware and software solutions. Hence, standard and fair benchmarking practices are essential to calculate FPGA architecture, design, configuration, verification and validation of FPGA device and verify their potential to support target applications. In this paper, We try to use some MCNC bench mark to evaluate the performance technology mapping algorithms for new HRCA.

3. Architecture of CPLD/FPGAs:

The two main types of programmable devices, field programmable gate arrays (FPGA) and complex programmable logic devices (CPLD), are both extensively used. Each device contributes particular strength in the development of reconfigurable system. FPGAs programmed with static RAM technology are usually based on lookup tables. A look up tables (LUT) is a group of memory cell, which contain all possible results of a given function for a given set of input values. An n-input LUT can be used to implement up to 2^{2^n} different functions, each of which can take 2^n possible values[14].

In FPGA, an LUT physically consists of a set of SRAM cells to store the value and decoder that is used to access the correct SRAM location to retrieve the result of functions. Their main strengths are very high logic capacity in the range of hundreds of thousands of equivalent logic gates and good speed performance up to 20-50 MHz system clock rates. SRAM based LUT is used in the most commercial FPGA as function generator[15].

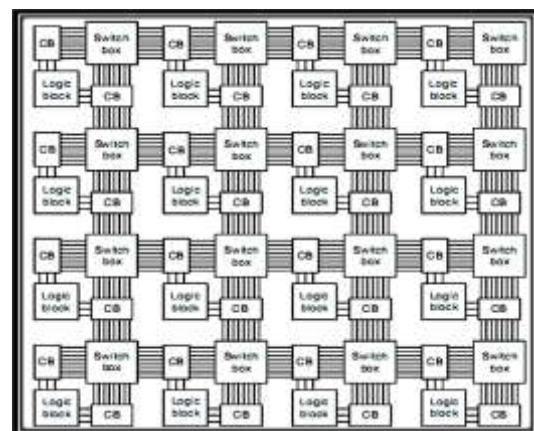
On the other hand, CPLDs consist of a set of macro cells, input/output blocks and an interconnection network. A macro A macro cell typically contains several PLAs and flip flops. Programmable logic arrays (PLA) consist of a plane of AND-gates connected to a plane of OR-gates and both the plane can be programmed by the user.



a) CPLD Device

Their characteristics include medium capacity, in the range of a few thousand gates, and ultrahigh speed performance, sometimes in excess of a 140-180 MHz system clock rate[15]. HRCA merge the two common technologies used in programmable logic devices: lookup table (LUT) based FPGA and PLA – like logic cell based CPLD. The most important initiative is to find out what function are suitable to be implemented on which logic resources[16].

Logic resources include LUTs as well as product term (pterm) like PLA logic cells. In this paper, we try to analysis the computational performance of HRCA in comparison with an architecture containing only LUTs. It indicates that the new architecture offers significant savings in computational delay as well as total chip area. Also, the HRCA can reduce the depth of circuits implemented in the FPGA, which may provide improvements in data communication speed and performance.



b) FPGA Device

Fig. 1: Structure of programmable logic devices

4. Hybrid Architecture(HRCA)

Reconfigurable computing systems regularly show remarkable performance and strength in the term of high speed, reduced energy and power consumption. The advances in high-performance computing and reconfigurable computing, based on field programmable gate arrays (FPGAs), form the basis for a new paradigm, called reconfigurable supercomputing. This can be achieved through hybrid of LUTs and PLAs of programmable logic devices.

FPGA programmed with SRAM technology are usually based on Look-Up Tables (LUTs). For implementing random logic circuits in LUT based FPGA, the cost of LUTs increases exponentially according to the inputs of circuits. So LUT is suitable for **low fan-in logic circuits**. In CPLDs are based on Programmable Logic Arrays (PLAs). The PLA usually have tens of inputs and is appropriate for **high fan-in logic circuits**. CPLDs are typically faster and have more predictable timing than FPGAs because FPGAs are generally more dense and contain more flip flops and registers than CPLDs[16].

As in most of the applications, due to fine granularity of FPGA, most of the Configuration Box's (CB) are never used but many Logic Blocks (LB) are used to implement logic functions. So a large percentage of chip area is wasted. To rectify above draw backs, it has been tried to extend a new structure of connection Box (CB) which will facilitate to work in LUTs mode or PLAs mode for algorithm computation depend upon circuit fan-in.

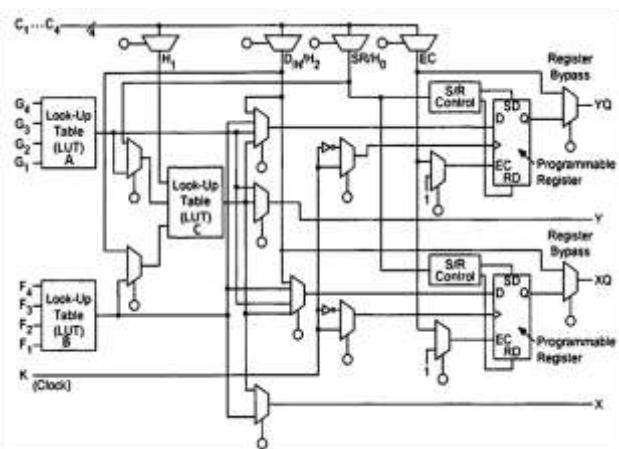


Fig. 2: Slice Structure of Configuration Logic Block

The basic building block of the Xilinx Virtex-E Configurable Logic Blocks (CLB) is the logic cell (LC). An LC includes a 4-input function generator, carry logic, and a storage element. The output from the function generator in each LC drives both the CLB output and the D input of the flip-flop. Each Virtex-E CLB contains four LCs, organized in two similar slices, the slice structure given in Fig 2. A devoted LUT C was designed to implement the OR plane of PLA and CBs of FPGA are engaged to implement the AND plane of PLA.

5. Architecture of Connection Box (CB) in HRCA

The HRCA architecture is based on the symmetrical FPGA, which utilizes the existing connection box (CB) for implementing logic function. The HRCA connection box (CB) operates in two mode, PLA mode and LUT mode. If channel width is n then we have use (n-1) 2X1 multiplexer to give the required input for PLA in CB. The selection line of each MUX is controlled by SRAM. The SRAM bit decides which input should give to PLA i.e regular input or inverse input from channel width. Let keep the SRAM bit "0" for inverse input to MUX and SRAM bit "1" for regular input to MUX. In LUT mode of CB, the channel data simply routed through the CB with regular input of the MUX to other logic cluster so corresponding selection SRAM bit of all MUX set to logic "1". On the other hand, in PLA mode of CB we need both inputs, so section SRAM bit of all MUX are depend upon the circuit which would be implement according to AND plane of PLA to produce p- term that to pass LUTs to realize the OR plane to give the final output in PLA mode.

6. Analysis of Technology Mapping Algorithm for logic optimization over MCNC circuit

In order to show how good the logic gate optimization into K-Input LUT (K= 4) circuits. We evaluate the extensive comparison of technology mapping algorithms over 20 MCNC benchmark circuits[17] from a common application ported on the given circuit by using the following algorithms, chortle-d, mispga-delay, FlowMap, and DAG-Map. The performance of an FPGA circuit is determined by two factors: the no of K-LUT's used in computation and the delay in the interconnection paths between logic blocks. In an LUT-based FPGA chip, the basic programmable logic block is a K-input lookup table (K-

LUT) which can implement any Boolean function of up to K variables. The technology mapping problem in LUT-based FPGA designs is to cover a general Boolean network (obtained by technology independent synthesis) using K-LUT's to obtain a functionally equivalent K-LUT network[16]. According to the objectives, The K-LUT-based FPGA mapping algorithms can be roughly divided into three classes

- **Area optimization:** Mapping algorithms minimize the number of LUTs used to implement the given circuit based on the assumption that the number of LUTs in the FPGA design is a good measurement of the logic area of FPGA implementations.
- **Performance optimization:** Mapping algorithms minimize the circuit delay time of the specified design. Because the propagation delay for every LUT

is almost identical, the most popular delay model used in FPGA synthesis is the unit delay model. That is, the circuit delay is estimated by the maximum level of LUTs in the synthesized circuit.

- **Rout-ability optimization:** Mapping algorithms maximize the rout-ability for easy placement and routing for the FPGA architectures.

In this paper, we use different technology mapping algorithm, like chortle-d, mispga-delay, FlowMap, and DAG-Map, to convert gate level circuit into K-LUT (where K is no of input to LUT) for an given user circuit description and find the optimal technology mapping algorithm in term of minimum number of LUTs equivalent to gate level circuit. According the given figure FlowMap is the optimal solution for technology mapping.

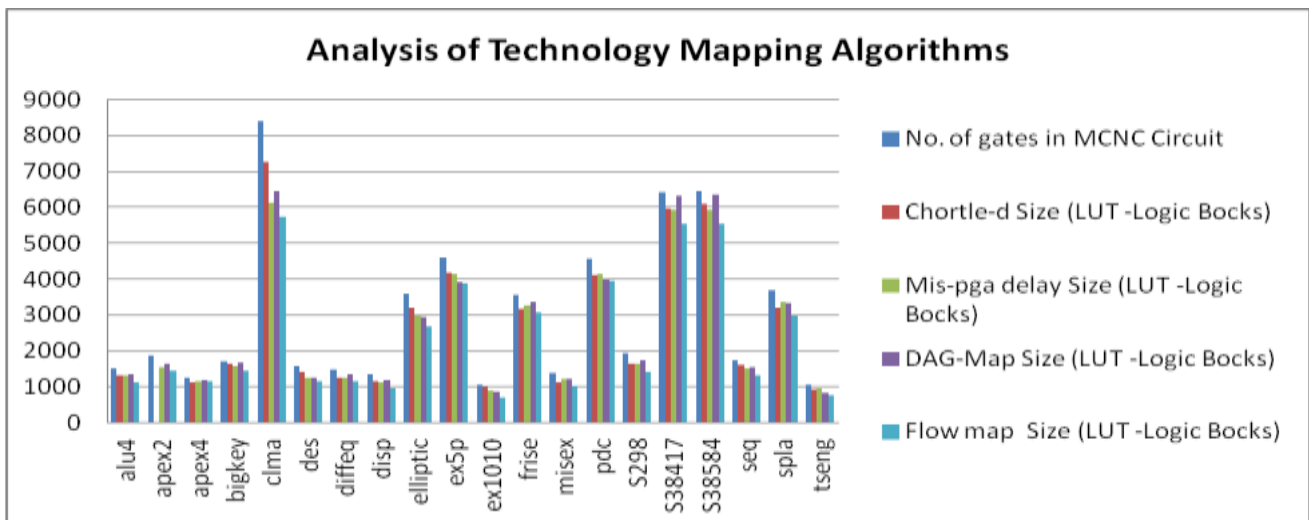


Fig 3: Comparative analysis of Technology mapping Algorithm

7. Conclusion

We have reviewed the factors which determine the logic performance of hybrid architecture HRCA over traditional FPGA. We determined the role of technology mapping algorithm in logic optimization. But according to the result, FlowMap technology mapping algorithm will give the optimal solution among all popular mapping algorithm. We use various MCNC benchmark circuits and reviewed other factors also, namely Speed-up and Power. We also discuss the modified CB architecture of our proposed HRCA architecture. In our experimental results, we notice that HRCA can provide not only excellent in logic optimization, but it also in delay optimization. In future, we would be evaluating the runtime area with delay optimization and performance

approach of our proposed HRCA with traditional FPGA using MCNC benchmarks circuit. MCNC benchmarks circuit are very popular in academic research.

References:

1. R. J. Francis et al, "Chortle: A technology mapping program for LUT-based FPGA," in ACM-IEEE Design Automation Conf., 1990, pp. 613-619.
2. K. Karplus. "Xmap: A Technology mapper for LUT-FPGA," in ACM-IEEE Design Automation Conf, 1991, pp. 240-243.
3. R. Murgai, et al., "Logic synthesis algorithms for PLA," in ACM-IEEE Design Automation Conf., 1990, pp. 620-625.

4. Jason Cong et.al, FlowMap: An Optimal Technology Mapping Algorithm for Delay optimization in LUT based FPGA Designs, IEEE Transaction on CAD, vol. 13, No. 1. 1994.
5. Altera Stratix Device Handbook (www.altera.com),2005
6. J. Rose et al., "The Effect of Logic Block Functionality on Area Efficiency," IEEE, Vol. 25, No. 5, 1990, pp. 1217-1225.
7. James O. Hamblen et al "Rapid prototyping of digital systems" - Quartus II ed, Springer Book, 2006
8. E. Tau et al., "A First Generation FPGA Implementation," Proc. on FPD, Montreal, 1995, pp. 138-143
9. A. Stansfield et al, "The Design of a New FPGA Architecture," Proc. on Field- Programmable Logic and Applications, Univ. of Oxford, London, 1995
10. J. He et al, "Advantages of Heterogeneous Logic Block Architectures for FPGAs," Proc. In Integrated Circuits Conference, San Diego, 1993.
11. J. L. Kouloheris and A. El Gamal, "PLA-based FPGA Area vs. Cell Granularity," Proceedings of the Custom Integrated Circuits Conference, 1992.
12. S. Wilton, et al, "Architecture of Centralized Field-Configurable Memory," 3rd ACM proc. on FPGA, Monterey Bay, CA,1995.
13. Li-Guang et . al, A Novel Hybrid FPGA Architecture, IEEE, 2006.
14. Christophe Bobda, Intro.mto Reconfigurable Computing Arch., Algo., and Appl., University of Kaiserslautern, Germany, Springer Book, 2007
15. Bob Zeidman, Designing with FPGAs and CPLDs, CMP Books Canada, 2002.
16. Sunil Kr. Singh et al" CAD for Automatic Logic Density Utilization of Symmetrical FPGA Architecture through Hybrid LUTs/PLA, VLSI ICIAICT,GBU, Gr. Noida, India, 2012.
17. S. Yang, "Logic Synthesis and Optimization Benchmarks, Version 3.0," Tech. Report, Microelectronics Center of North Carolina, 1991.