Voltage-mode Quaternary FPGAs: An Evaluation of Interconnections

Cristiano Lazzari, Paulo Flores, José Monteiro
ALGOS group at INESC-ID
Lisbon, Portugal
Email: {lazzari,pff,jcm}@inesc-id.pt

Luigi Carro
Institute of Informatics - UFRGS
Porto Alegre, Brazil
Email: carro@inf.ufrgs.br

***Abstract*—This work presents a study about FPGA interconnections and evaluates their effects on voltage-mode binary and quaternary FPGA structures. FPGAs are widely used due to the fast time-to-market and reduced non-recurring engineering costs in comparison to ASIC designs. Interconnections play a crucial role in modern FPGAs, because they dominate delay, power and area. The use of multiple-valued logic allows the reduction of the number of signals in the circuit, hence providing a mean to effectively curtail the impact of interconnections. The most important characteristic of the results are the reduced fanout, fewer number of wires and the smaller wire length presented by the quaternary devices. We use a set of arithmetic circuits to compare binary and quaternary implementations. This work presents the first step on developing quaternary circuits by mapping any binary random logic onto quaternary devices.**

I. Introduction

Interconnections are becoming the dominant aspect of the circuit delay for state-of-the-art circuits due to the advent of deep sub-micron technologies (DSM). This fact is even more significant with each new technology generation [1]. In DSM technologies, the gate speed and density scaling follows Moore's law. On the other hand, the interconnection resistance-capacitance product increases in each technology node, leading to an increase of network delay.

Interconnections play an even more crucial role in Field Programmable Gate Arrays (FPGA), because they not only dominate the delay, but they also have aggressive impact on power consumption [2] and area [3].

Multiple-valued logic (MVL) has received increased attention in the last decades because of the possibility to represent the information with more than two discrete levels. Representing data using a MVL system is more effective than the binary-based representation, because the number of interconnections can be significantly reduced, with major impact in all design parameters: less area dedicated to interconnections; more compact and shorter interconnections, leading to increased performance; lower interconnect switched capacitance, and hence lower global power dissipation [4].

MVL has been successfully accomplished in flash memories [5], for example, where a single memory cell can hold different logic values. Some combinational circuits such as adders [6] and multipliers [7], as well as programmable devices [8] were also proposed.

The main drawback of these previous systems is that they are based on current-mode devices. These circuits present successful improvements in reducing area, but their excessive power consumption and implementation complexities has prevented, until now, MVL systems from being a viable alternative to standard CMOS designs.

Recently, a technique to support voltage-mode MVL circuits was proposed in [9]. This technique uses a standard CMOS process that deals specifically with the power dissipation problem, and still maintains the logic compaction allowed by MVL. The proposed circuits intend to reduce the number of interconnections present in existing binary-based systems, without incurring on power consumption penalties.

The benefits of this new MVL implementation technique were considered for application in the reconfigurable domain. A new lookup table (LUT) structure was proposed in [10] where the information is represented by quaternary values. A new quaternary logic cell was presented and results demonstrate interesting area and power reductions in comparison to equivalent binary structures.

In this work we show the first approach to tackle the challenge of low power and high density FPGAs, by evaluating the impact of the interconnections in binary and quaternary LUTs. By using quaternary connections one is able to reduce the number of wires and switches, thus reducing area and power consumption of FPGAs.

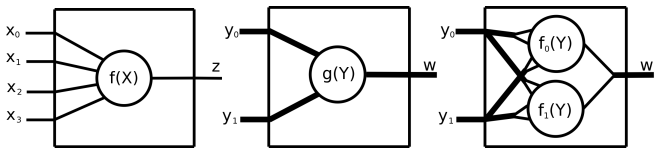
We have used a set of arithmetic circuits as case study to validate our experiments due to the possibility to manually generate, place and route them.

This paper is organized as follows. Section II discusses the differences between binary and quaternary implementations of lookup tables. Section III presents details about the modeling of interconnections. Section IV shows how the arithmetic circuits were implemented. Experimental results are presented in Section V. Finally, Section VI concludes the paper and outlines future works.

II. Binary and Quaternary LUTs

General Lookup Tables (LUT) are basically memories, which implement a given logic function. Values are initially stored in the lookup table structure, and once inputs are applied, the logic value in the addressed position is assigned to the output. The capacity of a LUT is given by

$$|C| = n \times b^k \tag{1}$$



(a) 4-input BLUT. (b) 2-input QLUT. (c) Equivalent QLUT using binary functions.

Fig. 1: Binary (BLUT) and quaternary (QLUT) lookup tables and the quaternary function.

where n is the number of outputs, k is the number of inputs and b is the number of logic values. For example, a 4-input binary lookup table with one output is able to store $1 \times 2^4 = 16$ Boolean values. For the purpose of this work, only 1-output LUTs ($n = 1$) are discussed in this paper.

A. Preliminaries

A binary function implemented by a Binary Lookup Table (BLUT) is defined as f: $\mathbf{B}^k \rightarrow \mathbf{B}$, over a set of variables $X = (x_0, \cdots, x_i, \cdots, x_{k-1})$, where each variable x_i represents a Boolean value. The total number of different functions $|F|$ that can be implemented in a BLUT with k input variables is given by

$$|F| = b^{|C|} \tag{2}$$

where $b = |B|$ (i.e. $b = 2$ in the binary case). Figure 1(a) illustrates a binary function where $k = 4$. Thus, a lookup table with 4 inputs can implement one of $|F| = 65,536$ different binary functions.

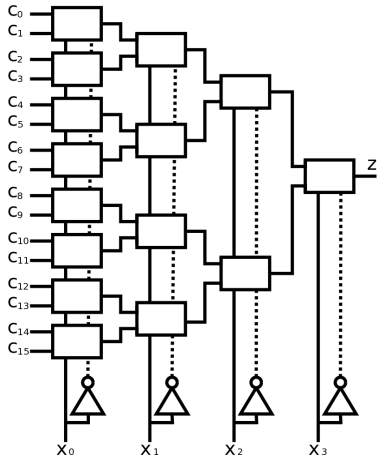
Quaternary functions are basically generalizations of binary functions. A quaternary function implemented by a quaternary lookup table (QLUT) is defined as g: $\mathbf{Q}^k \rightarrow \mathbf{Q}$, over a set of quaternary variables $Y = (y_0, \cdots, y_i, \cdots, y_{k-1})$, where the values of a variable y_i, as the values of the function $g(Y)$, can be in $\mathbf{Q}$= $\{0, 1, 2, 3\}$. As in the binary case, the number of possible function in QLUTs is given by (2), where $b = 4$. In the case of QLUTs 2 inputs, that have the same capacity of 4-input BLUTs, the number of quaternary functions that can be represented is about 4.29×10^9 for a QLUT with only two inputs, which is much larger than the BLUT. Figure 1(b) illustrates a 2-input quaternary function implemented in a QLUT.

Note that the function $g(Y)$ performs exactly the same function as the two binary BLUTs, $f_0(Y)$ and $f_1(Y)$, as depicted in Figure 1(c), where f_0 represents the least significant Boolean values and f_1 represents the most significant ones.

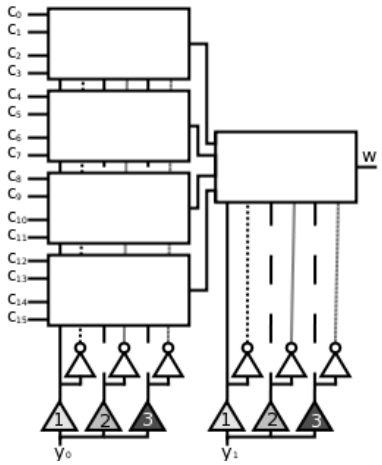
Since a quaternary variable y is capable of representing twice as much information as a binary variable x, we consider the cardinality of $|Q| = 2 \times |B|$ in our experiments. In other words, we assume that two binary variables can be grouped in order to represent a quaternary variable. Such grouping aims at reducing the total number of connections and the number of gates as well.

B. Lookup Tables Implementation

In this work, binary and quaternary lookup tables were implemented by a set of multiplexers, such as presented in [10] and illustrated in Figure 2.



(a) 4-input BLUT.



(b) 2-input QLUT.

Fig. 2: Binary and quaternary lookup tables implementation.

Figure 2(a) shows a binary 4-BLUT implementation ($b = 2, |X| = k = 4, |C| = 16$) where $x_i \in X$ are the inputs, $c_i \in C$ form the lookup table configuration and z is the output. The BLUT is composed of four stages as a consequence of the number of inputs. Multiplexers are responsible for propagating configuration values to the BLUT output. The multiplexers are composed of pass gates, which receive selection signals from the four BLUT inputs and associated inverters.

A quaternary lookup table (QLUT) follows the same structure as the BLUTs. However, Down Literal Circuits (DLCs) structures determine which configuration value must be propagated to the output [9]. Figure 2(b) illustrates the implementation of a 2-input QLUT ($b = 4, |Y| = k = 2, |C| = 16$). As in the binary case, $c_i \in C$ are the lookup table configuration values, $y_i \in Y$ are the inputs and w is the output. Due to the quaternary representation, each multiplexer has four configuration inputs, therefore only two multiplexer stages are required.

The DLCs (gray triangles $1, 2$ and 3 in Figure 2(b)) have structures similar to inverters (with 1 PMOS and 1 NMOS transistor). Transistors in each DLC circuit have modified

TABLE I: DLCs behavior according to the input value.

Input	1 (light gray)	2 (gray)	3 (dark gray)
0	3	3	3
1	0	3	3
2	0	0	3
3	0	0	0

V_{th} values in order to allow the switching at different input voltages. This way, the 3 DLC circuits work as a thermometer system. For each DLC the output values are '0' (GND) or '3' (V_{DD}), but the switching occurs on different voltage levels, according to the logic value applied to their inputs. Table I shows the DLC output logic values as function of the inputs.

Transistors used in the implementation of the DLCs present different threshold voltages (V_{th}), to allow the desired behavior. It is important to highlight that standard CMOS technology is used in the whole QLUT. Only the DLC structures are composed of 6 transistors with different threshold voltages (3 PMOS and 3 NMOS). The quaternary multiplexers are composed of transistors with the same V_{th} than the ones used in the binary multiplexers.

III. Interconnections

A general FPGA structure is composed by a fully programmable network connecting Configurable Logic Blocks (CLB), IOs, and other FPGA components. In order to increase the efficacy of the FPGA routing, different types of interconnections are implemented. For example, Xilinx Spartan-3 FPGAs [11] presents four types of interconnects: long lines, hex lines, double lines and direct lines.

A technique for modeling and analysis of FPGA interconnects are presented in [12], [13]. Using these techniques, we modeled the FPGA interconnections as distributed RC networks using the Predictive Technology Model (PTM) parameters [14] for a $45nm$ technology.

A. Impact of interconnects in the FPGA performance

One important drawback of the quaternary LUT, when compared with the binary LUT, is the increased propagation delay. Nevertheless, as quaternary LUTs do provide smaller wires due to the reduction in the number of lookup tables used to implement circuit functions, there are some margin for some performance gains. For this reason, we performed an experiment where we connected two BLUTs and two QLUTs, and evaluated the obtained delay.

The goal of this experiment was to detect the point where the delay in quaternary devices outperforms the binary devices, due to the reduction in the connection lengths. Interconnections were modeled as simple RC networks.

Figure 3 shows the results obtained from this experiment where the normalized delay of the binary circuit in comparison with the delay of quaternary circuit for different wire length reduction. The quaternary implementation obtains gains over the binary one when the wire length connection of the quaternary device is smaller than $\sim$80% of the binary connection. In other words, results show delay gains in the quaternary

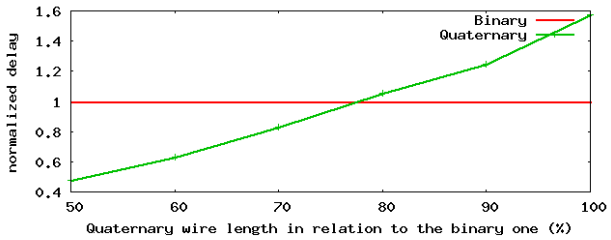


Fig. 3: Effects of wire lengths on the delay of the quaternary devices.

implementation, when the wire length is at least 22% smaller than the binary implementation.

Fanout is another relevant point when determining the delay of a device because it directly affects the load capacitance. The fanout is also expected to be smaller in the quaternary circuits due to the inherent information compaction present in the quaternary logic.

B. Tile Size

In order to determine a more realistic interconnection size, we have to define the FPGA tile size ($i.e.$, size of CLB + interconnection matrix). We defined the tile size by taking into account the work presented in [13], and tailoring it to a $45nm$ technology [14]. In our experiments we use a tile with $80\mu m$ of side. Since we do not have yet layout level implementations and the number of MOS devices is basically the same for both implementations, we will be assuming the same tile size for binary and quaternary devices. Note that considering identical tile sizes we are able to evaluate the effects of the interconnections on the FPGA performance with more accuracy.

IV. Arithmetic Circuits as a Case Study

Logic blocks in modern FPGAs are not composed only of LUTs. Additional logic to accelerate the carry propagation and multiplier structures are also present in order to improve the efficacy of FPGAs. In our experiments, CLBs are considered as composed only by LUTs, and they are connected to each other through the optimal set the previous mentioned types of interconnect lines.

We choose arithmetic circuits in our experiments because of the possibility to manually design the quaternary circuits. These circuits allow the evaluation of the interconnection effects in binary and quaternary implementations.

A. Synthesis & Mapping

For binary logic, the synthesis of the arithmetic circuits was performed by describing the circuits in BLIF format and using the Berkeley ABC tool [15] to perform the synthesis and technology mapping.

For the quaternary logic, the circuits were manually generated based on previous binary implementations, so that 4-input binary functions were grouped in pairs in such a way that 2-input quaternary functions are generated. Note that, as far as we know, there are no available tools able to map circuits in quaternary FPGAs.

TABLE II: Experimental results of some arithmetic circuits in the binary and quaternary FPGAs.

Ckt	Binary					Quaternary					Gain (%)			
	Area	Power	M. Freq.	WL	Fanout	Area	Power	M. Freq.	WL	Fanout	Area	Power	M. Freq.	WL
FA16	0.23	33.1	10.5	102	2	0.10	14.1	12.6	80	1	56.5	57.4	16.7	21.6
FA32	0.41	48.3	5.3	102	2	0.23	18.3	6.3	80	1	43.9	62.1	15.9	21.6
FA64	0.92	71.4	2.4	106	2	0.41	25.3	3.1	80	1	55.4	64.6	22.3	24.5
Average gain for the full adders											*51.9*	*61.3*	*18.3*	*22.5*
MUL4	0.23	14.0	8.57	351	3.4	0.16	9.9	16.21	166	2	30.4	29.3	47.1	52.7

Units are Area in μm^2, Power in μW, Maximum frequency (M. Freq.) in MHz and Average wire length (WL) in μm.

B. Placement and Routing (P&R)

The placement of the LUTs was manually performed in the FPGA structure, a procedure that is relatively straightforward for arithmetic circuits given its regular structure. The routing was accomplished through a greedy algorithm that evaluates the best way to connect the LUTs. After the routing, connections are converted to RC networks, as explained in Section III.

V. Experimental Results

Table II shows the experimental results, obtained from the comparison between binary and quaternary circuits. Our experiments were realized using full adders (FA) with 16, 32 and 64 bits, and a 4-bit multiplier. Once the circuits were generated, the P&R was performed as explained in Section IV-B. Power consumption and delay were obtained through Cadence UltraSim [16] simulation. Maximum frequency was obtained through the analysis of the critical path and power consumption is the result of 512 random generated inputs vectors.

Results show an important reduction on power consumption of 61.3% for full adders and 29.3% for the multiplier, for the quaternary circuits when compared to their binary counterparts. The frequency was increased by 18.3% in the implementation of adders in the quaternary FPGA, but near 50% in the multiplier. While the critical path is significantly smaller for the quaternary multiplier, the power consumption reduction is not so significant as in the full adders because we are not very effective in grouping binary signals to form quaternary ones, as in the full adders.

These interesting gains presented in the results are explained by the fact that both wire length and fanout were significantly smaller in the quaternary circuits when compared to the binary circuits. The average wire length (WL) is 22.5% smaller in the full adders and 52.7% in the multiplier. Furthermore, the average fanout in the multiplier reduced from 3.4 to 2 gates, which implies a smaller load capacitance to the QLUTs.

VI. Conclusion

This work presents important advances on the development of multi-valued circuits through the evaluation of the interconnections in binary and quaternary implementations of FPGAs for a $45nm$ process technology. Results show that a quaternary FPGA may be competitive with the binary one due to the important reductions on the connection sizes and its effects on the power consumption and circuit performance.

It is important to highlight that this work presents the first approach to develop competitive quaternary circuits, in which the application of arithmetic circuits is taken as a case study. Experiment results have successfully shown that significant power reduction and performance improvement can be achieved when a quaternary FPGA is used.

We believe that the quaternary representation applied to the random logic will allow not only the number and size of the connections, but most important, the reduction of the fanout and consequently the load applied to the logic blocks. For this reason, we are developing logic synthesis and technology mapping algorithms and tools focused on quaternary representation.

References

[1] A. K. Gupta and W. J. Dally, "Topology optimization of interconnection networks," *IEEE Comput. Archit. Lett.*, vol. 5, no. 1, p. 3, 2006.

[2] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, pp. 1712–1724, Nov. 2005.

[3] A. Singh and M. Marek-Sadowska, "Efficient circuit clustering for area and power reduction in FPGAs," in *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*. New York, NY, USA: ACM, 2002, pp. 59–66.

[4] E. Dubrova, "Multiple-valued logic in VLSI: challenges and opportunities," in *Proceedings of NORCHIP'99*, 1999, pp. 340–350.

[5] T.-S. Jung, Y.-J. Choi, K.-D. Suh, B.-H. Suh, J.-K. Kim, Y.-H. Lim, Y.-N. Koh, J.-W. Park, K.-J. Lee, J.-H. Park, K.-T. Park, J.-R. Kim, J.-H. Yi, and H.-K. Lim, "A 117-mm2 3.3-v only 128-mb multilevel NAND flash memory for mass storage applications," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1575–1583, Nov 1996.

[6] A. Gonzalez and P. Mazumder, "Multiple-valued signed digit adder using negative differential resistance devices," *IEEE Transactions on Computers*, vol. 47, no. 9, pp. 947–959, Sep 1998.

[7] T. Hanyu and M. Kameyama, "A 200 MHz pipelined multiplier using 1.5 v-supply multiple-valued mos current-mode circuits with dual-rail source-coupled logic," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 11, pp. 1239–1245, Nov 1995.

[8] Z. Zilic and Z. Vranesic, "Multiple-valued logic in FPGAs," in *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, Aug 1993, pp. 1553–1556 vol.2.

[9] R. Cunha, H. Boudinov, and L. Carro, "A novel voltage-mode CMOS quaternary logic design," *IEEE Transactions on Electron Devices*, vol. 53, no. 6, pp. 1480–1483, June 2006.

[10] R. da Silva, C. Lazzari, H. Boudinov, and L. Carro, "CMOS voltage-mode quaternary look-up tables for multi-valued FPGAs," *Microelectronics Journal*, vol. 40, no. 10, pp. 1466 – 1470, 2009.

[11] Xilinx Inc., "Spartan-3 FPGA family data sheet," 2008. [Online]. Available: http://www.xilinx.com/support

[12] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSIs," *Electron Devices, IEEE Transactions on*, vol. 40, no. 1, pp. 118–124, Jan 1993.

[13] T. Mak, C. D'Alessandro, P. Sedcole, P. Y. K. Cheung, A. Yakovlev, and W. Luk, "Global interconnections in FPGAs: modeling and performance analysis," in *SLIP '08: Proceedings of the 2008 international workshop on System level interconnect prediction*. New York, NY, USA: ACM, 2008, pp. 51–58.

[14] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," *International Symposium on Quality Electronic Design*, pp. 585–590, 2006.

[15] Berkeley Logic Synthesis and Verification Group, "ABC: A system for sequential synthesis and verification," 2009. [Online]. Available: http://www.eecs.berkeley.edu/∼alanmi/abc/

[16] Cadence Design Systems Inc., "Virtuoso ultrasim simulator user guide," 2009. [Online]. Available: http://www.cadence.com