

# A New Quaternary FPGA Based on a Voltage-mode Multi-valued Circuit

Cristiano Lazzari  
INESC-ID  
Lisbon, Portugal  
Email: lazzari@inesc-id.pt

Paulo Flores, José Monteiro  
INESC-ID / IST, TU Lisbon  
Lisbon, Portugal  
Email: {pff,jcm}@inesc-id.pt

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

**Abstract**—FPGA structures are widely used due to early 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. Multiple-valued logic allows the reduction of the number of signals in the circuit, hence can serve as a mean to effectively curtail the impact of interconnections. In this work we propose a new FPGA structure based on a low-power quaternary voltage-mode device. The most important characteristics of the proposed architecture are the reduced fanout, low number of wires and switches, and the small wire length. We use a set of FIR filters as a demonstrator of the benefits of the quaternary representation in FPGAs. Results show a significant reduction on power consumption with small timing penalties.

## I. INTRODUCTION

The large number of components in modern systems on chip (SoCs) presents new challenges to designers. The high integration of different systems increases the number and length of interconnections, hence the overall complexity involving the connections of these systems.

Moreover, 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 becoming even more significant with each new technology generation [1]. In DSM technologies, the gate speed, density and power scaling follows Moore's law. On the other hand, the interconnection resistance-capacitance product increases with the technology node, leading to an increase of network delay. Even after modifications in interconnections, from aluminum to copper and low-k inter metal dielectric materials, the problem remains and it is getting more significant [2].

Interconnections play an even more crucial role in Field Programmable Gate Arrays (FPGA), because they not only dominate the delay, but they also aggressive impact power consumption [3] and area [4]. Recent works suggest that in modern million-gates FPGAs, as much as 90% of chip area is dedicated to interconnections [5], because of the large number of wires and switches to select among them.

For FPGAs to reach a larger market, their excessive power dissipation must be severely reduced. Moreover, if one could reduce the FPGA area without losing logic capabilities, one could enhance the yield and reduce prices, or even increase the amount of memory available inside the FPGA. To reduce

the area of the FPGA, a reduction in the interconnection is mandatory, since interconnections take large amount of area.

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 in 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 [6].

The possibility to represent the information using MVL is not recent. MVL has been successfully accomplished in Flash memories [7], for example, where a single memory cell can hold different logic values. Some combinational circuits such as adders [8] and multipliers [9], as well as programmable devices [10] 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 voltage-mode MVL technique was proposed in [11], dealing specifically with the power dissipation problem using a standard CMOS process, and still maintaining 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 [5] 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. However, [5] only discusses the LUT, and not its application to real circuits. In a real reconfigurable device one must take into account the structure of the whole configurable logic block (CLB), not only the LUT.

In this contribution we show the first steps to tackle the challenge of low power and high density FPGAs, by proposing

a quaternary CLB. By using quaternary connections one is able to reduce the number of wires and switches, thus reducing area and power consuming part of current FPGAs. A complete arithmetic-oriented CLB is proposed, in which any logic operation can be implemented through a quaternary LUT. A fast carry look-ahead propagation unit and a register are also presented in the proposed CLB.

As case study to validate the proposed CLB we have used a digital signal processing application focusing on Finite Impulse Filters (FIR) filters. The synthesis of the filters is based on the work presented in [12]. Only adds/subtractions and shift operations are used in synthesis of the filters. We choose filters as the case study because their synthesis, placement and routing are simple tasks in the FPGA due to their regular structure. All our experiments were performed with a 45nm process technology [13].

This paper is organized as follows. Section II discuss the differences between binary and quaternary implementations of lookup tables. Section III presents the nquaternary FPGA, gives details about the new arithmetic-oriented logic block, and presents comparisons with the binary version. Section IV discusses FIR filter implementations using Multiple Constant Multiplications (MCM) as the case study adopted in this work and exemplifies how filters are deployed in the proposed FPGA structure. Experimental results are presented in Section V. Finally, Section VI concludes the paper and outlines future work.

## II. BINARY AND QUATERNARY LUTS OVERVIEW

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  $|C|$  is given by

$$|C| = n \times b^k \quad (1)$$

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  $X = (x_0, \dots, x_i, \dots, x_{k-1})$ , 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

$$|F| = b^{|C|} \quad (2)$$

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

Quaternary functions are basically generalizations of binary functions. A quaternary function implemented by a quaternary

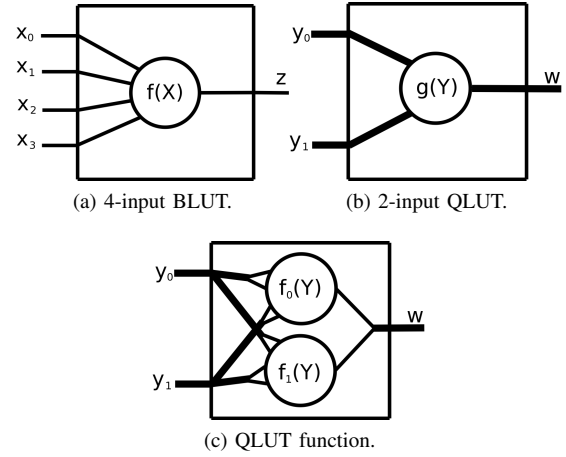


Fig. 1: Binary (BLUT) and quaternary (QLUT) lookup tables and the quaternary function.

lookup table (QLUT) is defined as  $g: \mathbf{Q}^k \rightarrow \mathbf{Q}$ , over a set of quaternary variables  $Y = (y_0, \dots, y_i, \dots, y_{k-1})$ , 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 this case, the number of functions that can be represented is around  $4 \times 10^9$  for a QLUT with only two inputs, which is much larger than the BLUT. Figure 1b 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 1c, 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 procedure aims at reducing the total number of connections and the number of gates as well.

### B. Lookup Tables Implementation

Binary and quaternary lookup tables were implemented by a set of multiplexers, such as presented in [5] and illustrated in Figure 2.

Figure 2a 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 propa-

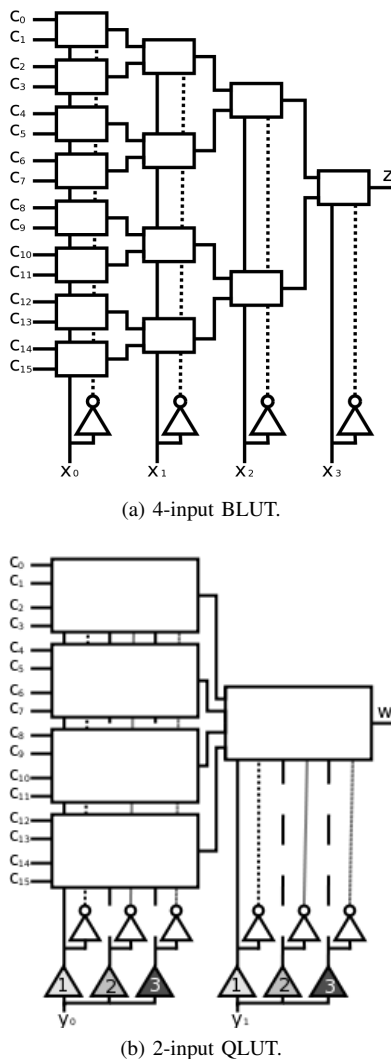


Fig. 2: Binary and quaternary lookup tables implementation.

gated to the output [11]. Figure 2b 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 2b) have structures similar to inverters (with 1 PMOS and 1 NMOS transistor). Transistors in each DLC circuit have modified  $V_{th}$  values in order to allow the switching at different input voltages. This way, the 3 DLCs circuits work as a thermometer system. The DLC output values are only '0' ( $GND$ ) or '3' ( $V_{DD}$ ), 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

TABLE I: Down literal circuits (DLCs) behavior according to the logic value at the input.

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

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 and, for this reason, DLCs produce 2-level output signals.

### III. THE QUATERNARY FPGA

Field Programmable Gate Arrays (FPGA) are widely used in commercial applications due to rapidly prototyping and reduced time-to-market in comparison with Application Specific Integrated Circuits (ASIC).

In general, FPGAs are basically sets of programmable Configurable Logic Units (CLBs) and interconnections. The CLBs contain LUTs to implement logic and storage elements [14]. CLBs in the Xilinx Spartan-3 FPGA family are composed by two independent groups of two slices. A Slice is a logic/storage unit.

The routing among logic blocks are performed through programmable switch matrices. The group of switch matrix and CLBs is called a tile.

#### A. The FPGA Logic Blocks

In this work we propose a new quaternary logic block targeting arithmetic functions. Figure 3 illustrates the structure

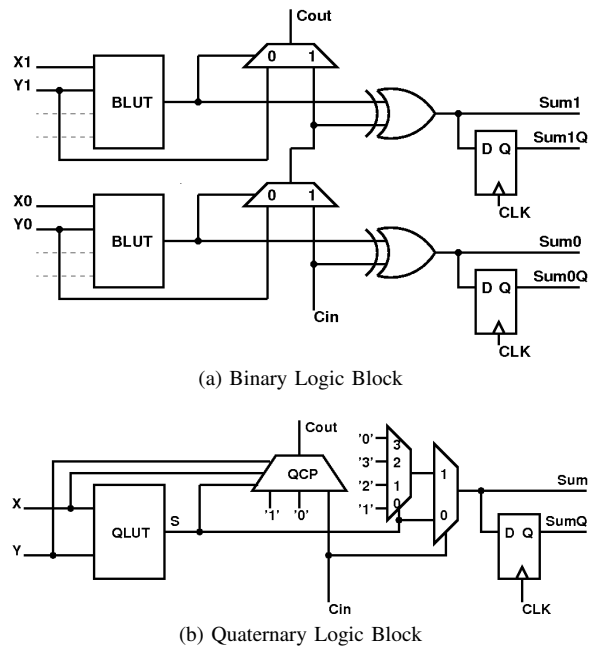


Fig. 3: Binary and Quaternary Logic Blocks

of binary and quaternary logic blocks of the FPGA configured to implement the sum of variables  $X$  and  $Y$ . The binary logic block (Figure 3a) represents two slices of the Xilinx Spartan-3 FPGAs [14].

Carry look-ahead is implemented by propagating the carry signal through two multiplexers from  $C_{in}$  to  $C_{out}$ . The carry propagation signal is defined by a XOR function implemented by the BLUT (*i.e.*  $X_i \neq Y_i$ ). Otherwise, the carry is generated as one of the inputs.

We developed the quaternary logic block following the same idea, but considering quaternary functions (Figure 3b). The QLUT implements functions of 2 variables as a generalization of the 4-input binary LUT. Table II shows the signal  $S$ , implementing the sum of  $X$  and  $Y$ , and the  $C_{out}$  as function of the inputs  $X$ ,  $Y$ ,  $S$  and  $C_{in}$ . Our QLUT implementation is based on the work proposed in [11] to the 45nm technology models presented in [13].

TABLE II: The QLUT output  $S$  and  $C_{out}$  functions

$X$	$Y$	$S$	$C_{out}$	$X$	$Y$	$S$	$C_{out}$
0	0	0	0	2	0	2	0
0	1	1	0	2	1	3	$C_{in}$
0	2	2	0	2	2	0	1
0	3	3	$C_{in}$	2	3	1	1
1	0	1	0	3	0	3	$C_{in}$
1	1	2	0	3	1	0	1
1	2	3	$C_{in}$	3	2	1	1
1	3	0	1	3	3	2	1

The carry propagation/generation in the quaternary element is defined by a modified multiplexer, in such a way that  $C_{out}$  is a function of the input signals  $X$ ,  $Y$  and the QLUT output signal as well. The Quaternary Carry Propagation (QCP) logic is illustrated in Figure 4 and implements the  $C_{out}$  function shown in Table II.

The QCP logic is divided in two parts. The first part is the carry propagation detection (*i.e.* generation of the function  $C_{out} = C_{in}$ ). Thus, the same DLC 3 used in the QLUT (Figure 2b) is used in the QLC to generate the signal  $S3$ .  $S3$  enables the propagation of the carry whenever the QLUT output  $S$  is equal '3', which implies  $S3 = '0'$ . See Table I for further details.

The carry generation is defined by  $S3 = 0$  and one of other two conditions  $K1$  and  $K2$ .  $K1$  and  $K2$  are generated by quaternary logic gates. These conditions determine  $C_{out} = '1'$

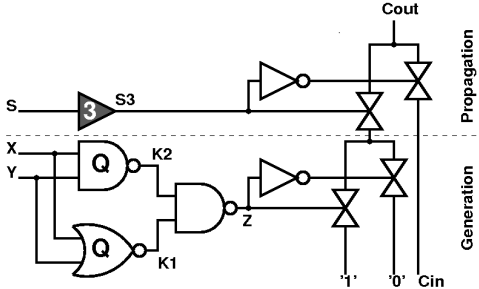


Fig. 4: Quaternary Carry Propagation (QCP) Logic

or  $C_{out} = '0'$ . First,  $K1$  defines  $C_{out} = '1'$  when  $X = '3'$  or  $Y = '3'$  and second,  $K2$  defines  $C_{out} = '1'$  when  $X \geq '2'$  and  $Y \geq '2'$ . Otherwise,  $C_{out} = '0'$ .

Note the  $Sum$  output is generated directly from the  $S$  signal if  $C_{in} = 0$ . In cases where there is a carry (*i.e.*  $C_{in} = '1'$ ), the output  $Sum$  is  $S$  incremented by '1'. This is done using quaternary multiplexers as shown in Figure 3b.

Table III shows the power consumption and the propagation delay for the binary and quaternary CLBs presented in Figure 3. The power consumption of the quaternary CLB is 20% smaller while the propagation delay is around 8.5% slower. Note that one expects to have a better power delay product characteristics in the quaternary FPGA because the quaternary representation allows reduced fanout, small bus width and, consequently, reduced wire length.

TABLE III: Power consumption and propagation delay for the Binary and quaternary CLBs.

	Power ( $\mu W$ )	Delay ( $ps$ )
Binary	2.4	836
Quaternary	1.9	913

A D-type Flipflop (FF) is also presented in the quaternary logic block. The FF is composed by quaternary inverters. Table IV shows the setup time  $T_{setup}$  and the  $CLK \rightarrow Q$  delay for the binary and quaternary flipflops.  $T_{setup}$  in the quaternary FF is smaller than the binary one due to the electrical characteristics of the quaternary transistors. Otherwise,  $CLK \rightarrow Q$  delay is similar for both binary and quaternary FFs.

TABLE IV: Binary and quaternary D-FF setup time  $T_{setup}$  and  $CLK \rightarrow Q$  delay.

	$T_{setup}$ ( $ps$ )	$CLK \rightarrow Q$ ( $ps$ )
Binary	70	88
Quaternary	380	85

We refer the reader to verify the work published in [11] for further details about the behavior of the quaternary CMOS transistors and the quaternary logic gates, as well.

## B. Interconnections

As previously discussed, the FPGA structure is composed by a fully programmable network connecting CLBs, IOs, and other FPGA components. In order to increase the efficacy of the FPGA routing, four types of interconnects are present in the Xilinx Spartan-3 FPGAs [14]: long lines, hex lines, double lines and direct lines.

Modeling and analysis of FPGA interconnects are presented in [15], [16]. We model the FPGA interconnections as distributed RC networks on the Predictive Technology Model (PTM) parameters [13]. Based on the work proposed in [16], we consider two different types of wires according to the two different sets of physical parameters presented in table V.

TABLE V: Physical parameters used in the simulations.

Technology	45nm	
FPGA Tile Size	0.08mm	
Line types*	(A, B)	(C, D)
<i>Design parameters:</i>		
Wire width ( $\mu\text{m}$ )	0.12	0.2
Wire spacing ( $\mu\text{m}$ )	0.12	0.2
Line thickness ( $\mu\text{m}$ )	0.3	0.3
Line-ground spacing ( $\mu\text{m}$ )	0.3	0.3
<i>Physical parameters:</i>		
Resistance ( $\text{Ohm}/\text{mm}$ )	611.11	366.66
Capacitance ( $\text{fF}/\text{mm}$ )	329.91	244.18

\*A) Direct lines, B) Double lines, C) Hex lines and D) Long lines.

### C. Tile Size

The size of the tile (*i.e.* CLB + switch matrix) was defined taking into account the work presented in [16], and scaling to a 45nm technology. In respect to CLB sizes, we have a binary CLB with 180 transistors, considering the whole structure, and 166 transistors to the quaternary version. Anyway, considering we do not work at layout level and, as a consequence, design rules are not being taken into account, it is reasonable to assume the same tile size for both binary and quaternary structures. Note that considering identical tile sizes we also are able to evaluate the effects of the interconnections on the FPGA performance with more accuracy.

## IV. FIR AS A CASE STUDY

In several computationally intensive operations, notably Finite Impulse Response (FIR) filters, the same input is multiplied by a set of constant coefficients. This operation is called Multiple Constant Multiplications (MCM). MCMs are commonly used in Digital Signal Processing (DSP) applications and are an important choice for reduce the power consumption due to the high level of sharing of operations and the possibility to implement multiplications by using only adders/subtractions and shifts.

For the purpose of this work, we choose filters as the case study because the synthesis, placement and routing are simple tasks in the FPGA structure due to their regular structure, and they would give us a first idea about the viability of our quaternary device.

Figure 5 illustrates the implementation of a filter with 4 taps, in which the sharing of partial terms can be verified. The input  $x$  is multiplied by the constants 117, 100, 13 and 36.

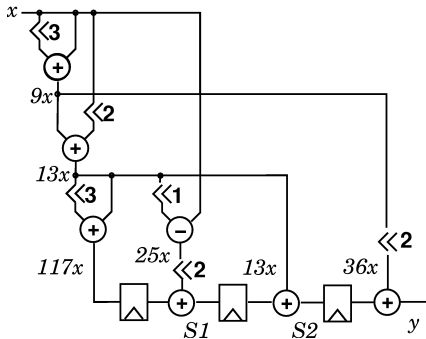


Fig. 5: An example of FIR Filter with 4 taps.

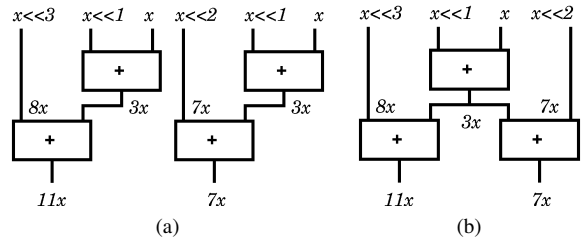


Fig. 6: Sharing partial terms for the computation of  $7x$  and  $11x$ . a) no sharing and b) sharing the partial term  $3x$ .

### A. Synthesis & Mapping

The synthesis of the filters is performed by describing synthesis problem as Integer Linear Programming based on the algorithm proposed in [12]. For each set of constant coefficients there are a wide range of possible mapping solutions, such as illustrated in Figure 6. In this example, instead of using two adders per coefficient (Figure 6a), the adder that generates the value  $3x$  is shared in order to reduce the number of adders as in Figure 6b.

### B. Placement and Routing (P&R)

The placement & routing of the filters is very simple to implement in FPGAs. Operators are placed in the CLB columns in order to take advantage of the fast carry look-ahead chain. Horizontally, CLBs are placed according to the succession of operators.

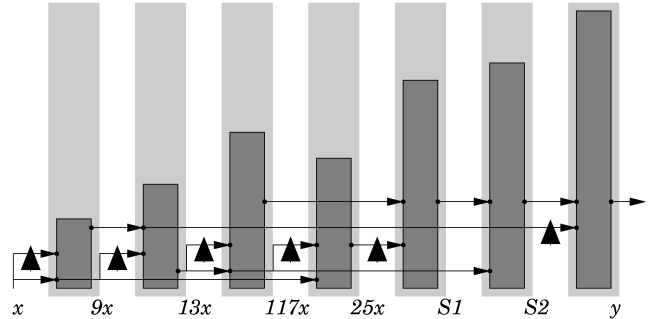


Fig. 7: Placement for the filter shown in figure 5.

Figure 7 illustrates one possible placement of the filter exemplified in Figure 5. Light gray rectangles represent the available CLB columns in the FPGA, while dark gray rectangles represent the used CLBs. Connections among operators are represented by left-to-right arrows and up arrows represent the shift of the operands. Note that the shift of operands does not use extra logic hardware, but only connections that are rearranged.

The routing of operators is done according to the distance between two connected CLBs. A greedy algorithm evaluates all the available options among the four types of lines and selects the best option according to the distance among CLBs. After routing, connections are converted to RC networks, as explained in Section III-B.

TABLE VI: Experimental results of some filters in the binary and quaternary FPGAs.

# Taps	Binary						Quaternary						Gain (%)						
	CLBs	Area	Power	Freq	WL	Sw	CLBs	Area	Power	Freq	WL	Sw	Area	Power	Freq	WL	Sw		
10	333	2.1	168.6	5.8	10.3	1,105	216	1.4	138.3	5.0	4.9	602	33.3	18.0	-13.8	52.5	45.5		
15	510	3.2	218.8	2.7	11.8	1,467	324	2.0	166.9	2.5	5.6	871	37.5	23.7	-7.4	52.5	40.6		
20	731	46.7	157.0	1.5	16.4	2,365	487	31.1	144.2	1.3	7.5	1,188	33.4	8.2	-13.3	54.2	49.8		
30	1,288	8.3	961.7	2.23	19.8	3,901	931	5.9	824.8	2.0	10.5	2,315	28.9	14.2	-10.0	46.8	40.7		
Average Gain															33.3	16.0	-11.1	51.5	44.1

Units are Area in  $mm^2$ , Power in  $\mu W$ , Maximum frequency (Freq) in  $MHz$  and Wire length (WL) in  $mm$ .

## V. EXPERIMENTAL RESULTS

Table VI shows the experimental results, obtained from the comparison between binary and quaternary FPGAs. Our experiments were realized with some filters with 8-bit random coefficients. Once circuits were generated, the P&R was performed as explained in Section IV. Results are obtained through Cadence UltraSim [17] simulation.

Results shown an important reduction of 16% on power consumption (PWR) with a small penalty on timing (Freq). The operation frequency is slower in the quaternary implementation due to the number of CLBs in the critical path. In binary implementations of the filters, the number of bits may increase only by one from one adder to the next one. Hence, only a slice (not a complete CLB) is inserted in the critical path. For the quaternary version, the critical path is increased by the delay of the full CLB, because it cannot be separated in two as in the binary case.

Wire length (WL) and the number of switches (Sw) used in the routing are the most important data in the results. Quaternary circuits present important gains due to the smaller bus width, but also because shift operations can be performed with reduced vertical connections. This way, the overall performance can be increased, since less switches will be present in the critical path.

## VI. CONCLUSION

This work presents important advances on the development of multi-valued circuits through the implementation of a transistor level arithmetic-oriented quaternary FPGA structure. Results show that the proposed quaternary FPGA is competitive with the binary one because of the important reductions on the connection sizes and number of switches, 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 filters is taken as a case study. Filters are an interesting case study because the placement and routing are simple to implement in the FPGA due to the regular structure.

In this paper we have successfully shown that significant power reduction can be achieved by a quaternary device. Increased frequency can be also obtained by implementing random logic in the quaternary LUT due to the possibility to reduce the number of CLBs without increasing the number of CLBs in the critical path.

The quaternary representation applied to the random logic will allow, not only the reduction of the number and size of the connections, but most important, the reduction of the fanout

and the load applied to the logic blocks. For this reason, we are developing logic synthesis and technology mapping algorithms 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] K. Banerjee, S. Souri, P. Kapur, and K. Saraswat, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 602–633, May 2001.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] E. Dubrova, "Multiple-valued logic in vlsi: Challenges and opportunities," in *Proceedings of NORCHIP'99*, 1999, pp. 340–350.
- [7] 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-mm<sup>2</sup> 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.
- [8] 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.
- [9] 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.
- [10] Z. Zilic and Z. Vranesic, "Multiple-valued logic in FPGAs," Aug 1993, pp. 1553–1556 vol.2.
- [11] 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.
- [12] L. Aksoy, E. da Costa, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 6, pp. 1013–1026, June 2008.
- [13] 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.
- [14] Xilinx Inc., "Spartan-3 fpga family data sheet," 2008. [Online]. Available: [http://www.xilinx.com/support/documentation/data\\_sheets/ds099.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf)
- [15] 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.
- [16] 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.
- [17] Cadence Design Systems Inc., "Virtuoso ultrasim simulator user guide," 2009. [Online]. Available: <http://www.cadence.com>