An Efficient Low Power Multiple-value Look-up Table Targeting Quaternary FPGAs

Cristiano Lazzari[1], Jorge Fernandes[2], Paulo Flores[2], and José Monteiro[2]

[1] INESC-ID, Lisbon, Portugal.
[2] INESC-ID / IST, TU Lisbon, Lisbon, Portugal.
{lazzari,jorge.fernandes,pff,jcm}@inesc-id.pt

Abstract. FPGA structures are widely used as they enable 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 interconnections in the circuit, hence can serve as a mean to effectively curtail the impact of interconnections. In this work we propose a new look-up table structure based on a low-power high-speed quaternary voltage-mode device. The most important characteristics of the proposed architecture are that it is a voltage-mode structure, which allows reduced power consumption, and it is implemented with a standard CMOS technology. Our quaternary implementation overcomes previous proposed techniques with simple and efficient CMOS structures. Moreover, results show significant reductions on power consumption and timing in comparison to binary implementations with similar functionality.

Key words: Multiple-value Logic, Quaternary Logic, Look-up Tables, FPGAs, Standard CMOS Technology

1 Introduction

Designers face new challenges in modern systems on a chip (SoCs) due to the large number of components. The high integration of different systems increases the number and length of interconnections, which 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, 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].

In particular, interconnections play a crucial role in Field Programmable Gate Arrays (FPGA), because they not only dominate the delay, but they also present a significant impact on power consumption [3] and occupied area [4].

Recent work suggests that in modern million-gates FPGAs, as much as 90% of chip area is dedicated to interconnections [5].

In order to keep the wide range of applications of the FPGAs in the market, one must deal with their excessive power dissipation, and this must be reduced without compromising computational power. One way to deal with this problem is to reduce the area occupied by the interconnections by, not only reducing the number of interconnections, but also the length of these interconnections.

Multiple-valued logic (MVL) has received increased attention in the last years because of the possibility to represent the information with more than two discrete levels in a single wire. Hence, 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].

MVL has been successfully accomplished in several type of devices such as adders [7] and multipliers [8], as well as programmable devices [5, 9] were also proposed. The main drawbacks of these previous MVL implementations are that they are either based on current-mode devices or demand extra steps in the fabrication process (for the generation of transistors with different V_{th}s). Current-based 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. On the other hand, while it is true that technologies with multiple V_{th}s deal very well with the power dissipation problem, as stated in [5, 10], their additional phases on the fabrication process make their implementation more difficult, more susceptible to variability problems and more expensive.

In this work we present a new implementation of a multiple-valued look-up table based on the quaternary representation, taking advantage of the analog nature of the multiple-valued representation. We implemented the quaternary look up-table by using a simple and efficient analog structure able to deal with the quaternary signals. Results show that our implementation overcomes the drawbacks of previous implementation and are competitive when compared to binary LUTs with the same functionality.

This paper is organized as follows. Section 2 discusses the differences between binary and quaternary look-up table implementations. Section 3 presents the new quaternary look-up table, giving details about the proposed structure. A comparison between the binary and quaternary look-up tables is presented in Section 4. Variability and the reduced noise margin effects in quaternary circuits are discussed in Section 5, and finally, Section 6 concludes the paper and outlines future work.

2 Binary and Quaternary Look-Up Tables Overview

General Look-Up Tables (LUT) are basically memories, which implement a logic function according to their configuration. Configuration values $C = (c_0, \cdots, c_i,$

$\cdots, c_{k-1}$) are initially stored in the look-up table structure, and once inputs are applied to it, the logic value in the addressed position is assigned to the output. The capacity of a LUT $|C|$ is given by

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

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 look-up 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 binary function implemented by a Binary Look-Up 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 $b = |B|$ ($b = 2$ in the binary case). For example, a look-up table with 4 inputs ($k = 4$) can implement one of $|F| = 65,536$ different functions.

Quaternary functions are basically generalizations of binary functions. A quaternary function implemented by a quaternary look-up 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 this case, the number of functions that can be represented is around 4.3×10^9 for a QLUT with only two quaternary inputs ($k = 2$), which is much larger than for the BLUT.

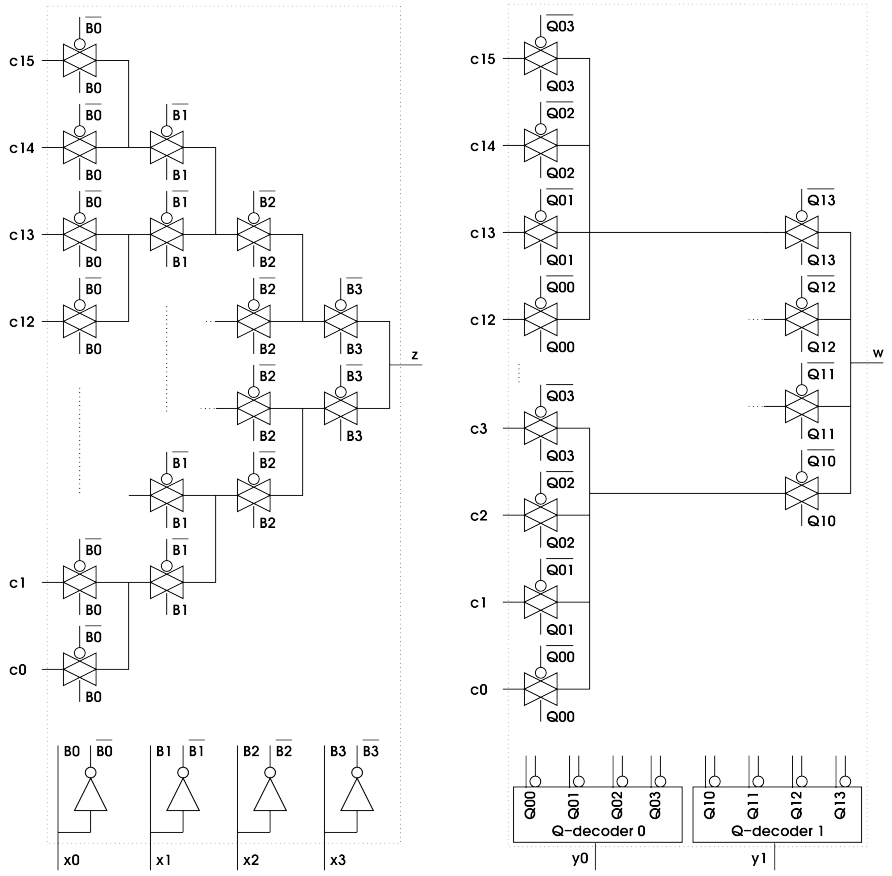
It is important to highlight that the function $g(Y)$ performs exactly the same function as two binary BLUTs, $f_0(Y)$ and $f_1(Y)$, where f_0 represents the least significant Boolean value and f_1 represents the most significant one. Following the same idea, the configuration values are also quaternary for the QLUT, which represent the values for two binary configuration values.

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

3 Look-up Tables Implementation

Binary and quaternary look-up tables were implemented with transmission gates. For the binary version, transmission gates are controlled by the BLUT inputs, while the QLUT is composed of transmission gates controlled by a new quaternary to binary device.

Fig. 1a shows a binary 4-input BLUT implementation ($b = 2, k = 4, |C| = 16$) where $x_i \in X$ are the inputs, $c_i \in C$ form the look-up table configuration



(a) 4-input BLUT. (b) 2-input QLUT.

Fig. 1: Binary and quaternary look-up table implementations.

and z is the output. The BLUT is composed of four stages as a consequence of the number of inputs. Multiplexers (implemented using transmission gates) are responsible for propagating the configuration values to the BLUT output. The transmission gates receive selection signals from the four BLUT inputs and associated inverters.

A quaternary look-up table (QLUT) follows the same structure as the BLUTs. Fig. 1b illustrates the implementation of a 2-input QLUT ($b = 4, k = 2, |C| = 16$). As in the binary case, $c_i \in C$ are the look-up table configuration values, $y_i \in Y$ are the inputs and w is the output. Due to the quaternary representation, only two stages of transmission gates are required.

The transmission gates are controlled by binary signals. Therefore, we need a special circuit to convert the quaternary inputs y_0 and y_1 to the correspondent control signals – the quaternary-to-binary converter (Q-decoder).

Table 1: The Q-decoder behavior as a funtion of the quaternary logic value at the input.

Q	Q_0	Q_1	Q_2	Q_3
0_4	1_2	0	0	0
1_4	0	1_2	0	0
2_4	0	0	1_2	0
3_4	0	0	0	1_2

3.1 Quaternary-to-binary Converter

Table 1 shows the Q-decoder binary output logic values as function of the quaternary input Q. Outputs Q_0 to Q_3 determine which transmission gates (in Fig. 1b) are propagating the configuration value $c_i \in C$ to the QLUT output w. Note that values for the controlling signals Q_0, Q_1, Q_2 and Q_3 are binary values, meaning 0 ($0V$) or 1_2 (V_{DD}).

The Q-decoder outputs may be considered as flags that determine which quaternary value is applied to Q-decoder input. Once we are able to determine the quaternary value in the Q-decoder input Q, the transmission gates connected to the Q-decoder outputs may be properly controlled. In other words, with the Q-decoder structure we are able to convert a quaternary input to a 4-bit word in one-hot codification and its inverted value.

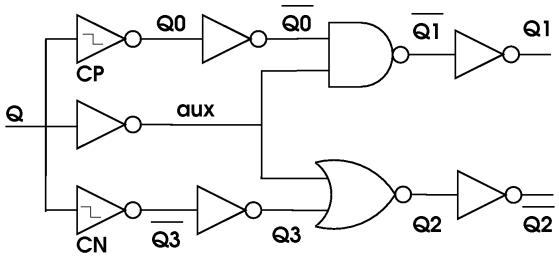
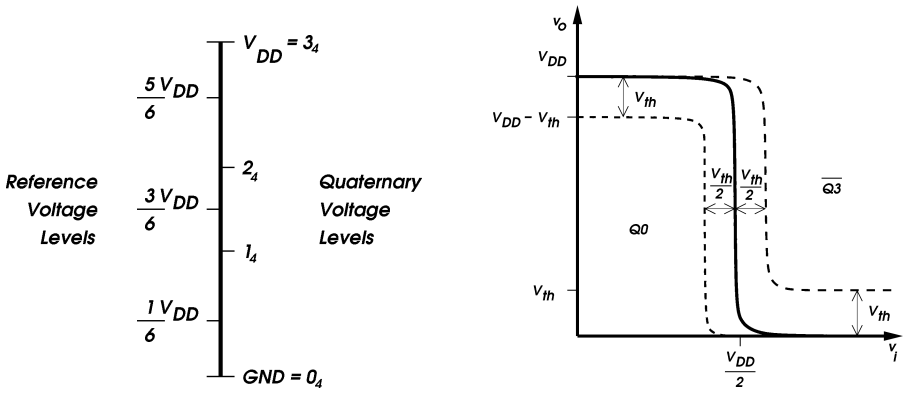


Fig. 2: The Q-decoder logic structure.

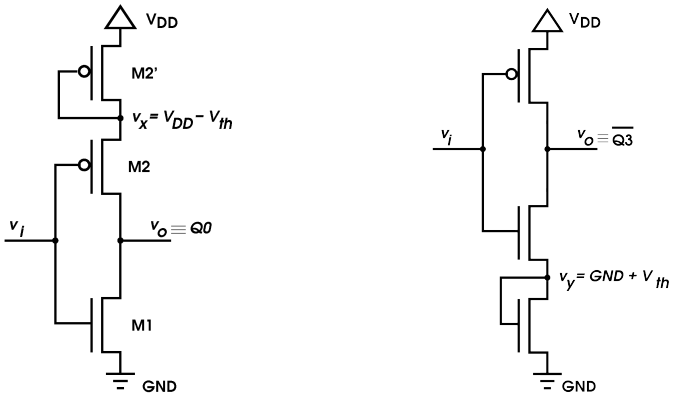
The Q-decoder structure is shown in Fig. 2. The main advantage of this structure over previous proposed implementations is that it is has standard CMOS structures. The Q-decoder is composed of two comparators *CP* and *CN*, and other traditional digital circuits such as inverters, NANDs and NORs.

The *CP* and *CN* are self-reference analog comparators shown in Fig. 3. With these structures we are able to detect the four possible voltage levels. In a binary implementation, an inverter may be seen as a comparator where the voltage reference is $V_{DD}/2$. For our quaternary device, we need three voltage references in order to determine a quaternary value, at $1/6V_{DD}$, $3/6V_{DD}$ and $5/6V_{DD}$, as depicted in Fig. 3a.

One way to obtain this comparator behavior is by designing inverters with unbalanced PMOS and NMOS transistor widths. The main drawback of this



(a) Logic levels. (b) CP and CN transfer functions.



(c) CP Structure. (d) CN Structure.

Fig. 3: Quaternary logic levels and comparators details.

technique is that it leads to large transistors widths with large gate capacitances, penalizing speed and power. Furthermore, in technologies with low V_{DD}, reference voltage values are below V_{th}, which makes this sizing technique impracticable.

To overcome this problem, we propose the use of the comparator circuits in Fig. 3c and Fig. 3d that add an extra transistor connected as a "diode" to shift the supply voltage by V_{th}.

In a first order approach, we consider simplified transistor models, and that transistors are equally sized ($k_1 = k_2 = k'_2 \rightarrow \mu_n(W/L)_1 = \mu_p(W/L)_2$), with equal threshold voltages ($V_{th1} = V_{th2} = V'_{th2} = V_{th}$). This simplified analysis is confirmed by simulations with more accurate models that will be presented in the next sections.

Reference points are defined by calculating v_x for $v_i = 0$ (3) and the transitions points (4), leading to the transfer function curves represented in Fig. 3b.

$$
\begin{aligned}
v_x\big|_{v_i=0} &\Rightarrow i_{D2} = 0 \\
&\Rightarrow k_2(V_{DD} - v_x - V_{th2})^2 = 0 \\
&\Rightarrow v_x = V_{DD} - V_{th2}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
i_{D1} = i_{D2} \Rightarrow k_1(v_i - V_{th1})^2 &= k_2(\underbrace{v_x}_{V_{DD}-V_{th}} - v_i - V_{th2})^2 \\
&\Rightarrow v_i - V_{th1} = V_{DD} - v_i - 2V_{th} \\
&\Rightarrow 2v_i = V_{DD} - V_{th} \\
&\Rightarrow v_i = \frac{V_{DD} - V_{th}}{2}
\end{aligned} \tag{4}
$$

The Q-decoder was implemented with the UMC 130nm technology. Simulations waveforms are shown in Fig. 4, where Q-decoder outputs are shown as expected and described in Table 1. The largest propagation delay from the Q-decoder input to the outputs ($Q \rightarrow \overline{Q_2}$) is 196ps for this technology. This result is very important, because an inverter connected to the same transmission gates (*i.e.*, same output load) presents a 81ps propagation delay, and the transmission gates are the main contributors to the look-up table propagation delay. More details about the comparison of binary and quaternary LUTs are given in the next section.

4 Binary vs Quaternary Look-up tables

We also implemented the complete binary and quaternary look-up tables with the UMC 130nm technology in order to evaluate their performance and power consumption. The development of the binary and quaternary LUTs was performed according to the Fig. 1. Transistor widths were kept to the minimum value in order to have a fair comparison between binary and quaternary versions.

We inserted buffers in the binary structure in order to reduce the impact of the gate capacitances. According to Fig. 1a, a cell connected to the BLUT input x_0 should drive 16 transistors. We balanced this gate capacitances by inserting 4 buffers, and thus improving the propagation delay. The power consumption was also reduced due to the faster transitions, and as a consequence, smaller short circuit times.

Experimental results are shown in Table 2, where the quaternary structure proposed in this paper outperforms the binary implementation in both power consumption and propagation delay. These results were obtained through CADENCE Spectre simulation [11]. The propagation delay is simply the largest delay from an input to the output of each LUT. The average power consumption

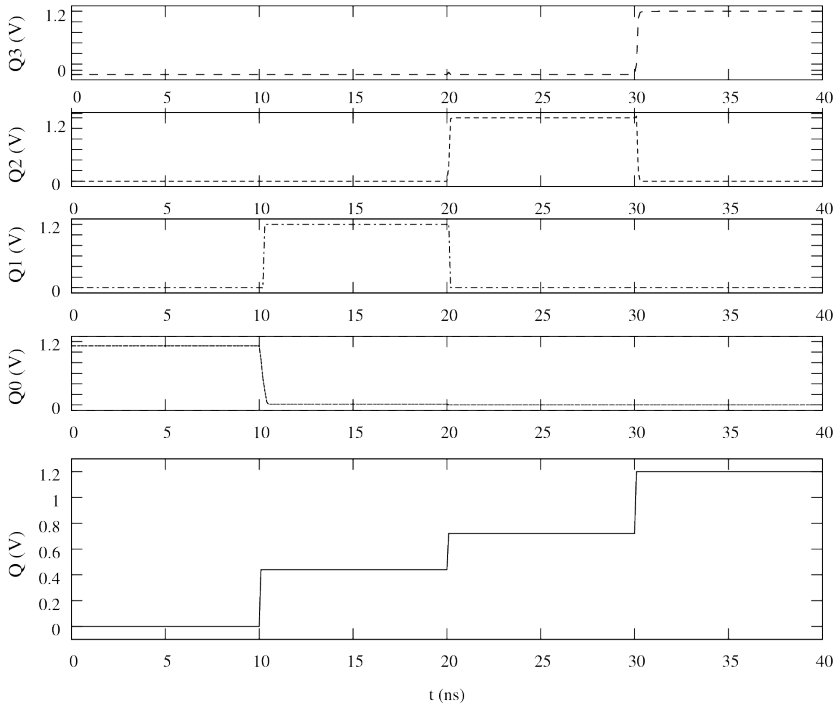


Fig. 4: The Q-decoder inputs and outputs waveforms.

is obtained from the simulation of 1024 random input vectors, when circuits were running at 100MHz.

For the quaternary circuits, we carefully took in consideration every single voltage source (*e.g.*, used to drive c_i values of the QLUT), so that the results shown in Table 2 reflect the real power consumption (*i.e.*, currents flowing from a voltage source to another are considered).

Results highlight that the quaternary look-up table, proposed in this paper, is very promising. In terms of delay, the quaternary LUT presents a very similar behavior, but better results are obtained when the load capacitance is 0.5pF or larger.

The power consumption is the most important result. According to Table 2, the quaternary LUT presents gains ranging from 22% (C_l=0.2pF) to 39% (C_l=1pF) in terms of power consumption. Note that, as for the propagation delay, gains are more important when the load capacitance increases.

It is clear that these gains related to the power consumption are obtained due to the reduced voltage levels. While binary transitions range from $0V$ to $1.2V$ (for this technology), quaternary transitions may vary from $0V \rightarrow 0.44V$ to $0V \rightarrow 1.2V$, demanding different current flows. Considering that all the possible transitions have the same probability, quaternary transitions have a smaller average voltage transition, reducing the average current flow and consequently the power dissipation.

Table 2: Delay and power consumption comparison of two 4-input BLUTs and one 2-input QLUT, both implemented with UMC 130nm process technology.

Output Load (C_l)	**2 4-input Binary LUTs**		**2-input Quaternary LUT**	
	Delay	**Power@100MHz**	**Delay**	**Power@100MHz**
0.2pF	0.91ns	45μW	0.95ns	35μW
0.5pF	1.9ns	68μW	1.7ns	43μW
1.0pF	3.4ns	94μW	3.0ns	57μW

In a practical implementation of a FPGA, there will be a smaller number of interconnections due to the quaternary representation, and hence we will also be able to reduce the wire length, and the parasitics capacitance will be smaller, as a consequence. For this reason, we expect to have better results than the ones presented in this paper, when developing a complete FPGA, based on the proposed circuits, to implement the quaternary logic.

5 Variability and Noise Margin in Quaternary Circuits

In current sub-micron and future technologies, process variability and reduced noise margin are important challenges for the development of multiple-valued devices. Voltage-mode multiple-valued logic devices present reduced voltage levels to represent logic values in comparison to binary circuits, and for this reason they may be, in theory, more susceptible to errors.

However, we performed Monte Carlo simulation with 500 runs to show that our quaternary LUT is robust to process variations when considering random process and mismatch variations. In this simulations, voltage variations are kept below 90mV for all the critical transition points (Q_0 and $\overline{Q3}$). Even with this variation range, we still have a 100mV gap between logic level transitions for other sources of noise or perturbations.

Noise levels are indeed reduced in quaternary circuits due to the fact that we have four voltage levels while keeping the same supply voltage. However, we may argue from a different perspective. In the last years, supply voltages have been reduced from $5V$, to $3.3V$, and recently to $1V$. This is a huge reduction in the noise margin and circuits have successfully coped with it.

It is important to highlight that the perturbations in the quaternary devices should be smaller than the binary ones because of the smaller average voltage transitions. Therefore a lower noise coupling between lines.

In summary, we may see the quaternary devices as a specific type of analog device. The knowledge and experience acquired by analog designers applied to the development of these devices in sub-micron technologies may be very useful in an effort to develop new multiple-value devices.

6 Conclusions

This work presents important advances in the development of multi-valued circuits through the implementation of a quaternary look-up table targeting multiple-valued FPGAs. Results show that the proposed structure is competitive with the binary one with significant reductions on power consumption and propagation delay. The technique proposed in this paper is simpler to implement than the previous proposed multiple-valued circuits. Furthermore, as far we know, no other proposed work is more efficient than our technique when comparing to binary circuits.

As future work, we are developing a complete FPGA (logic block, switch matrix, etc). A functional quaternary FPGA will allow the study of viability and the comparison with current binary circuits. We are also planning to implement our quaternary device in more recent technologies such as 45nm and below.

Acknowledgments. This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds and by the FCT project PTDC/EEA-ELC/72933/2006.

References

1. Gupta, A.K., Dally, W.J.: Topology optimization of interconnection networks. IEEE Comput. Archit. Lett. **5**(1) (2006) 3
2. Banerjee, K., Souri, S., Kapur, P., Saraswat, K.: 3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. Proceedings of the IEEE **89**(5) (May 2001) 602–633
3. Li, F., Lin, Y., He, L., Chen, D., Cong, J.: Power modeling and characteristics of field programmable gate arrays. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **24**(11) (Nov. 2005) 1712–1724
4. Singh, A., Marek-Sadowska, M.: 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) 59–66
5. da Silva, R., Lazzari, C., Boudinov, H., Carro, L.: CMOS voltage-mode quaternary look-up tables for multi-valued FPGAs. Microelectronics Journal **40**(10) (2009) 1466 – 1470
6. Dubrova, E.: Multiple-valued logic in vlsi: Challenges and opportunities. In: Proceedings of NORCHIP'99. (1999) 340–350
7. Gonzalez, A., Mazumder, P.: Multiple-valued signed digit adder using negative differential resistance devices. IEEE Transactions on Computers **47**(9) (Sep 1998) 947–959
8. Hanyu, T., Kameyama, M.: 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 **30**(11) (Nov 1995) 1239–1245
9. Zilic, Z., Vranesic, Z.: Multiple-valued logic in FPGAs. In: Proceedings of the 36th Midwest Symposium on Circuits and Systems. (Aug 1993) 1553–1556 vol.2
10. Cunha, R., Boudinov, H., Carro, L.: Quaternary look-up tables using voltage-mode CMOS logic design. ISMVL 2007. 37th International Symposium on Multiple-Valued Logic (May 2007) 56–56
11. Cadence Design Systems Inc.: Virtuoso spectre simulator user guide. (2010)