Journal of
Low Power Electronics
Vol. 7, 1–8, 2011

Low Power Multiple-Value Voltage-Mode Look-Up Table for Quaternary Field Programmable Gate Arrays

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*

(Received: 24 December 2010; Accepted: 6 February 2010)

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 using 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.

Keywords: 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 types of devices. However, the main drawbacks of these 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_{ths}).

In this work we present a new implementation of a multiple-valued look-up table based on the quaternary

*Author to whom correspondence should be addressed.
Email: lazzari@inesc-id.pt

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. In Section 2 we give an overview of related work. Section 3 discusses the differences between binary and quaternary look-up table implementations. Section 4 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 5. Variability and the reduced noise margin effects in quaternary circuits are discussed in Section 6. Section 7 highlights some characteristics of the quaternary lookup table in quaternary FPGAs, and finally, Section 8 concludes the paper and outlines future work.

2. RELATED WORK

The concept of representing the information using MVL is not recent. Multiple-value Logic has been proposed in several devices, such as flash memories, some combinational circuits and multipliers, as well as programmable devices.

MVL has been successfully accomplished in Flash memories,[12] for example, where a single memory cell is able to hold different logic values. The proposed MVL memory device allows the storage of 2 bits in a single cell, which reduces the total memory area.

Other devices such as adders[7] and multipliers,[8] as well as programmable devices[9] were also proposed. Current-mode circuits allow successful improvements in terms of area, however their excessive power consumption and implementation complexities has prevented, until now, MVL systems from being a viable alternative to standard CMOS designs.

A voltage-mode look-up table implementation was presented in Refs. [5, 10], dealing specifically with the power dissipation problem, and still using a standard CMOS process. The main idea in the proposed work was to insert additional steps on the fabrication process, allowing the implementation of transistor with different V_{th}s. While it is true that technologies with multiple V_{th}s deal very well with the power dissipation problem, as stated in Refs. [5, 10], the required additional phases on the fabrication process make their implementation more difficult, more susceptible to variability problems and more expensive.

3. 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 $C = (c_0, \ldots, c_i, \ldots, c_{k-1})$ 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 $X = (x_0, \ldots, x_i, \ldots, x_{k-1})$, where each variable 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 $Y = (y_0, \ldots, y_i, \ldots, y_{k-1})$, 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 have the cardinality of $|Q| = 2 \times |B|$. 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.

4. 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