On the Analysis of an FPGA Architecture

Tsutomu SASAO  
Department of Computer Science and Electronic Engineering  
Kyushu Institute of Technology  
lizuka 820, Japan

Jon T. Butler  
Department of Electrical & Computer Engineering  
Naval Postgraduate School, Code EC/Bu  
Monterey, CA 93943-5004  
U.S.A.

Abstract
An architectural model of a field programmable logic array (FPGA) is presented. It consists of \( r \)-copies of \( k \)-input ROMs connected by a universal interconnection matrix. In this FPGA, an arbitrary multilevel network using at most \( r \) ROMs is realizable. Upper and lower bounds on the number of the functions realizable in this FPGA are derived.

1 Introduction

Field programmable gate arrays (FPGA's) are devices that can be programmed by the user to implement a logic function. The ease with which this can be done makes them suitable for rapid prototyping applications. In addition, they are inexpensive to manufacture and are fully testable. Automatic tools exist that allow one to program a high-level description onto these architectures. All FPGA architectures consist of repeated arrays of identical logic blocks. A logic block is a versatile configuration of logic elements which can be programmed by the user. The interconnections to realize the circuit must also be programmed. A method for doing this is shown in [6].

Among various FPGA architectures, two are especially popular, the ROM type and the MUX (multiplexer) type. In the ROM type architecture, each logic block can realize an arbitrary \( k \) variable function (Fig. 1). An example is a Xilinx (3090) FPGA, where \( k = 5 \), in which there are 320 logic blocks[11]. It is also called a Table Look Up (TLU) based FPGA or a RAM based FPGA. For convenience, we call a logic block a ROM.

In the MUX (Fig. 2) architecture, each logic block is a multiplexer. An example is the Actel FPGA. In such a FPGA, both logic blocks and the interconnections are programmable, as in the ROM type FPGA.

In this paper, we analyze the ROM type FPGAs. Suppose that the total area of the FPGA is fixed. When \( k \), the number of inputs to each ROM is large, each ROM can realize a complex function. However, the number of the ROMs in the FPGA must be small. On the other hand, when \( k \) is small, each ROM can realize only a simple function. However, the number of ROMs can be large. In both cases, the area for the interconnections among ROMs must be considered. An interesting question is "What value of \( k \) yields the optimum functionality?" That is, what value of \( k \) produces the largest number of functions?" Some interesting results have already been obtained in answer to this question. Assuming certain interconnections and logic blocks, the best value for \( k \) has been shown to be three or four[14, 17].

In this paper, we study a similar problem using a different FPGA model based on the Xilinx 3090. One problem with the Xilinx FPGA architecture is rout-
ing. Indeed, routing appears to be more difficult than logic synthesis. Another problem is the propagation delay. Three kinds of interconnections are available in Xilinx FPGAs, each with different wiring delays. A typical delay of the logic block is 15 ns, but the wiring delay varies from 2 to 80 ns. Because the timing is routing dependent, the design of Xilinx FPGAs is very complex.

In order to realize an arbitrary function of \( n \) variables, we must find \( r \) and \( k \) properly. When \( r = 1 \), then \( k \) must be \( n \). Thus,

\[
A = 2^n + n(2^\frac{n}{2}) + n^2.
\]

However, this is essentially the same as a ROM that realizes all functions on \( n \) variables, and the chip is too large, except for small \( n \).

## 3. On the Number of Functions Realizable by FPGAs

In most applications, FPGAs need not realize all the functions of \( n \) variables. For example, in the case of PLAs, to realize an arbitrary function of \( n \) variables, \( 2^{(n-1)} \) products are necessary. However, in most applications, the number of the products needed in PLAs is typically much smaller. Indeed, in PLA's there is a point, \( p \), of diminishing returns with respect to \( t \), the number of products, where for \( t > p \), there is little benefit in additional functions realized. When \( t = 2^{(n-1)} \), the PLA is universal, but too large. When \( t \) is very small, the full benefit of the PLA is not realized. Research work on the number of the functions realizable by a PLA with \( t \) products have been done [1, 18].

In this section we will consider the number of different functions realizable by an FPGA. We will use the following symbols to denote the parameters of the FPGAs.

- \( n \): number of the external inputs.
- \( k \): number of the inputs for each ROM.
- \( r \): number of ROMs in a FPGA.

\( k = 5 \) holds special interest because, in the Xilinx 3090 FPGA, each ROM has five inputs; further, \( r = 320 \).

### 3.1 Upper bound on the number of the functions

**Theorem 3.1** Let \( N(n, k, r) \) be the number of functions realizable with a FPGA, then

\[
N(n, k, r) \leq 2^n((n+r-1)k+2^r)
\]
(Proof) The number of the different patterns in the UIM is \(2^{W \times H}\), where \(W = n + r - 1\) and \(H = r \cdot k\). The number of different functions realizable in each ROM is \(2^r\). Therefore, we have the theorem. (Q.E.D.)

For example, when \(k = 5\) and \(r = 320\), then
\[
N(n, k, r) \leq 2^{320(6n+1627)}.
\]

3.2 Lower bounds on the number of functions.

To obtain the lower bounds, we consider various network structures.

3.2.1 When \(k = 6\).

Consider the network shown in Fig. 4, where, \(z_1, z_2, \ldots, z_6\) are control variables for the quaternary tree. Variables \(z_7\) to \(z_n\) are connected to the ROMs of the first level. Note that the total number of ROMs is \(1 + 4 + 16 + 64 = 127\).

Theorem 3.2 The different number of functions realized by the network in Fig. 4 is at least
\[
\left( \sum_{i=0}^{6} \binom{n-6}{i} M(i) \right)^{64},
\]
where
\[
M(i) = \sum_{j=0}^{i} (-1)^{(i-j)} \binom{i}{j} 2^{3j}.
\]

(Proof) For the quaternary tree for \(z_1\) to \(z_6\), suppose that each ROM realizes a multiplexer with two control variables. For example, the output ROM realizes the function
\[
f = \alpha z_1 z_2 \lor b z_1 z_2 \lor c z_1 z_2 \lor d z_1 z_2.
\]

Then, the quaternary tree selects only one ROM out of 64 ROM's in the first (leftmost) level. For each ROM in the first level, there are \(\binom{n-6}{i}\) ways to choose \(i\) variables from \(n - 6\). Each ROM can realize a function of \(i\) variables in
\[
N = \sum_{i=0}^{6} \binom{n-6}{i} M(i)
\]
ways, for \(0 \leq i \leq 6\), where \(M(i)\) is the number of functions dependent on exactly \(i\) variables. By inclusion-exclusion,
\[
M(i) = \sum_{j=0}^{i} (-1)^{(i-j)} \binom{i}{j} 2^{3j}
\]
Figure 4: Network structure for $k = 6$
Because, for each combination, a distinct function is realized, the total number of functions realized is \( N^{64} \). This proves the theorem. (Q.E.D.)

3.2.2 When \( k = 5 \).

Consider the network shown in Fig. 5, where, \( x_1, x_2, \ldots, x_8 \) are control variables for the ternary tree. Variables, \( x_9 \) to \( x_n \) are connected to the ROMs of the first level. Note that the total number of ROMs is \( 1 + 3 + 9 + 81 + 243 = 337 \).

**Theorem 3.3** The different number of functions realized by the network in Fig. 5 is at least

\[
\left( \sum_{i=1}^{5} \binom{n-8}{i} M(i) \right)^{243} 4^{94},
\]

where

\[ M(i) = \sum_{j=0}^{i} (-1)^{(i-j)} \binom{i}{j} 2^{2^j}. \]

(Proof) Let the ternary tree with inputs \( x_1 \) to \( x_8 \) realize a multiplexer. For example, each ROM can realize

\[ f = a x_1 x_2 \vee b \overline{x}_1 x_2 \vee c x_1 x_2, \]

where exactly one of \( a, b, \) and \( c \) is selected. Thus, the ternary tree selects one of the 243 ROM's at the first level.

For each ROM on the first level, there are \( \binom{n-8}{i} \) ways to choose \( i \) variables from \( n-8 \). Each ROM can realize a non-constant function of \( i \) variables in

\[ N = \sum_{i=1}^{5} \binom{n-8}{i} M(i) \]

ways, for \( 1 \leq i \leq 5 \), where \( M(i) \) is the number of functions dependent on exactly \( i \) variables, which is given in (1).

If the first level ROMs generate non-zero functions, then all the different MUX configurations will produce different functions. First, we consider the case when only one MUX is used. We have four different expressions for the configurations of the MUX.

\[ f = x_1 x_2 f_1 \vee x_1 x_2 f_2 \vee x_1 \overline{x}_2 f_3, \]

\[ f = x_1 x_2 f_1 \vee \overline{x}_1 x_2 f_2 \vee x_1 \overline{x}_2 f_3, \]

\[ f = x_1 \overline{x}_2 f_1 \vee x_1 x_2 f_2 \vee \overline{x}_1 \overline{x}_2 f_3, \]

\[ f = x_1 x_2 f_1 \vee \overline{x}_1 x_2 f_2 \vee \overline{x}_1 \overline{x}_2 f_3. \]

Because they are different canonical expressions, they represent different functions. (No matters what function \( f_j \) is, where \( j = 1, 2, 3, \ldots \). Note that \( f_j \) is not a constant-zero function.) In the case of Fig. 5, the output function can be represented as

\[
\bigwedge_{j=1}^{243} f_j(x_8 x_7^r)(x_6 x_5^r)(x_4 x_3^r)(x_2 x_1^r),
\]

where \( x_i^r \) denotes \( x_i \) or \( \overline{x}_i \). For each configuration of MUXs, we have a distinct expression. Because the expression is canonical, each function is distinct. For each ROM in the ternary tree, we have four different ways to specify the MUX. Because there are \( 1 + 3 + 9 + 81 = 94 \) ROMs for MUX, the number of different configuration of MUXs is \( 4^{94} \). For each ROM on the first level, and for each configuration of MUXs, a distinct function is realized. Thus, the total number of functions realized is \( N^{243} 4^{94} \). This proves the theorem. (Q.E.D.)

4 Conclusion and Comments

In this paper, we presented a new FPGA architecture. The feature of the architecture is that the wiring is very easy. We also derived the upper and lower bounds on the number of the functions realizable by such FPGAs. They are useful to estimate the capability of the commercially available ROM based FPGAs such as Xilinx 3090.

Acknowledgments

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan, by the International Information Science Foundation, and by the Naval Research Laboratory through direct-funds to the Naval Postgraduate school. Prof. G. DeMicheli sent us a copy [14].

References

Figure 5: Network structure for $k = 5$


International Symposium on
Logic Synthesis and Microprocessor Architecture

As a part of
International Symposia on Information Sciences (ISKIT'92)

July 12–15, 1992

Kyushu Institute of Technology
Iizuka, Fukuoka, Japan