Field-Programmable Gate Array

2

Mário Pereira Véstias
https://orcid.org/0000-0001-8556-4507
INESC-ID, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Portugal

INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) are integrated circuits whose logic and their interconnections are configurable. These devices are field-programmable, that is, they can be configured by the hardware designer without any intervention of the manufacturer. Most FPGAs can be reprogrammed as many times as we want with a vast variety of digital circuits.

The applicability of FPGAs have changed since they appeared in the mid-1980s. Initially they were very limited in terms of resources and therefore were used to implement glue logic and small sized digital systems. With the advance of the integrated circuit technology, its density has increased dramatically permitting the implementation of a vast set of applications in different areas like consumer, industrial, automotive, high-performance and communications. Some recent FPGA families are system-on-chips (SoC) with one or more microprocessor cores, memory, cache and reconfigurable logic allowing the implementation of complex hardware/software systems in a single programmable device.

FPGAs are now an alternative for ASIC (Application Specific Integrated Circuits). ASIC are the best solutions for implementing digital systems since they offer better performance and lower power consumption at smaller silicon area. The problem is that designing an ASIC is very time-consuming, with high nonrecurring engineering (NRE) costs requiring expensive tools to design and develop. Besides, after fabrication of the final design in ASIC it cannot be modified without creating a new chip.

The cost unit of an ASIC is lower than that of an FPGA but has a much higher NRE cost. Therefore, the ASIC is the best option only for high volume products. This makes FPGAs the chosen device for small to medium scale product deployment. Designing an FPGA is easier, it permits modifications and upgrades of the design, has a faster time-to-market and so the only viable solution for small companies that cannot incur the high NRE costs and expensive design tools.

This article focus on the architecture of FPGAs, including the so called SoC FPGA. It explains the main blocks of the FPGA, how they have evolved along the last decades and the perspectives of next generation FPGAs. It also describes some applicability areas and how its architecture have evolved to adapt to some of these target markets.

BACKGROUND

Field-Programmable Gate Arrays are silicon devices whose fabric hardware can be programmed to implement a particular digital system. The hardware programming capability of FPGAs gives them some advantages over Application Specific Integrated Circuits (ASIC) whose underlying fixed hardware is design to implement a specific digital system. Deploying and maintaining an FPGA-based system is faster and cheaper than implementing the same system in ASIC, since the ship is already made and just

DOI: 10.4018/978-1-7998-3479-3.ch020

have to be configured to implement a specific function. Later, this function can be changed in the field without having to redesign a new chip and can even be reconfigured dynamically while running some other functionality. While is cheaper to design FPGA than an ASIC, the unit cost is higher for an FPGA. Therefore, the average cost of each of these devices depend on the production volume. The average unit cost of an ASIC decreases faster with the production volume than the cost of an FPGA unit. Therefore, there is a threshold from which ASICs are more cost effective.

The programming flexibility of FPGAs comes with a cost in terms of performance and power consumption. A large area of the FPGA (from 80 to 90%) is dedicated to routing and programming resources. It means that only 10-20% of the chip area is available to implement the hardware system. So, compared to an ASIC, a larger FPGA is necessary to implement the same functionality, the performance is worst due to the delay associated with the configurable routing and logic and the higher power consumption includes the power consumption of programming resources.

Any digital system consists of functional blocks, interconnections between these blocks and input/outputs connections between the functional blocks and off-chip circuits. An FPGA consists of programmable logic, interconnect and I/O modules that are tailored to implement a specific digital system.

FPGA Functional Blocks

The granularity of functional blocks determines the flexibility of the FPGA. The smaller the granularity the more flexible it becomes and the FPGA can be more deeply adapted to the required logic system. In the limit, we can use transistors as the basic cells that could then be interconnected to implemented basic logic gates. This approach was followed in (Marple, D., & Cooke, L., 1992) without success mainly because this very fine granularity requires large amounts of interconnections turning the device quite area-inefficient, with large propagation delays and high power consumption associated with the capacitances of the programmable interconnections.

Different granularities were tried in the last decades, from NAND gates (Plessey, 1992), trees of multiplexers (Gamal et al., 1989) and lookup tables (Carter et al., 1986). Today, an FPGA consists of fine-grain configurable circuits of logic, I/O and interconnections arranged in a regular map see Figure 1).

Configurable input/output (I/O) blocks establish the communication of data to and from the system configured in the FPGA. Configurable logic blocks (CLB) implement the functionality of the system and routing switch (SW) provide configurability to the routing resources.

A configurable logic block issues fine-grained configurable logic and memory resources organized in a hierarchical structure (see figure 2).

A configurable logic block (different vendors call different names to these blocks: CLB, LAB – logic array block, etc.) encapsulates fine-grained logic and memory and some specialized configurable circuitry that improves the design of some functions, like arithmetic. One CLB contains one or more cells with dedicated connections between neighbor cells (the designation of these elements are also vendor dependent; e.g. ALM – adaptive logic module and slice).

Logic in FPGAs is implemented with small look-up tables (LUT) which are basically small memories whose content can be changed to implement a specific function. A LUT with k inputs, designated LUT-k, implements any boolean function with k-inputs. Each cell of a CLB contains one or more locally interconnected LUTs and flip-flops. The size of the LUT determines the area and the delay of the circuit according to two trading dimensions. Larger LUTs reduce the number of logic blocks in the critical path, the amount of inter logic routing, which improves performance. However, as we increase the size of the LUT, the internal delay increases and so smaller logic functions mapped to the LUTs have higher delays.