

APPLICATION-DOMAIN SPECIFIC RECONFIGURABLE FPGA PLATFORM: AN INTEGRATED HARDWARE AND SOFTWARE DESIGN APPROACH

Kostas Siozios and Dimitrios Soudris

{ksio, dsoudris}@ee.duth.gr

Department of Electrical and Computer Engineering,
Democritus University of Thrace, 67100, Xanthi, Greece

Abstract

This work aims at the development of systematic methodologies both in hardware and software level for efficient application implementation on low-energy and high performance heterogeneous FPGA.

1. Introduction

In this work we describe the goals that have been succeeded up to now in the design of the application-domain specific reconfigurable FPGA platform. Among them, we design a low-energy FPGA, we introduce the heterogeneous interconnection architectures, we develop a number of methodologies for controlling the switching activity (which leads to reduced power consumption and temperature distribution across the device), as well as the development of a complete framework of CAD tools for application implementation onto the proposed FPGA.

2. FPGA Architecture

The first part of the research work concerns the design of a full-custom fine-grain reconfigurable architecture that supports partial and dynamic reconfiguration and can be either an embedded or a discrete FPGA. The designed device has low-energy and high performance characteristics. Initially, we designed a low-energy Configurable Logic Block (CLB) at 0.18 μm STM technology. Several circuit-level low-power design technologies were used. Among others, double-edge triggered flip-flops and gated clock signaling low power optimization design techniques were applied.

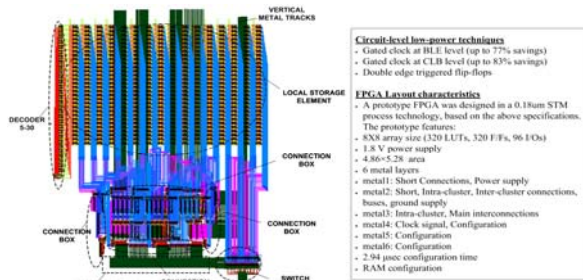


Fig. 1: Tile layout

The characteristics of the low energy and high performance FPGA are: a) Cluster of 5 BLEs, b) 4-inputs LUT per BLE, c) One double edge-triggered Flip-Flop per BLE, d) One Gated Clock signal per BLE and CLB, e) $I=12$ inputs and $N=5$ outputs provided by each CLB, f) All 5 outputs can be registered g) A fully Connected CLB resulting to 17-to-1 multiplexing in every input of a LUT, h) One asynchronous Clear signal for whole CLB and i) One Clock signal for whole CLB. The detailed design and circuit characteristics of the CLB, as well as the interconnection network, are determined and evaluated for energy, delay and area. SPICE simulations showed energy

savings around 38% in average at CLB level. Detailed info can be found in [2]. The layout of a single tile is shown in Fig. 1.

3. Heterogeneous Architectures

The routing architecture is the second main building block of an FPGA. We proposed a heterogeneous interconnection network consisting of more than one Switch Box (SB) patterns placed onto the same device (see Fig. 2). Taking into account the application-domain characteristics, is to we determined the optimal combination among the available SBs. This information can be extracted from the statistical and spatial routing restrictions of the implemented applications on a homogeneous (conventional) FPGA. More specifically, we have proposed a new technique for selecting the appropriate combination of SBs, depending on the localized performance and power consumption requirements of each specific region of FPGA architecture [3].

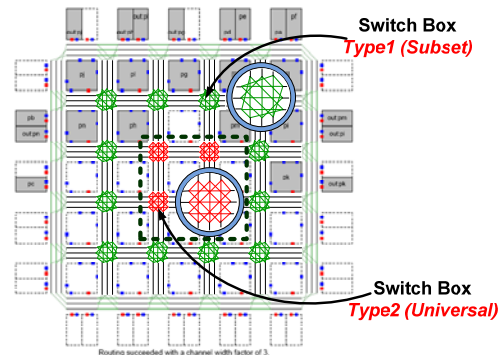


Fig. 2: FPGA architecture consisted by two different SBs

If the designer needs additional performance and lower power consumption, he/she can replace some existing transistors that form routing connections (placed within SBs) with hardwired connections. By this way the network's capacitance is reduced, leading to reduced delay and power consumption, as wires are faster and consume less power compared to transistors. Eventually, the designer can handle the routing components as building blocks of a platform-based design approach.

Extensive comparative study of various DSP applications (image processing, cryptography, elliptic filter, etc.) proved that performance gains up to 73% compared with commercial devices (Stratix) and energy savings up to 10% can be retrieved.

4. Switching Activity Management

Special attention was paid on developing methodologies for controlling the switching activity across the FPGA. The aim of these is to "transfer" switching activity from high-switching activity regions to the rest device. By this way, we achieve a more "uniformly" distributed picture of the switching activity

¹ This work was partially supported by the project IST-34793-AMDREL, the project PENED '03 and the project PYTHAGORAS II which are funded by the European Commission and the GSRT of Ministry of Development.

across the whole FPGA. Due to the proportional relation between switching activity, power consumption and temperature, we can conclude that these methodologies lead to lower on-chip temperatures.

Fig. 3 compares the average percentage of FPGA die that appears specific temperature values (in normalized manner) between the well-known VPR tool and our proposed temperature-aware solution, over the 20 biggest MCNC benchmarks. The horizontal axis represents the different temperature classifications ranging from the minimum value that appears on the die (50% of the maximum temperature) up to the maximum temperature value (100%). Based on this graph we can conclude that we achieved to increase the percentage of die area with lower temperatures (ranging from 50% up to 70% of the maximum temperature value) from 9% up to 24%, while we reduced the percentage of the die area where the hotspots appear (ranging from 70% up to 100% of the maximum temperature value) about 33%.

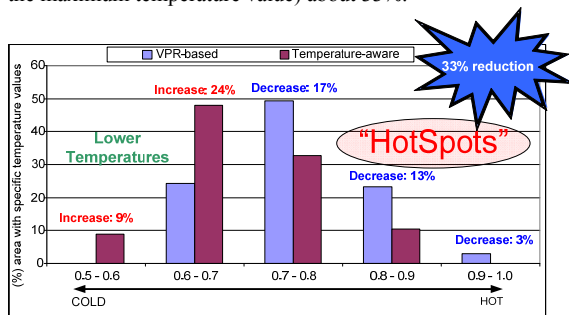


Fig. 3: Average temperature variation

The gains of the proposed methodology for the controlling of switching activity can be concluded as follows: It spreads power (and temperature) across the whole FPGA, it reduces the maximum values of power (and temperature) at the device up to 33%, it eliminates the number of power (and temperature) spikes, both the power and the temperature have a more uniform picture and eventually, it has no impact on device performance, total power, energy consumption, as well as silicon area. The new methodology is a power/temperature management methodology.

5. Supporting CAD Tools

Equally important to an FPGA platform is the tool set, which supports the implementation of digital logic on the proposed FPGA. The second part of the work concerns the development of a Linux-based design framework, named *MEANDER* [2] (see Fig. 4), consisting of easy to use tools capable to program an FPGA with features that described previously. The framework fulfilling both the needs of experienced designers by providing practical answers to state-of-the-art problems (e.g. logic synthesis, bitstream generation), and novice designers by providing a simple and consistent set of tools. To best of our knowledge, it is the *first complete academic design flow* beginning from an RTL description of the application and producing the configuration bitstream. It should be mentioned that the framework provides technology independence in order to allow designers to easily implement their designs in different process technologies, while it is easily extended in order to handle more advanced architectures (such as 3D FPGA, NoC, etc.). All the tools are open-source and available for on-line execution through AMDREL's project website [2].

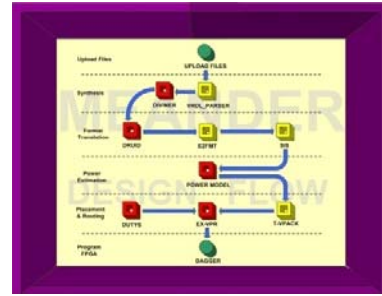


Fig. 4: MEANDER design framework

Table 1 shows a qualitative comparison among the proposed design framework, two commercial flows (Xilinx/Altera) and two other academic approaches. The (✓) symbol indicates that the corresponding feature is available in the design framework, while the (✗) symbol indicates that the specific feature is not supported by the design framework. Based on the available features, the proposed design flow is the most complete academic framework, and is at least in terms of provided features comparable with commercial tools.

Table 1: Qualitative comparison study

FEATURE	MEANDER	Xilinx / Altera	Univ. of Toronto	Alliance
Input Format	VHDL/ Verilog	VHDL/ Verilog	BLIF	VHDL
Synthesis	✓	✓	✗	✓
Power/Area estimation	✓	✓	✓	✗
Architecture description	✓	✓	✗	✗
Architecture exploration	✓	✗	✓	✗
Multiple Switch Boxes	✓	✓	✗	✗
Insertion of IP modules	✓	✓	✗	✗
Placement & Routing	✓	✓	✓	✓
Bitstream	✓	✓	✗	✗
Partial Reconfiguration	✓	✓	✗	✗
Back-annotation	✗	✓	✗	✗
GUI	✓	✓	✗	✗
Access through HTTP	✓	✗	✗	✗

Beside the other features of the flow, the MEANDER Design Framework integrates the only known academic tool for generating (from scratch) configuration files (named *DAGGER*) with such features. Among them are the run-time, partial- and dynamic-reconfiguration, the memory management, the bitstream compression and encryption, the read-back technique, the bitstream re-allocation, the used low-power techniques as well as its graphical user interface. MEANDER is the *only one* in the market FPGA framework, which allows designers to performing detailed exploration within an FPGA device. In addition to, it will be very useful instrument for educational purposes [4].

6. References

- [1] K. Siozios et. al., "Platform-based FPGA Architecture: Designing High-Performance and Low-Power Routing Structure for Realizing DSP Applications", 13th Reconfigurable Architectures Workshop, 2006.
- [2] <http://vlsi.ee.duth.gr/amdrel>
- [3] K. Siozios, et. al., "An Efficient Interconnection Structure for Energy-Delay Product Optimal FPGA Architectures", IFIP Int. Conf. on Very Large Scale Integration (VLSI-SoC), 2006
- [4] S. Vassiliadis and D. Soudris, "Fine- and Coarse-Grain Reconfigurable Computing," Springer (appear April 2007)