Institute of Physics Publishing Journal of Physics: Conference Series **10** (2005) 352–356
doi:10.1088/1742-6596/10/1/086 Second Conference on Microelectronics, Microsystems and Nanotechnology

Fine-Grain reconfigurable platform: FPGA hardware design and software toolset development

I Pappas[1], **V Kalenteridis**[1], **N Vassiliadis**[1], **H Pournara**[1], **K Siozios**[2], **G Koutroumpezis**[2], **K Tatas**[2], **S Nikolaidis**[1], **S Siskos**[1], **D J Soudris**[2] **and A Thanailakis**[1]

[1]Electronics and Computers Div., Department of Physics, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece
[2]VLSI Design and Testing Center, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
E-mail: ilpap@auth.gr,vkale@skiathos.physics.auth.gr

E-mail: ksiop@ee.duth.gr

Abstract. A complete system for the implementation of digital logic in a fine-grain reconfigurable platform is introduced. The system is composed of two parts: The fine-grain reconfigurable hardware platform (FPGA) on which the logic is implemented and the set of CAD tools for mapping logic to the FPGA platform. A novel energy-efficient FPGA architecture is presented (CLB, interconnect network, configuration hardware) and simulated in STM 0.18μm CMOS technology. Concerning the tool flow, each tool can operate as a standalone program as well as part of a complete design framework, composed by existing and new tools.

1. Introduction

FPGAs have recently benefited from technology process advances to become significant alternatives to ASICs. An important feature that has made FPGAs, particularly attractive is a logic mapping and implementation flow similar to the ASIC design flow (from VHDL or Verilog down to the configuration bitstream) provided by the industrial sector [1][2].

In this paper, an energy efficient FPGA architecture is presented at both CLB and interconnection network level. The design is mostly focused on minimizing energy dissipation, without significantly degrading delay and area. Additionally, a complete tool-supported design flow for mapping logic on the FPGA is also presented starting from a VHDL circuit description down to the FPGA configuration bitstream. Section 2 describes the proposed structure for CLB, interconnect network and configuration architecture. Section 3 presents the proposed design flow. Finally, conclusions and future work are discussed in section 4.

2. FPGA architecture

In this section the FPGA architecture, which can be configured using the developed toolset, is presented. The main design constraint is the energy minimization under delay constraints, while keeping a reasonable silicon area.

2.1. CLB architecture

The design of the CLB architecture is crucial to the CLB granularity, performance, and power consumption. The proposed FPGA is cluster-based [3], and consists of a collection of Basic Logic

Elements (BLEs), which are interconnected by a local network. Figure 2a shows the structure of a Basic Logic Element (BLE), which is formed by a Look-Up Table (LUT), a D-F/F and a 2-to-1 multiplexer, while in figure 2b a cluster of BLEs form the CLB. After an exhaustive exploration [9][10], it has been resulted that the selected features of the CLB which lead to minimization of energy consumption 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) 12 inputs and 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.

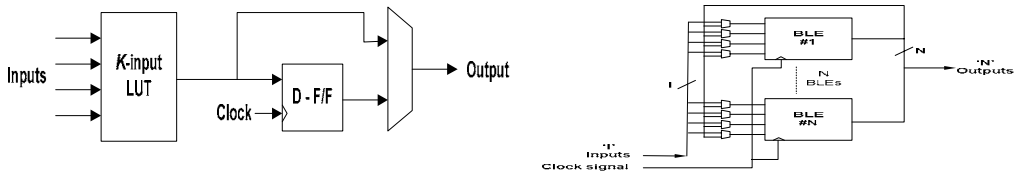

Figure 1. a) Basic Logic Element (BLE) b) Proposed cluster-based CLB

2.2 CLB Interface
Another critical issue for the FPGA performance is the CLB I/O pin positioning around the CLB perimeter. It has been shown that the full perimeter versus top/bottom pin positioning results in the better performance.

Further exploration was attempted to specify the exact best positioning. Our experiement flow used some of the MCNC benchmarks circuits. We tried the same architectures with different pins positioning. Fig. 2 shows a Rounded Pins Positioning in which I/Os pins are distributed everly around the CLB perimeter and Biased Pins Positioning in which Outputs indicated only at the right of the CLB. Our exploration showed that the Rounded Pins Positioning results in more routable FPGA and therefore better area, speed and power efficiency

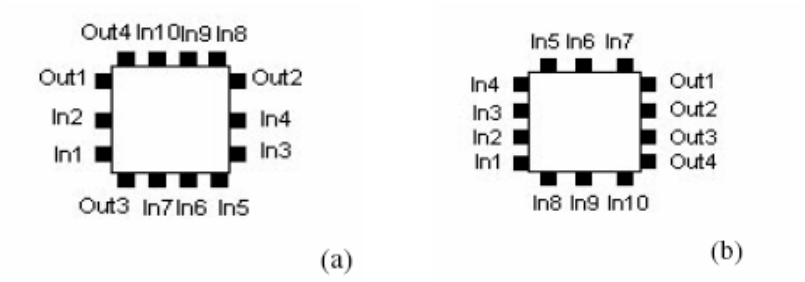

Figure 2. a) Rounded Pins Positioning b) Bias Pins Positioning

2.3. Interconnect architecture
Alternative interconnection architectures were examined and their performances in terms of speed, energy consumption and area were determined and evaluated. A detailed exploration was performed for the switch block types, the length of the segments, the connectivity factors (F_C, F_S) and for the population factor [11]. In addition at the circuit design the best type and size of routing switch as well as the optimum width and spacing between metal wires have been determined in order to optimize energy dissipation without degrading speed and area of the system [9][10].

Based on the above experimental results the proposed network architecture characteristics are: a) segment length L1 b) Full population for Connection and Switch boxes c) F_c =1 for the input and the output connection boxes d) Disjoint Switch box with F_s =3, e) pass transistor with size ten times the minimum width as routing switch f) minimum width, double-spacing for the metal wires. In addition based on the application requirements the number of tracks in the routing channel was selected to be 20 and the array dimensions 8X8 CLBs. Finally, for the communication with the FPGA, 24 I/O pads were placed on each side of the FPGA. Each pad contains three I/O pins and each pin can connect to each one of the routing tracks with a configurable pass transistor.

2.4. Configuration Architecture

The proposed configuration architecture consists of the following components: the memory cell, where the programming bits are stored, the local storage element for each tile (a tile consists of one CLB, CB input-output, SB and the decoder which controls the configuration procedure of all the FPGA.

The memory cell is based on a typical 6T memory cell with all transistors having minimum size. This cell is provided with a reset mechanism to disable the switch to which it is connected by preventing the short-circuit currents that can occur in an FPGA if it is operated with unknown configuration states at start-up. Moreover since there is no need for high performance from the memory cell, neither the pre-charging technique nor a sense amplifier was used, resulting in energy savings. The energy consumption and the delay during the write procedure are shown in Table 1.

Table 1. Memory Cell Delay and Energy Consumption

Transition	Energy(fJ)	Delay(ps)
1→0	17.23	80.54
0→1	9.45	88.7

The configuration element consists of 480 memory cells because the tile requires 465 configuration bits, so there is an array of 30 columns and 16 rows. The 16 memory bits of a row compose a “word”. During the write procedure the configuration bits are written per “word” because we have a 16-bit write configurations bus. A 5-to-30 decoder is used in order to control which “word” will be written each time. The 5-inputs of the decoder are connected to the address bus. The decoder was implemented by using 5-input NAND and 2-inputs NOR gates because of the small number of the inputs. There is also a chip select signal. The NOR gates are used in order to idle the decoder when the chip select has value “0”. A pre-decoding technique was not used because of the increased area and energy consumption that it produces.

The write procedure can take part only in one storage element each time. All the other storage elements must be idle. That is why we use the chip select signal, which is managed by a 7-to-90 decoder. The whole FPGA needs 70 storage elements for a 8x8 array of CLBs (64 storage elements one for each tile, plus 6 storage elements for the periphery switch boxes and I/O pads). Additionally the configuration architecture supports partial reconfiguration. This means that each tile can be selected to be configured individually while the rest of the FPGA can be in process mode.

The configuration specifications are summarized as: a) 4.2KB size b) 16-bits data bus c) 12-bits address bus d) 1.4ns delay for writing a row of 16 memory cells e) 2100 cycles for entire 8×8 array configuration, f) partial reconfiguration.

2.5. FPGA Physical Implementation

A prototype full-custom FPGA was design in a 0.18μm STM process technology. The prototype features:

- 8x8 array size (320 LUTs, 320 FFs, 96 I/Os)
- 1.8 volts supply voltage

- 4.86x 5.28 mm^2 area
- 6 metal layer assignment
  - Metal 1 : Short connections, Power supply
  - Metal 2 : Short, intra-cluster, inter-cluster connections, buses, ground supply
  - Metal 3 : Intra-cluster, Main interconnections
  - Metal 4 : Clock signal, Configuration
  - Metal 5 : Configuration
  - Metal 6 : Configuration
- 2.94μsec configuration time
- RAM configuration
- Partial reconfiguration

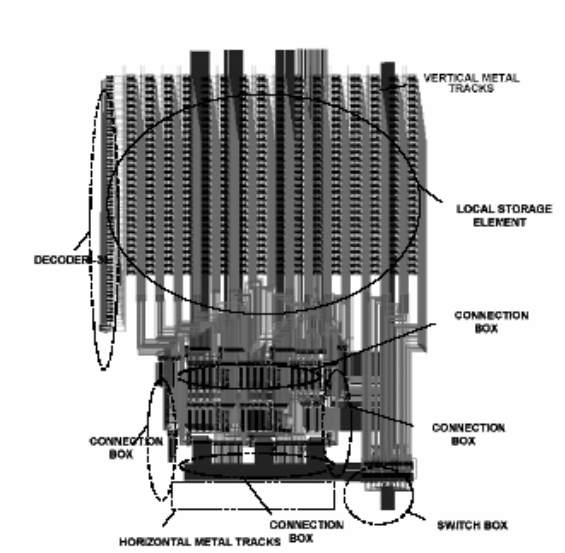

Figure 3. FPGA Physical Implementation

3. Proposed Design Flow

Equally important to an FPGA platform is a tool set, which supports the implementation of digital logic on the proposed FPGA. Therefore, such a design flow was realized. The input is the RTL-VHDL circuit description, while the output of the CAD flow is the bitstream file that can be used to configure the FPGA. Three different types of tools comprise the flow: i) non-modified existing tools, ii) modified existing tools, iii) and new tools. A brief description of the tools that compose the design flow is given below: a) **VHDL Parser**: performs syntax checking of VHDL input files b) **DIVINER**: is used as a synthesizer of RTL VHDL language c) **DRUID**: this tool is used to modify the EDIF [8] output file that is produced during the synthesis step, so that is can be used by the following tools of the design flow d) **E2FMT**: translates EDIF format to BLIF [6] e) **SIS**: is used for mapping the logic described in generic components (such as gates and arithmetic units) into the elements of the proposed fine-grain reconfigurable architecture [7] f) **T-Vpack**: reads in a BLIF format netlist of a circuit that has been technology-mapped to LUTs and flip-flops, packs the LUTs and flip-flops into the desired FPGA logic block, and outputs a netlist in VPR's netlist format [4] g) **DUTYS**: creates the architecture file of the FPGA that is required by T-VPack and VPR. h) **VPR:** is responsible for placement and routing of the target circuit in the FPGA i) **Power Model**: estimates power consumption j) **DAGGER:** generates the configuration bitstream for the fine-grain reconfigurable

hardware architecture. In addition there is a number of attractive features that characterize these tools which are technology independence, portability, modularity, compactness and easy of use [9][10].

3.1 Graphical User Interface

The Graphical User Interface (GUI) provides the designer with the opportunities to easily use all (or some of the tools) that are included in the proposed design flow. It consists of six independent stages: i) the File Upload, ii) the Synthesis, iii) the Format Translation, iv) the Power Estimation, v) the Placement and Routing and vi) the FPGA Program stage. Until now, there is no other academic implementation of such a complete graphical design chain. The main GUI advantage is the fact that it is friendly to the non-experienced designer who does not need to be familiar with the Linux OS. It is possible to run it from a local PC or through the Internet/Intranet, and the source code can be easily modified in order to add more tools. Regardless of the execution (locally or through the network) the proposed interface runs on the web-browser, and can program an FPGA that is attached to the user's PC.

4. Conclusions

This paper demonstrated the first complete system for implementing digital logic on a fine-grain reconfigurable platform. It includes the design of both the FPGA architecture and the complete design flow (from VHDL to bitstream) consisting entirely of academic tools, which allows the mapping of logic on the presented novel FPGA architecture. The presented FPGA architecture was designed and implemented in STM 0.18μm CMOS technology and future plans include redesign at 0.13. The obtained simulation results prove the attractive features of the proposed architecture. On the other hand, in contrast to commercial CAD systems, the proposed design flow can accomplish a FPGA design and is publicly available and very friendly to the non-experienced designers. Among our future plans is also the extension of some of the existing tools like SIS and T-VPack.

5. Acknowledgement

This work was partially supported by the project IST-34379-AMDREL which is funded by the European Commission.

6. References

[1] http://direct.xilinx.com/bvdocs/publications/ds003.pdf
[2] http://www.altera.com/products/devices/dev-index.jsp
[3] Betz V, Rose J and Marquardt A 1999 *Architecture and CAD for Deep Submicron FPGAs (*Kluwer Academic Publishers)
[4] http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html
[5] Betz V and Rose J 1997 *IEEE Custom Integrated Circuits Conference* pp.551-554
[6] Weste N and Eshraghian K 1993 *Principles of CMOS VLSI Design: A Systems Perspective* (Addison-Wesley)
[7] http://www.vlsi.ee.duth.gr/amdrel
[8] Betz V and Rose J 1999 *International Symposium on Field Programmable Gate Arrays* Monterey CA pp.59-68
[9] Kalenteridis V, Pournara H, Siozos K, Tatas T, Vassiliadis N, Pappas I, Koutroumpezis G, Nikolaidis S, Siskos S, Soudris D and Thanailakis A 2004 *J. Microprocessors and Microsystems*
[10] Kalenteridis V, Pournara H, Siozos K, Tatas K, Pappas I, Nikolaidis S, Siskos S, Soudris D J and Thanailakis A 2004 *11th Reconfigurable Architectures Workshop RAW* (Santa Fi New Mexico, USA, 26-27 April 2004)
[11] Vassiliadis N, Nikolaidis S, Siskos S and Soudris D J 2003 *13th International Workshop PATMOS 2003 (Turin Italy September 2003)* pp. 607-16