Microprocessors and Microsystems 37 (2013) 693–700

Contents lists available at SciVerse ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

# ELEON3LP – Superscalar and low-power enhancements of single issue general purpose processor model

Krzysztof Marcinek *, Witold A. Pleskacz

*Institute of Microelectronics and Optoelectronics, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warsaw, Poland*

ARTICLE INFO

*Article history:*
Available online 4 July 2012

*Keywords:*
Superscalar processor
Low-power
High-performance
Power management
LEON3
DSM

ABSTRACT

Low power consumption and high-performance are the most important factors in modern embedded System-on-Chip (SoC) designs. Increasing computation complexity and incessant growth of clock frequency reveals the necessity for dynamic and smart utilization of the available hardware resources. The paper presents the modified LEON3 processor IP core as an exemplary process of enhancing single issue general purpose processor with superscalar abilities and low-power features. The results of this work can be applied to many existing general purpose processor models to achieve low-power and high-performance systems suitable for modern embedded applications. In comparison with the original LEON3 IP core, the new one may execute up to two instructions per cycle and dynamically manage incorporated power domains. The Enhanced LEON3 IP core was synthesized for 500 MHz using UMC 90 nm CMOS technology. Performed gate level VCD-based (Value Change Dump) power estimation shows that combining superscalar and low-power techniques allows performing faster program execution with less energy consumption than the original design.



## 1. Introduction

Power consumption in SoC Integrated Circuits (ICs) can be divided into two categories. The first one is the dynamic power consumption caused by performing useful operations and it is proportional to the number of transistor state changes per second. Clock gating [1] is a well-known low-power technique used to reduce dynamic power consumption in unused system blocks. The easiest way to reduce dynamic power in working functional units is to decrease the clock frequency. However, the clock frequency defines the system performance, so nowadays it cannot be taken into consideration in many applications demanding high speed processing or throughput. Superscalar processing [2] is one of the most exploited approaches to maintain system performance with lower clock frequency, which is essential for reducing dynamic power consumption for low-power applications. The second one, the static power consumption, is the result of the subthreshold leakage current and it is dissipated even if the device has no clock applied. Although in the past days static power could be ignored, in contemporary deep sub-micron (DSM) technologies it plays a significant role in the total power consumption [3]. Therefore, it is essential to incorporate power management unit [4] utilizing low-power techniques involved in reducing both, dynamic and static power.

A typical superscalar processor performs more than one instruction during the same clock cycle, which means faster program execution and minor demands for operating frequency. By using superscalar and low-power techniques one can choose to work with lower clock frequency and, as a result, lower average power consumption. On the other hand, high frequency and superscalar execution is suitable for more exigent applications. Studies in superscalar processors [5] show that for a significant amount of time particular functional units (FUs) remain idle consuming static power. Power gating [6] is one of the most commonly used low-power techniques in order to avoid this power loss. While cutting off power supply of the unused FU, the source of all parasitic currents is removed. Exploiting superscalar and multiple power domain system results in spare time when the whole power domain can be shut down reducing power consumption, not only by the dynamic but also by the static power part.

The main goal of this work is to present a process of enhancing single issue general purpose processor with superscalar and low-power features on an example of a well-known open-source LEON3 IP core processor [7]. LEON3 is available in the form of a synthesizable VHDL model based on SPARC V8 instruction set architecture (ISA). Although this work is focused on a particular processor system, the design process of superscalar and low-power enhancements can be applied to other processor models adjusting them to modern technology requirements.

* Corresponding author.
*E-mail addresses:* K.Marcinek@imio.pw.edu.pl (K. Marcinek), W.Pleskacz@imio.pw.edu.pl (W.A. Pleskacz).

0141-9331/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2012.06.004

The paper is organized as follows. The microarchitecture of Enhanced LEON3 processor IP core is presented in Section 2. Section 3 presents ELEON3LP pipeline. The following Section 4 shows simulation results. Section 5 describes low-power modifications. Synthesis results and power considerations are presented in Section 6. The paper ends with conclusions.

## 2. ELEON3LP processor microarchitecture

The LEON3 processor system allows the user to customize it for certain applications due to its configuration abilities. The system is based on the LEON3 core connected through an AMBA bus to the system peripherals. For the purpose of this work the core was configured to incorporate separated data and instruction cache controllers with AMBA bus interface, interrupt port, hardware divide and multiply unit. The register file is a three-port memory with separated write port and two read ports. It contains general purpose global registers and register windows, whose number is limited by the SPARC V8 standard. The integer unit (IU) datapath is based on a seven-stage pipeline. For this research, a grlib-gpl-1.0.17-b2710 version of GRLIB was used. In order to handle the execution of two instructions at the same time, the original register file was rewritten to enable reading of four and writing of two independent data words at the same time. The instruction cache controller was modified to enable fetching of two consecutive and valid instructions. During cache miss, however, the instructions are fetched from an external memory through the AMBA interface. In this case, only the *Primary IU* has the data to handle and the *Secondary IU* executes the NOP (no operation) instruction. The *Superscalar Controller* is a stand-alone entity that extends the original IU pipeline with the additional *Prefetch* stage (Fig. 1). It is situated between the instruction cache controller and both integer units [8]. The decision of using a synchronous logic rather than the combinational one was caused by possible critical path occurrence. The future work on extending the number of integer units and dispatch algorithms was also taken into consideration. The consequence of this decision will be described in the next section. After fetching two consecutive instructions, the *Prefetch* stage decides whether the instructions can be executed in parallel or one after another. In the first case, the fetched instructions are directed to the appropriate integer units. In the second case, only the first instruction feeds the *Primary IU* pipeline, while the *Secondary IU* remains unused. The second instruction is fetched again in the following cycle. The original VHDL source code of IU was changed in order to be used either as *Primary IU* or *Secondary IU*. However, only the *Primary IU* is connected to the instruction cache controller and data cache controller and it has access to the processor's internal registers (SPARC V8 ISA registers: PSR, Y, ...). Therefore, it is the

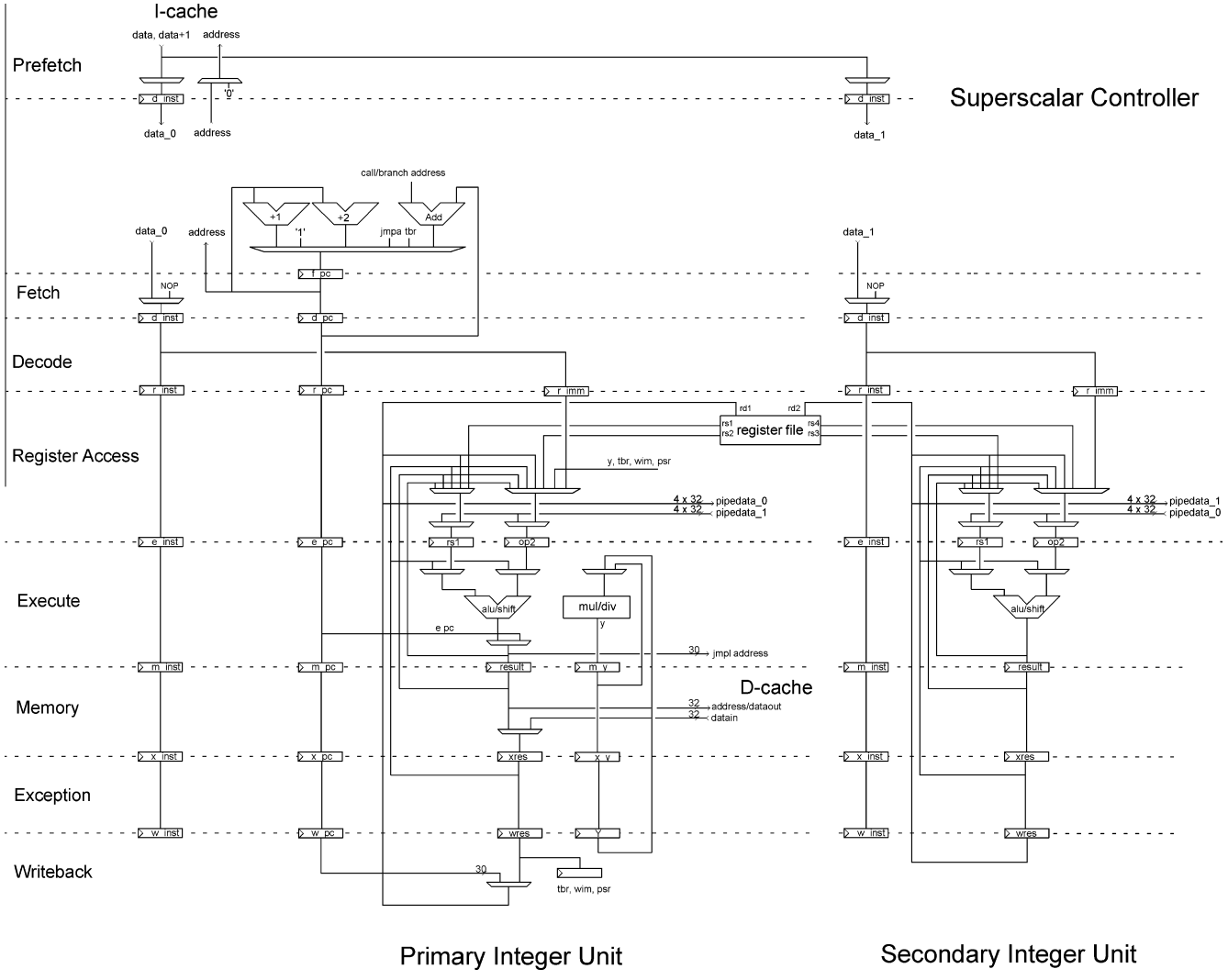


**Fig. 1.** Enhanced LEON3LP integer unit datapath diagram.