Study on Optimized Design of CPU Architecture Based Real-Time Scheduling and Pipeline Registers - PhD thesis summary

This paper was supported by the project "Sustainable performance in doctoral and post-doctoral research PERFORM–Contract no. POSDRU/159/1.5/S/138963", project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

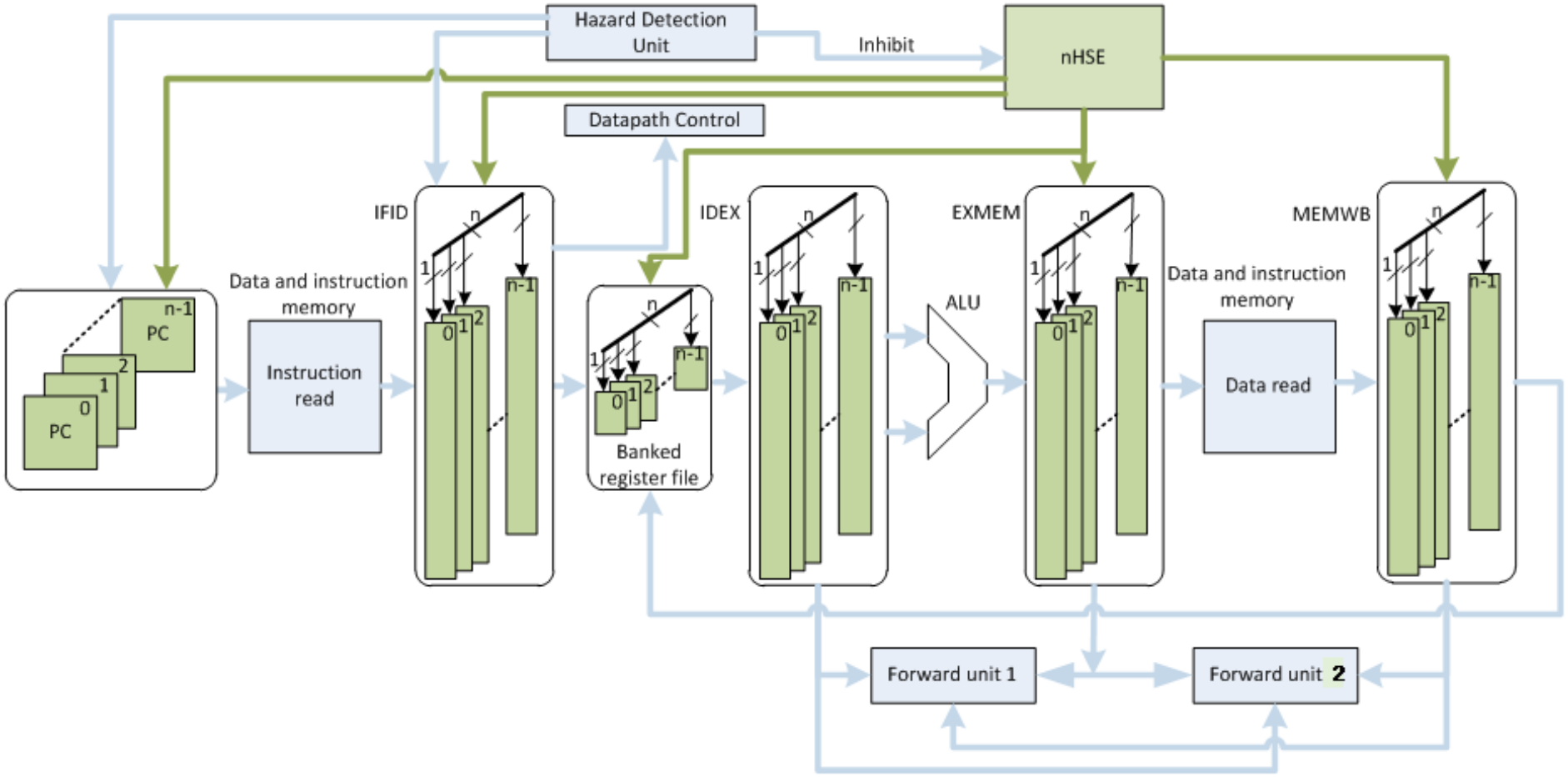
Elena-Eugenia Moisuc (Ciobanu)

Ştefan cel Mare University of Suceava

Faculty of Electrical Engineering and Computer Science

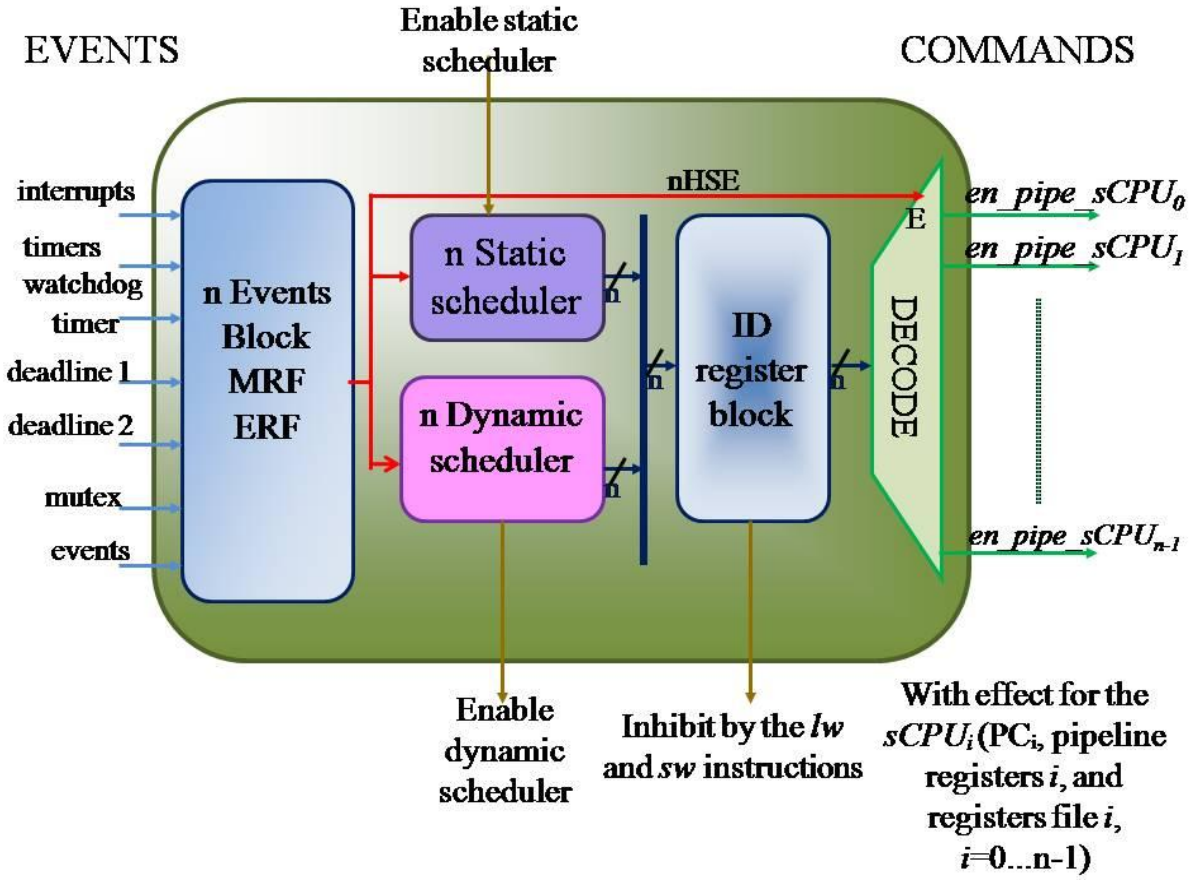
THE STUDY PRESENTS

- A functional architecture of the processor
- MPRA: a multi-pipeline architecture ⟹ each task has its own set of pipeline registers
- Interrupt handling (the first part of the study) ⟹ the MPRA interrupts are viewed as tasks.



nMPRA

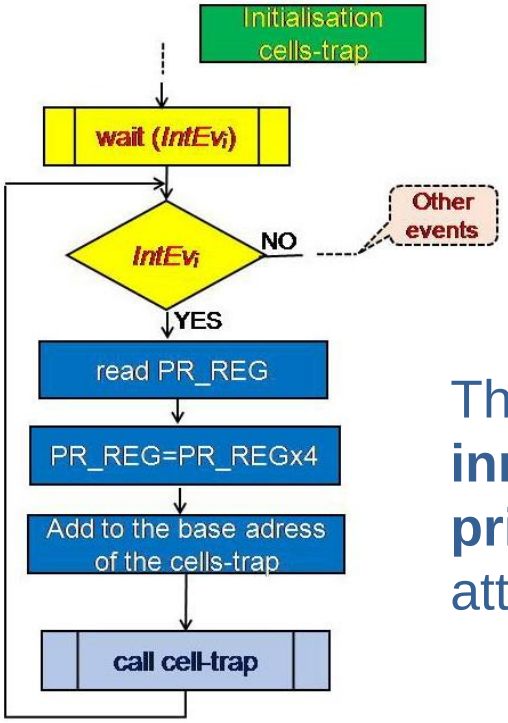
MPRA multiplied by *n* times ⟹ *nMPRA*



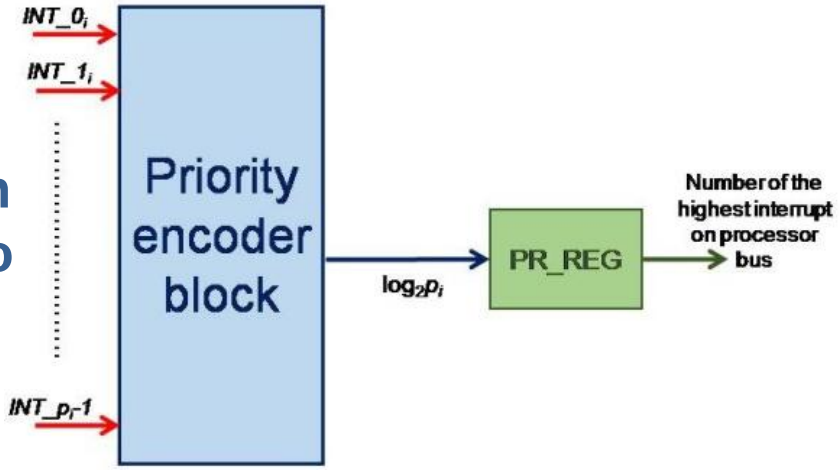
- **HSE** unit (***H**ardware **S**cheduler **E**ngine*) is integrated into the processor.

- **FPGA** (***F**ield **P**rogrammable **G**ate **A**rray)* devices offer integrated elements of the complexity of the possible application oriented integrated circuits (ASIC - *Application Specific Integrated Circuit*), with **the advantage of programmability** or better said, of **configurability**.
- The architecture will be analyzed and implemented in **VHDL** with **FPGA Virtex6 a ML605** reference board produced by Xilinx.

CONCLUSIONS



The study proposed **an innovative solution to prioritize interrupts** attached to a task.



- there aren't specialized interrupts controller;
- interrupts inherit the priority task (*sCPU*);
- interrupt attached to a task can interrupt a lower-priority task but cannot interrupt the execution of the task who is attached, or higher-priority task;
- interrupt can be a task;
- all interrupts can be attached to a single task;
- interrupt doesn't reset another sCPUi pipeline; the interrupts can be nested;
- interrupts priorities can be dynamic (by reattaching to a task, or by changing the priority of the task to which it is attached).

UNIVERSITAS GALATIENSIS

UGAL INVENT

Prima ediţie a Salonului "UGAL INVENT" 2014
Universitatea „Dunărea de Jos" din Galaţi, România
8 – 10 Octombrie 2014

Universitatea Ştefan cel Mare Suceava

STUDIU ASUPRA ARHITECTURII OPTIMIZATE A UCP BAZATĂ PE PLANIFICARE REAL-TIME ŞI REGIŞTRI PIPELINE

Elena-Eugenia Moisuc (Ciobanu)
Universitatea Ştefan cel Mare Suceava, Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor
Str. Universităţii, nr. 13, 720229 Suceava, România

STUDY ON OPTIMIZED DESIGN OF CPU ARCHITECTURE BASED REAL-TIME SCHEDULING AND PIPELINE REGISTERS

ABSTRACT

The most effective interrupt handling is by hardware, which increases the speed of response to external events and reduces overload. Therefore, the hardware method is used in most of the RTOS. Analyzing traditional models for interrupts management found their inability to provide the temporal determinism required in real time systems. The study refers to the architecture of a real time scheduler implemented in hardware, which eliminates the override feature of the operating system and times determined by contexts' switching. First, the study proposes *Multi Pipeline Register Architecture*, which uses a unified space of priorities for tasks and interrupts, so there must be no specialized interrupts controller.

INTRODUCTION

The study presents a functional architecture of the processor, which provides very low switching times of contexts because of the way it is designed. This processor, further research of *[illegible]*, "Custom designed CPU architecture based on a hardware scheduler and independent pipeline registers – architecture description", *IEEE 35th Jubilee International Convention on Information and Communication Technology, Electronics and Microelectronics, Croatia, May 2012* and *[illegible]*, "Custom designed CPU architecture based on a hardware scheduler and independent pipeline registers – concept and theory of operation", *2012 IEEE EIT International Conference on Electro-Information Technology, Indianapolis, IN, USA, 6-8 May 2012, [illegible]*, is capable of performing automatic switching contexts and to launch the new task in a range between 1 and 3 clock cycles. The processor implements a multiplexed structure resources pipeline registers and file registers and relies on integrated hardware scheduler. The MPRA is a multi-pipeline architecture, which means that each task has its own set of pipeline registers. HSE unit (*Hardware Scheduler Engine*) is integrated into the processor. The MPRA interrupts are viewed as tasks. The first part of the study concerned the interrupts' handling. The used architecture is *nMPRA* – MPRA multiplied by *n* times (*[illegible]*, "CPU Architecture based on a Hardware Scheduler and Independent Pipeline Registers", *submitted to IEEE Transactions on VLSI System, 2014*), presented in Fig. 1. Because each task has its own set of pipeline registers and its own set of general registers, context' switching can be done in one processor cycle, and the response to an external event may be delayed up 1.5 processor cycles. For these reasons, the architecture is very fast. All tasks share other resources. An instance of this process we'll call it semi CPU for the task *i* (sCPU*i*) and performs a single task.

DEVELOPEMENT

The nHSE (Fig. 2) has as input such as events like interrupts, deadline, watchdog timers, timers, mutexes, message events, self-support event execution, as well as enabling signals of static and dynamic schedulers and inhibiting the execution of load and store instructions, generating sCPU activation signals.

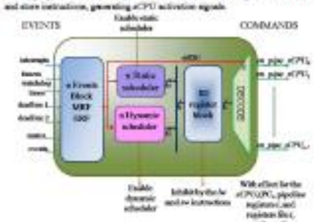


Fig. 2. nHSE Architecture

HSE enables activation or deactivation system interrupts. Interrupts follow the same procedure execution as tasks so that enabling or disabling their execution is done using the same instructions that are addressed to tasks. In the nMPRA, each timer associated with a task can be configured to generate an interrupt when the time allocated to the task is nearing completion. A task can respond to an external event if the event is masked by blocking wait instruction.

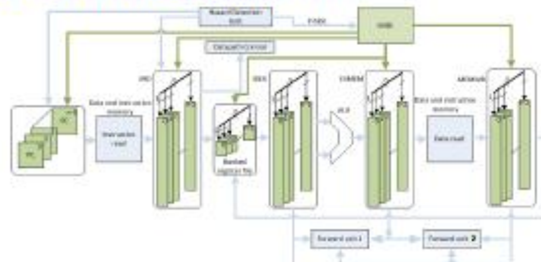


Fig. 1. nMPRA
PC program counter, IF/ID Instruction Fetch Instruction Decode stage, ID/EX Instruction decode execute stage, EX/MEM execute memory stage, MEM/WB memory write back stage

The wait instruction is very powerful because it allows synchronizing the execution while multiple events. Under software control, following biased sCPU*i* (task *i*) tasks, these events are treated and paid. Interrupts in our model are treated as events that are attached to real-time executive or tasks, borrowing in this way the priority of the tasks to which they are attached. The kernel must be a real-time monitor type. Once entered into the monitor, this one cannot be interrupted (interrupts are disabled). As a result, the monitor functions must be short, without interactions and sites. The interrupts attached to a task can interrupt only strictly lower priority tasks, which are into the running state.

The solution is faster (Fig. 3), but requires an additional hardware block (Fig. 4), whose complexity is given by the total interrupts of the processor.

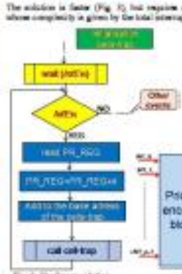


Fig. 3. Hardware solution

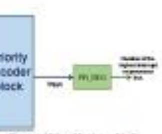


Fig. 4. Additional hardware block

IMPLEMENTING THE ARCHITECTURE

FPGA (*Field Programmable Gate Array*) devices offer integrated elements of the complexity of the possible application oriented integrated circuits (ASIC – *Application Specific Integrated Circuit*), with the advantage of programmability or better said, of configurability. FPGA devices allow the design of specialized hardware architectures with the advantage of flexibility programmable environment that is implemented. Currently, FPGA devices are widespread with equivalent capacity in logic gates that exceed the millions of logic gates at a low cost price. For this reason, this paper proposes an implemented hardware support for RTOS functionalities based on the FPGA systems.

The architecture will be analyzed and implemented in VHDL with FPGA Virtex5 a ML605 reference board (Fig. 5) produced by Xilinx.

Fig. 5. ML605 reference board

In the next period will implement in VHDL: control unit, data memory and instruction, hazard detection unit, forwarding unit, ALU, multiplexers, registers PC, pipeline registers, file registers.

CONCLUSIONS

The presented paper defines new original functionalities for the nMPRA and nHSE. It proposed an innovative solution for attachment the interrupts to the tasks (sCPU). In this context, the interrupt leads the task's priority and behavior. Such interrupts' behavior is more predictable in the context of real-time application (a task cannot be interrupted only interrupts attached a more priority task). The proposed scheme has some powerful and interesting features such as: there aren't specialized interrupts controller; interrupts inherit the priority task (sCPU), a task can attach anyone, several, or all the *p* system interrupts, programmer sets the priority of the interrupts attached to the same task; interrupt attached to a task can interrupt a lower priority task but cannot interrupt the execution of the task who is attached, or higher priority task; the interrupt may be attached to a single task, interrupt can be a task; all interrupts can be attached to a single task; interrupt doesn't reset another sCPU pipeline; it doesn't require saving and restoring the context; the interrupts can be nested; interrupts priorities can be dynamic (by reattaching to a task, or by changing the priority of the task to which it is attached). It proposed an innovative solution to prioritize interrupts attached to a task. Unlike loop testing solution disruptions, the proposed solution provides a uniform response time for any interruption if the time is the only active interrupt. In addition, the proposed solution also provides fixed prioritized interrupts. We can say that the presented solution contains a unitary interrupts management, and a hardware solution to attach the interrupts to the task of the hardware RTOS.

In the future, we will explore ways to create the priority encoder block depending on the number of interrupts attached and possibly the direct hardware upload to PC with the interrupts' handler address.

ACKNOWLEDGEMENT
This paper was supported by the project "Sustainable performance in doctoral and post-doctoral research PERFORM - Contract no. [illegible]", project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

CUPRINS

UGAL INVENT
The first event supporting innovation promoted by "Dunarea de Jos" University of Galati
Galați, România, 8 – 10 October 2014
www.invent.ugal.ro

Organized by

INVENTION CATEGORY 2

Informatics – Computers – Electronics – Electricity – Communication devices

„Investește în oameni!"
Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.
Axa Prioritară 1 „Educația și formarea în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere"
Domeniul Major de Intervenție 1.5 „Programe doctorale și post-doctorale în sprijinul cercetării"

UGAL INVENT

The first event supporting innovation promoted by "Dunarea de Jos" University of Galati

Galați, România, 8 – 10 October 2014

www.invent.ugal.ro

Organized by

Invention Registration FORM

Company Name / Institution:	Ștefan cel Mare University of Suceava, Faculty of Electrical Engineering and Computer Science
Address:	Street Universitatii, no. 13, 720229 Suceava, Romania
Phone:	-
Fax:	-
Invention title:	Study on Optimized Design of CPU Architecture Based Real-Time Scheduling and Pipeline Registers
Authors:	Moisuc (Ciobanu) Elena-Eugenia
Main author status:	PERFORM Project
Patent no.:	-
Contact person	Moisuc (Ciobanu) Elena-Eugenia
E-mail:	neli.ciobanu@eed.usv.ro
Description of the invention:	The most effective interrupt handling is by hardware, which increases the speed of response to external events and reduces overload. Therefore, the hardware method is used in most of the RTOS. Analyzing traditional models for interrupts management found their inability to provide the temporal determinism required in real-time systems. The study refers to the architecture of a real-time scheduler implemented in hardware, which eliminates the override feature of the operating system and those determined by contexts' switching. First, the study proposes *Multi Pipeline Register Architecture*, which uses a unified space of priorities for tasks and interrupts, so there must be no specialized interrupts controller.
Image/photo:	-
Invention category:	2

Invention categories: 1. Mechanics – Motors – Machines – Equipment – Industrial procedures – Metallurgy. 2. Informatics – Computers – Electronics – Electricity – Communication devices. 3. Sanitation facilities – Ventilation facilities – Heating and cooling installations. 4. Agriculture – Horticulture – Gardening. 5. Medicine – Surgery – Orthopaedics. 6. Teaching methods and materials. 7. Means of transportation – Vehicles – Ships – Airplanes. 8. Food – Beverages – Cosmetics – Hygiene materials – Medication. 9. Sport - Leisure. 10. Publicity – Printing – Packages – Packaging. 11. Environment protection – Energy