Hardware Event Handling in the Hardware Real-Time Operating Systems

Elena-Eugenia (CIOBANU) MOISUC
Alexandru-Bogdan LARIONESCU
Ioan UNGUREAN
Department of Electrical Engineering and Computer Science
Stefan cel Mare University Suceava, Romania

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.

18th International Conference on System Theory, Control and Computing
October 17-19, 2014, Sinaia, Romania

INTRODUCTION

Events in RTS (*R*eal *T*ime *S*ystems):

- time events
- watchdog timer events
- deadline events
- interrupt events
- mutex events
- synchronization and inter-task communication events

PROBLEM #1

RTOS FOR EMBEDDED SYSTEM

Can’t allow a task to synchronize simultaneously with multiple events utilized for resource sharing, synchronization and communication between tasks such as semaphores, flags, mutexes, signals, events, messages.

PROBLEM #2

GENERAL-PURPOSE PROCESSORS FOR EMBEDDED SYSTEMS

Inefficient power consumption and non-deterministic performances ⟶ Use of an oversized platform ⟶ are not suitable for embedded systems with low power consumption requirements and real-time capabilities

SOLUTIONS

Hardware Event Handling in *nMPRA*

- **Section II –** the *nMPRA*;
- **Section III –** the hardware events handling;
- **Section IV –** PC_i modified architecture;
- **Section V –** the final conclusions.

THE NOVELTY

The new hardware implementation of the event selection mechanism using trap

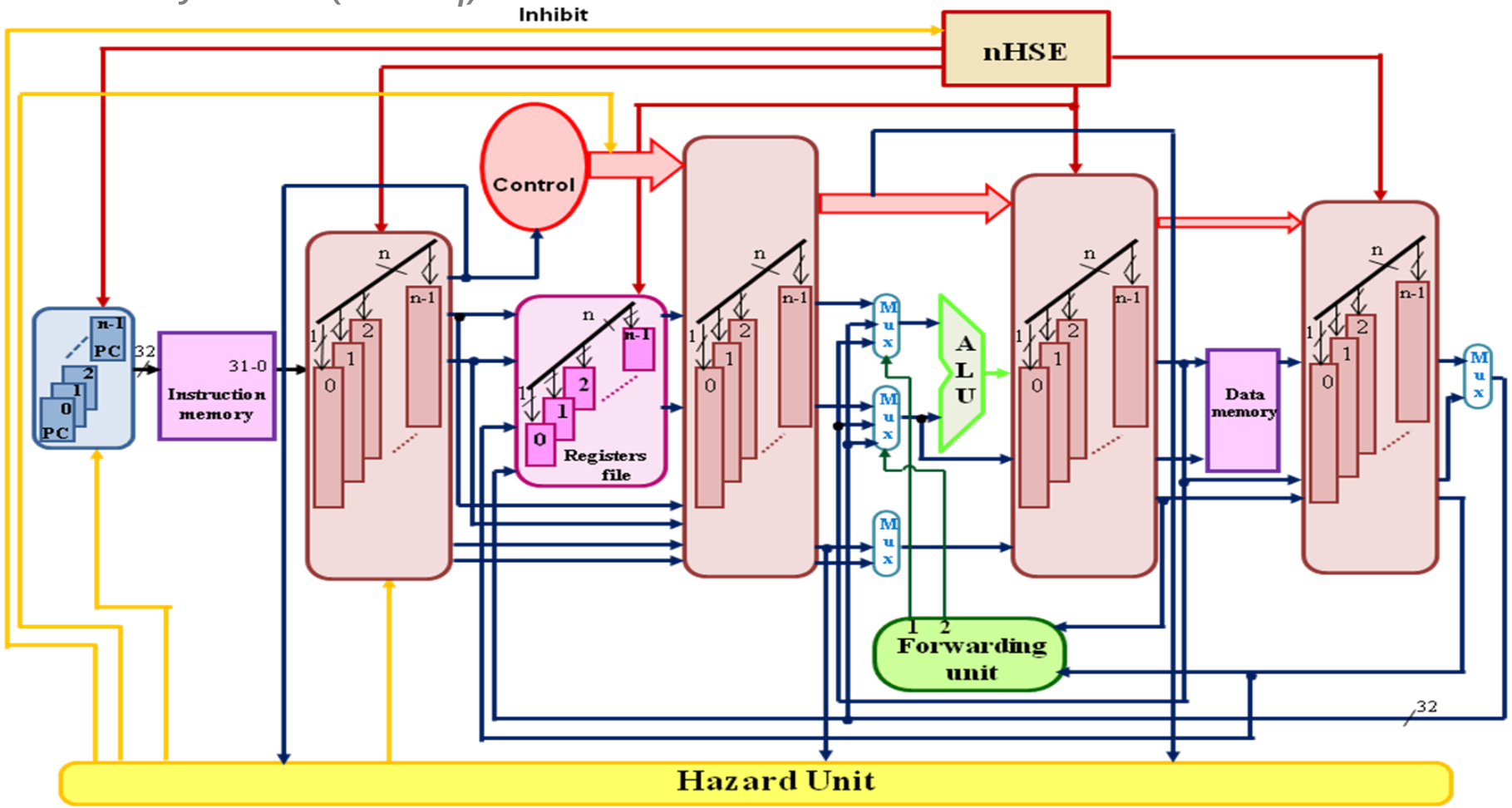
- Using a hardware prioritization scheme for the *n-task* Multi Pipeline Register Architecture (***nMPRA***)

The architecture of PC register and the introduction of the *ret_esr* (*return from the event service routine*) instruction

- Corresponding $sCPU_i$ (PC_i)

nMPRA

- a hardware implemented support for event prioritization and treating
- MPRA ⟹ n-task MPRA (*nMPRA*): to replicate the pipeline registers; to create multiple instances of the CPU ⟹ *semi CPU* for every *task i* ($sCPU_i$)



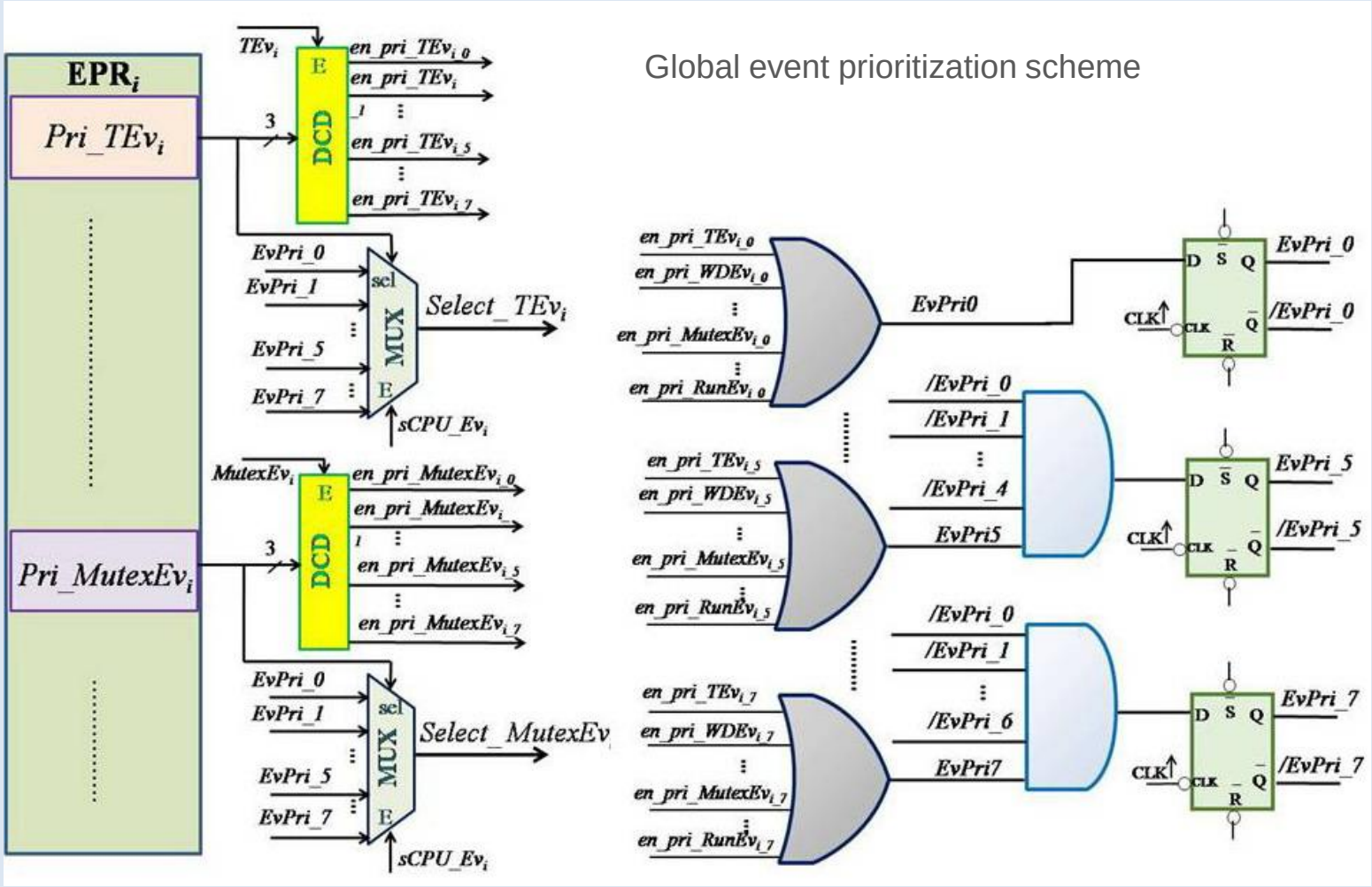
The *nMPRA* ⟶ an original hardware structure utilized for task scheduling, both static and dynamic, and provides unitary events management.

WHAT WE AIM FOR

- to improve through hardware the performances of RTOS for microcontrollers
- to switch faster between the tasks
- to improve the response time to external events
- to improve the behavior of the interrupts, which are treated as events in this case
- to offer several types of inter-process communication primitives (messages, mutexes, and so on)

Global Events Prioritization

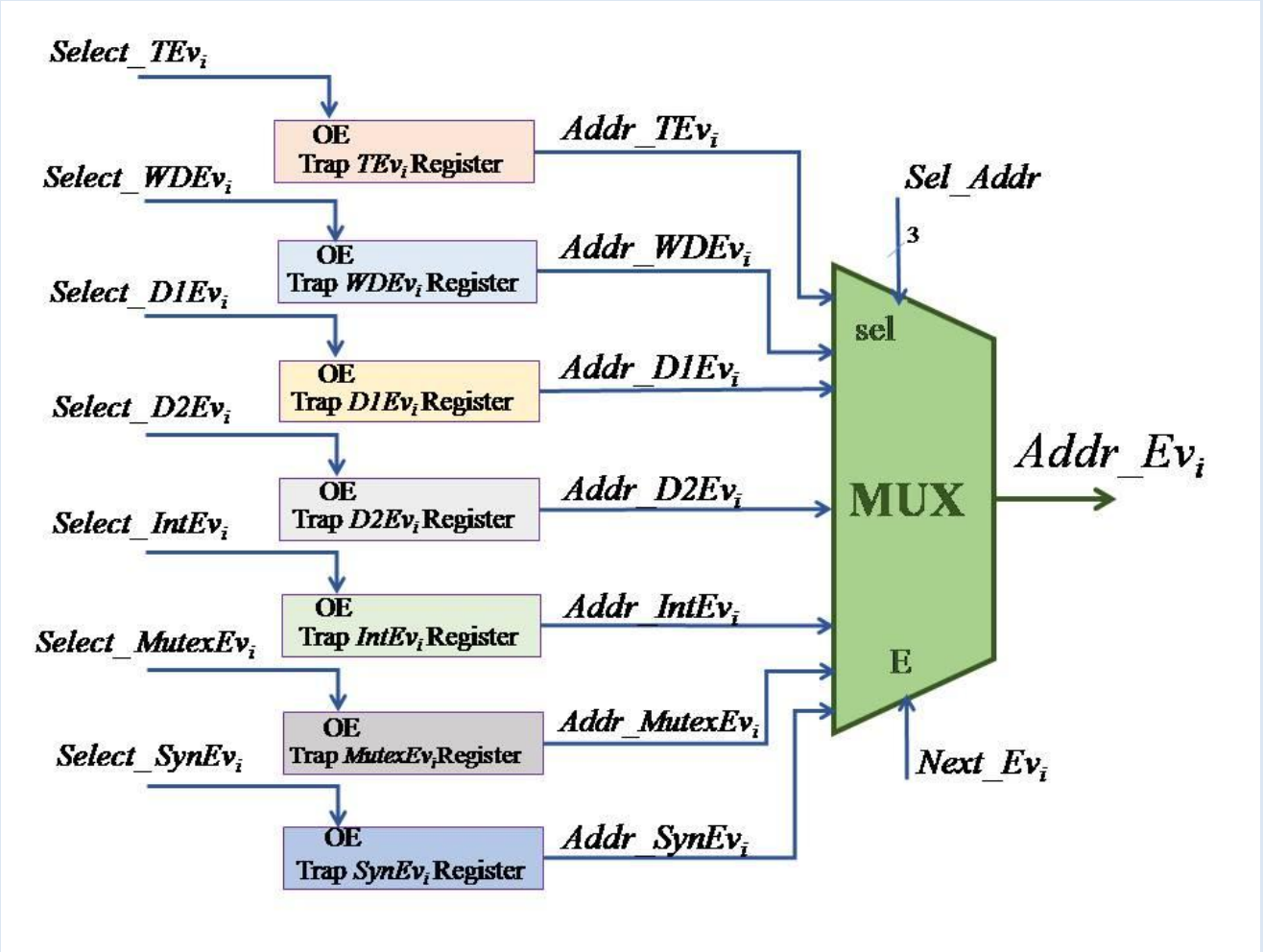
- every $sCPU_i$ has attached an **E**vent **P**riority **R**egister (EPR_i), which contains the priority level of each type of event associated with that $sCPU_i$
- the priority is different for each type of events



- The prioritization scheme selects the active event category with the highest priority, in order to be treated

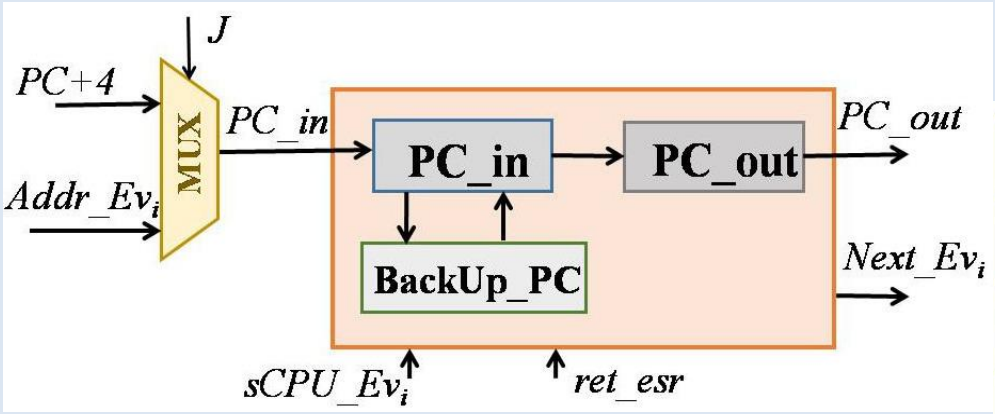
- The priority level of each event category can be static or it can be changed dynamically, depending on the system's demands.
- If there are multiple active events in the selected category, it must be done another selection of the event from that category that will actually be treated first.
- This second selection depends on the type of the event category.

Selection of the highest priority event handler address



- It obtains the address of the event service routine associated with the selected event (the event service routine is executed by the $sCPU_i$ task)

PROGRAM COUNTER ARCHITECTURE



Program Counter architecture

Event vector registers

address of the *Timer* event servicing routine	Trap TEv_i Register
address of the *WatchDog* event servicing routine	Trap $WDEv_i$ Register
address of the *Deadline1* event servicing routine	Trap $D1Ev_i$ Register
address of the *Deadline2* event servicing routine	Trap $D2Ev_i$ Register
base address of the trap-cells table for interrupts	Trap $IntEv_i$ Register
address of the *Mutex* events servicing routine	Trap $MutexEv_i$ Register
address of the *Syn* events servicing routine	Trap $SynEv_i$ Register

- an event occurs ⟹ the associated task become ready for executions (it has a higher priority that the task in the execution state) ⟹ the PC_i (corresponding $sCPU_i$), is set by the content of the trap register (the address of the associated routine for the event occurred)

CONCLUSIONS

- The advantage of the hardware treating of the events is reducing the time necessary to identify the source of the event and to launch the appropriate event servicing routine.
- The prioritization scheme is simple and can be applied to all the events.
- It permits the introduction of new events updating the global event prioritization scheme and inserting a new trap register for each new event category.

FUTURE PAPERS

- The optimization of the implementation for mutex events and synchronization and communication primitives

THANK YOU!

18th International Conference on System Theory, Control and Computing

Joint Conference SINTES 18, SACCS 14, SIMSIS 18

17 - 19 October 2014, Sinaia, Romania

Mrs. Elena-Eugenia Moisuc (Ciobanu)
Stefan cel Mare University, Suceava
nell.ciobanu@eed.usv.ro
720229 Suceava
Romania

July 4, 2014

Dear Mrs. Elena-Eugenia Moisuc (Ciobanu),

On behalf of the Program Committee, it gives me great pleasure to invite you to participate in *the 18th International Conference on System Theory, Control and Computing ICSTCC 2014* which will be held at the Rina Sinaia Hotel, Sinaia, ROMANIA, during October 17 - 19, 2014.
The *ICSTCC 2014* is technically co-sponsored by the IEEE Control Systems Society (CSS). The Proceedings will be published in the *IEEE Xplore Digital Library* and will be submitted for indexing in the *Conference Proceedings Citation Index*.

Your paper submitted to the *ICSTCC 2014* has been accepted for presentation by the conference. As indicated in the notification letter sent to you about your paper's acceptance, at least one author of your paper must attend the conference to present the paper. We hope that you will participate in this scientific meeting.
Acceptance of your paper for presentation does not, in any way, financially oblige *ICSTCC 2014* for the expenses incurred by you to travel and attend the conference. If you have any questions, please contact us at icstcc2014@ac.tuiasi.ro.

WARNING: Depending on your citizenship, you may require visa to enter Romania. For additional information about visa and travel authorization, please visit the following website: http://www.mae.ro/en/node/2040

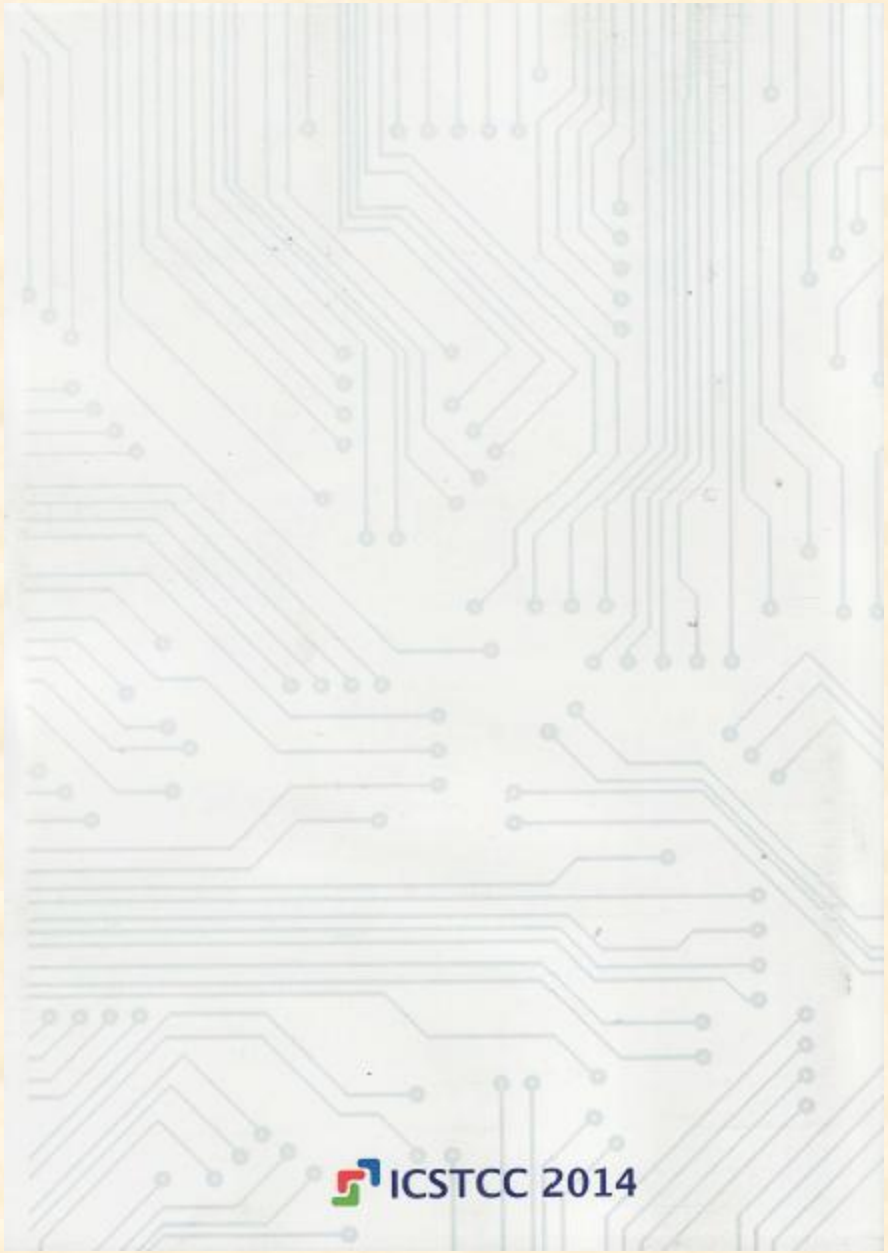
Thank you in advance for your participation. I look forward to seeing you in Sinaia.

Sincerely,
Prof. Mihail Voicu, General Chair of the ICSTCC 2014

Accepted Paper details:

Elena-Eugenia Moisuc (Ciobanu), Alexandru-Bogdan Larionescu, Ioan Ungurean, "Hardware Event Handling in the Hardware Real-Time Operating Systems."


2014 18th International Conference on
ICSTCC
System Theory, Control and Computing
Joint Conference
SINTES 18 SACCS 14 SIMSIS 18
PROGRAM & BOOK OF ABSTRACTS
technically
co-sponsored by
IEEE
CSS
ORGANIZERS
October 17-19, 2014
SINAIA - ROMANIA


ICSTCC 2014

18th International Conference on System Theory, Control and Computing

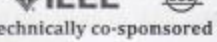
(Joint conference SINTES 18, SACCS 14, SIMSIS 18)

October 17 – 19, 2014
Sinaia, ROMANIA

Editors:
Mihaela Hanako Matcovschi
Lavinia Ferariu
Florin Leon

Organizers:

- "Gheorghe Asachi" Technical University of Iaşi
  Faculty of Automatic Control and Computer Engineering
- University of Craiova
  Faculty of Automation, Computers and Electronics
  Automatic Control Research Centre
- "Dunărea de Jos" University of Galaţi
  Faculty of Automatic Control, Computers, Electrical and Electronics Engineering



Technically co-sponsored by
IEEE – CSS Control System Society

Strîmbeanu, Daniel — Univ. of Craiova

This paper presents the sensorial system for structure of a backbone hyper-redundant arm with an electro-pneumatically system for position control. A system of cables actuated by DC motors is used for bending. Control system is based on a PIC microcontroller. The position of the robot can be obtained by bending it with the cables and by blocking the position of the elements we need, using the electro-pneumatically system. The major advantage of this type of actuation consists in the fact that the robot can be actuated using a boundary control by cables, the position blocking system for any element being relatively simple. The sensorial system is described and the main features of the global system are presented. The advantages of this sensorial system for this robot architecture are discussed

12:30-12:50 — FrA2.5

Hardware Event Handling in the Hardware Real-Time Operating Systems

Moisuc (Ciobanu), Elena-Eugenia — *Stefan cel Mare* Univ. of Suceava
Larionescu, Alexandru-Bogdan — *Stefan cel Mare* Univ. of Suceava
Ungurean, Ioan — *Stefan cel Mare* Univ. of Suceava

An important issue related to Real-Time Operating Systems is the handling of interrupts, timers, mutexes, watchdog timers, synchronization and communication directives as unitary events. In order to obtain a predictable response time for the different types of events that occur simultaneously, it is necessary to have a prioritization mechanism for them. Therefore, this paper proposes a hardware mechanism for handling the events enumerated above, in order to improve the software solution, which is not efficient because it generates delays. The method is implemented in n-tasks Multi Pipeline Register Architecture, that is a microcontroller architecture with Real-Time Operating Systems capabilities implemented in hardware, which allows switching time between tasks of one processor cycle and a response time to events of up to 1.5 processor cycles.

FrA3 — Carpati

Fractional Order Modeling and Control (Invited Session)

Chair: Copot, Cosmin — Univ. of Ghent
Co-Chair: Muresan, Cristina Ioana — Tech. Univ. of Cluj-Napoca
Organizer: Copot, Cosmin — Univ. of Ghent
Organizer: Muresan, Cristina Ioana — Tech. Univ. of Cluj-Napoca

11:10-11:30 — FrA3.1

Stabilizing Control Strategies: A Comparison between the Fractional Order Controller and the IMC (I)

Folea, Silviu — Tech. Univ. of Cluj-Napoca