



Universitatea “Ștefan cel Mare” Suceava

Facultatea de Inginerie Electrică și Știința Calculatoarelor

IMPLEMENTAREA ȘI ANALIZA COMPARATIVĂ A PERFORMANȚELOR ARHITECTURII HARDWARE PROPUSE PENTRU SOTR

-raport de cercetare nr. 3-

Îndrumător:

Profesor univ. dr. ing. Vasile Gheorghiuță GĂITAN

Candidat:

ing. Ionel ZAGAN

16 decembrie 2016

CUPRINS

1. Introducere
2. Implementarea procesorului nMPRA și a planificatorului dinamic nHSE
3. Implementarea mecanismelor de sincronizare și comunicație inter-task
 - 3.1. Implementarea mecanismului de sincronizare prin mutex-uri
 - 3.2. Implementarea mecanismului de comunicație inter-task
4. Planificarea task-urilor cu arhitectura de timp real nMPRA-MT
5. Costuri de implementare ale arhitecturii nMPRA
 - 5.1. Necesarul de resurse pentru implementarea arhitecturii nMPRA și a proiectului SoC
 - 5.2. Analiza puterii consumate
6. Criterii de performanță. Comparații cu alte implementări de procesor
7. Direcții viitoare de cercetare
8. Concluzii
9. Publicații

1. Introducere

Proiectul System-on-Chip dezvoltat cu kit-ul de dezvoltare cu **FPGA Virtex-7** include procesorul nMPRA ce se bazează pe multiplexarea resurselor multiplicate.

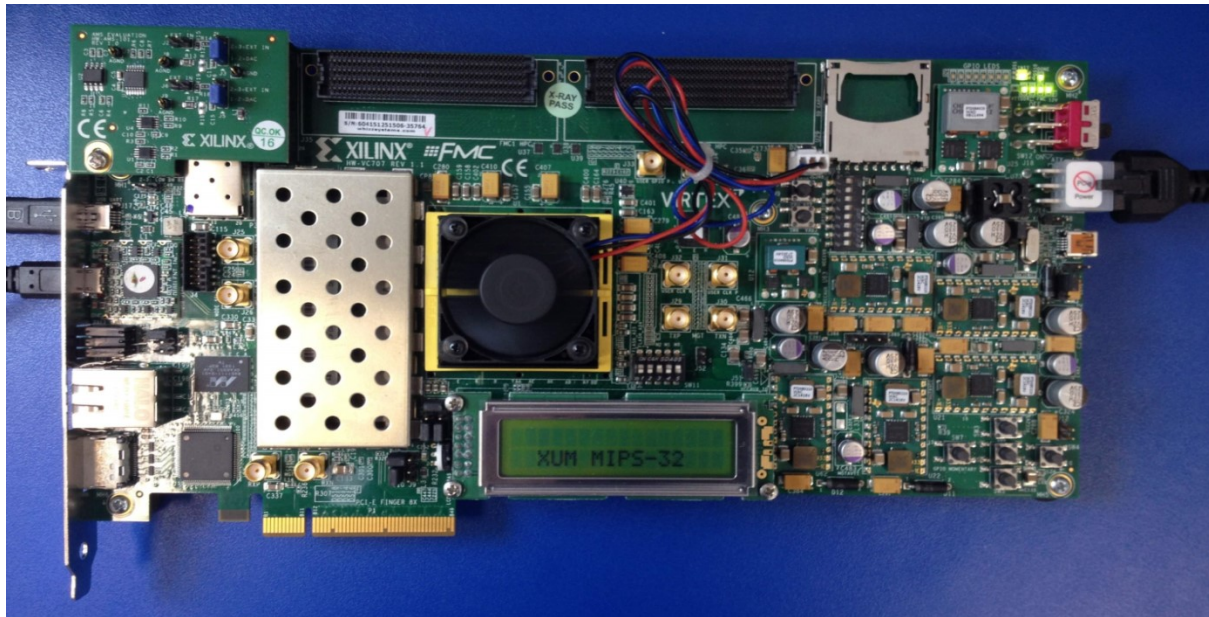
În cadrul acestui referat sunt prezentate următoarele aspecte:

- ✓ rezultatele obținute în urma sintetizării procesorului nMPRA și a planificatorului nHSE
- ✓ comparații cu alte implementări de procesor descrise în literatura de specialitate

În urma testelor efectuate putem afirma că rezultatele experimentale demonstrează implementarea în practică a aspectelor teoretice, obținând astfel timpi foarte mici pentru operațiile de schimbare a contextelor.

1. Introducere

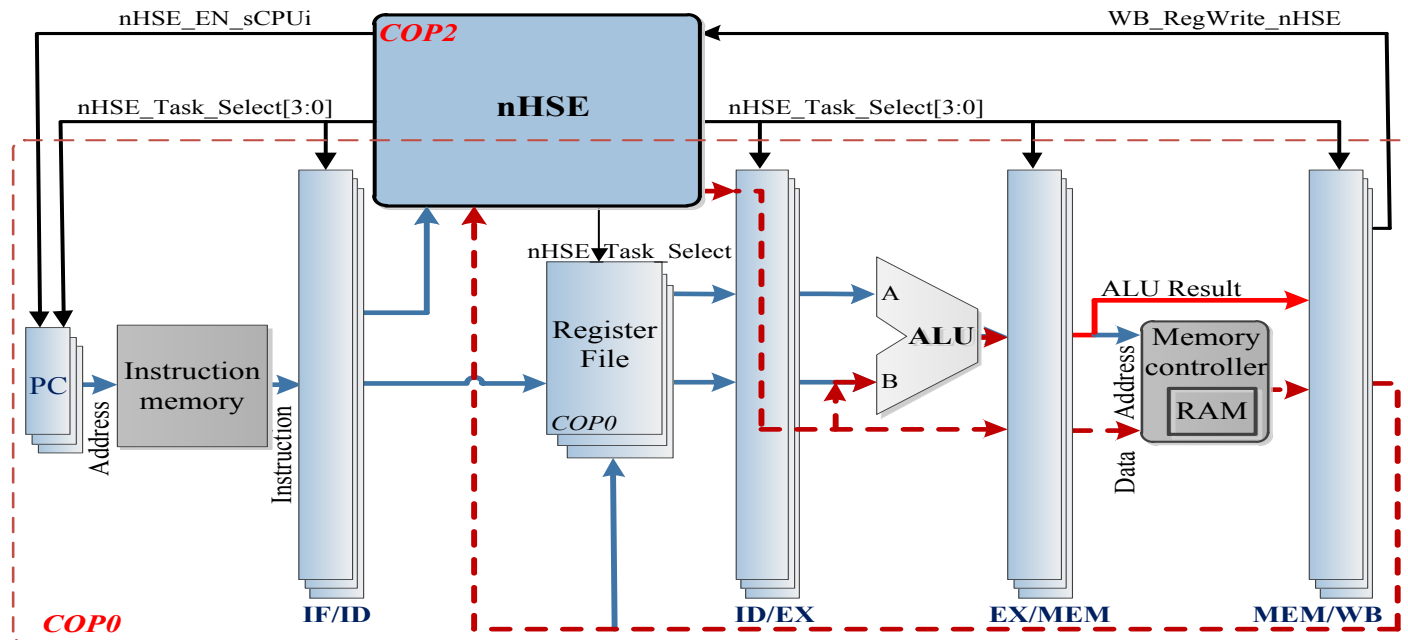
Această implementare de procesor a fost proiectată și realizată utilizând platforma de dezvoltare bazată pe FPGA Virtex-7 și Verilog HDL. Utilizând kit-ul de dezvoltare VC707 produs de Xilinx, prin intermediul unui proiect System-on-Chip, s-a proiectat, sintetizat și simulat arhitectura de procesor nMPRA ce înglobează planificatorul hardware integrat nHSE.



Kit-ul de dezvoltare cu FPGA Virtex-7 (MANSiD)

2. IMPLEMENTAREA PROCESORULUI nMPRA ȘI A PLANIFICATORULUI DINAMIC NHSE

Procesorul nMPRA pe 32 de biți este în conformitate cu **MIPS32 Release 1 ISA**. MIPS pune la dispoziția utilizatorului un sistem de coprocesoare pentru extensia de funcționalități ale procesorului de bază. Figura următoare ilustrează efectul execuției instrucțiunilor **LWC2** și **SWC2**, acestea făcând parte din categoria instrucțiunilor de lucru cu memoria de date.



Arhitectura nMPRA și efectul execuției instrucțiunilor LWC2 și SWC2

2. IMPLEMENTAREA INSTRUCȚIUNILOR DEDICATE PLANIFICATORULUI DINAMIC NHSE

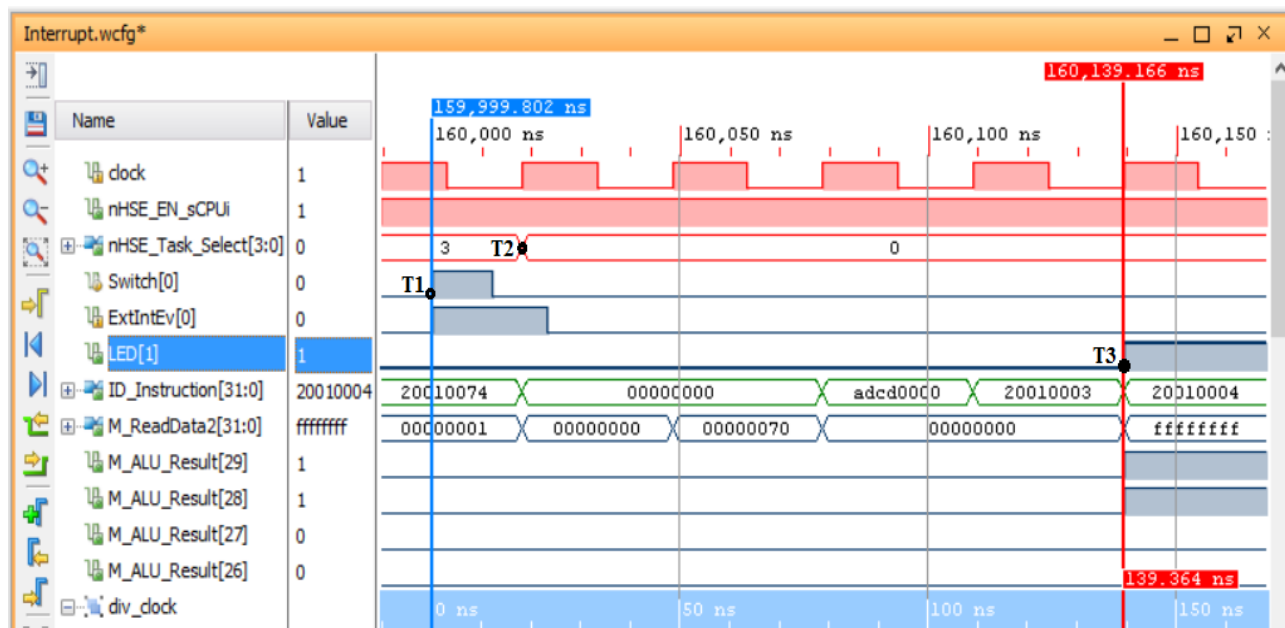
Arhitectura MIPS alocă coprocesorul 2 (COP2) pentru implementările specifice utilizatorului. În Tabelul 2.1 (pag. 11, referat3) sunt descriși regiștrii **locali** (lr), **globali** (gr), de **control** (cr) și de **monitorizare** (mr). Prezența noilor regiștri dedicați planificatorului nHSE necesită și extensia setului de instrucțiuni.

Mnemonică	Ce execută instrucțiunea	OpCode	Rs	Operația	Regiștrii implicați
CFC2	Transferă un cuvânt de control de la COP2	010010	CF 00010	$GPRi[Rt] \leftarrow COP2[Immediate]$	<u>crTRi</u> , <u>crEVi</u> , <u>crEPRi</u> , <u>cr0D1</u> , <u>cr0D2</u> , <u>cr0MSTOP</u> , <u>cr0RESET</u> , <u>cr0CPUID</u> , <u>crEMRij</u> , <u>crEERij</u>
CTC2	Transferă un cuvânt de control la COP2	010010	CT 00110	$COP2[Immediate] \leftarrow GPRi[Rt]$	<u>crTRi</u> , <u>crEVi</u> , <u>crEPRi</u> , <u>cr0D1</u> , <u>cr0D2</u> , <u>cr0MSTOP</u> , <u>cr0RESET</u> , <u>crEMRij</u> , <u>crEERij</u>
MFC2	Transferă un cuvânt de la COP2	010010	MF 00000	$GPRi[Rt] \leftarrow COP2[Immediate]$	<u>mrPRIsCPUi</u> , <u>mrTEvi</u> , <u>mrWDEVi</u> , <u>mrD1EVi</u> , <u>mrD2Evi</u> , <u>mrCntRuni</u> , <u>mrCntSleepi</u> , <u>mr0CntSleep</u> , <u>mrCommRegij</u>
MTC2	Transferă un cuvânt la COP2	010010	MT 00001	$COP2[Immediate] \leftarrow GPRi[Rt]$	<u>mrTEvi</u> , <u>mrWDEVi</u> , <u>mrD1EVi</u> , <u>mrD2Evi</u> , <u>mrCntRuni</u> , <u>mrCntSleepi</u> , <u>mr0CntSleep</u> , <u>mrCommRegij</u>
LWC2	Încarcă un cuvânt dublu în COP2	110010	rs	$COP2[Rt,0] \leftarrow MEM[GPRi[Rs]+Immediate]$	<u>grINT Idi</u> , <u>grMutexi</u> , <u>grERFi</u> , <u>grEViG</u> , <u>grINT PR</u>
SWC2	Memorează un cuvânt de la COP2	110011	rs	$MEM[GPRi[Rs]+Immediate] \leftarrow COP2[Rt,0]$	<u>grINT Idi</u> , <u>grMutexi</u> , <u>grERFi</u> , <u>grEViG</u> , <u>grINT PR</u>

Instrucțiunile dedicate planificatorului nHSE

2. IMPLEMENTAREA PLANIFICATORULUI DINAMIC NHSE - TESTAREA JITTER-ULUI

Prin intermediul formelor de undă din figura următoare, obținute prin intermediul simulatorului Vivado, se va măsura jitter-ul procesorului nMPRA la apariția unui eveniment extern asincron, cât și timpul necesar pentru schimbarea contextelor. Aceste forme de undă reprezintă semnalele proiectului System-on-Chip cât și a procesorului nMPRA.



Timpul de răspuns la apariția unei întreruperi externe

3. IMPLEMENTAREA MECANISMELOR DE SINCRONIZARE ȘI COMUNICAȚIE INTER-TASK

Mecanismele de sincronizare și comunicație inter-task sunt alte două aspecte importante specifice SOTR. Implementarea acestor mecanisme în hardware îmbunătățește în mod convenabil coeficientul WCET, furnizând astfel o soluție optimă pentru comunicația între task-uri și **excluderea mutuală în cazul resurselor partajate**.

După cum se poate observa în tabelul următor, fiecare registru **grMutexi** are un bit pentru a memora starea mutex-ului și 31 de biți pentru a memora ID-ul semiprocessorului proprietar.

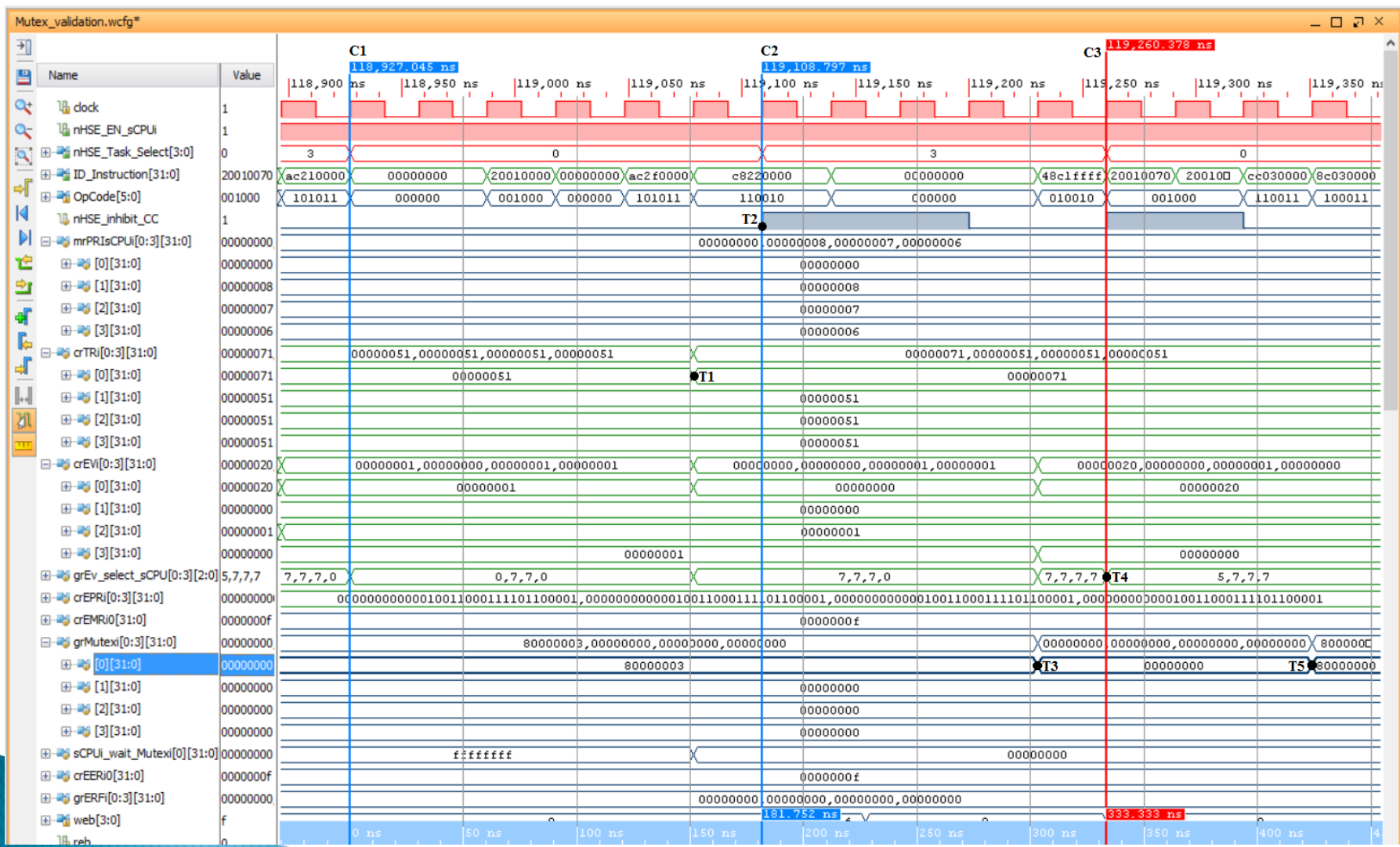
grMutexi	31	30..5	4	3	2	1	0
	Mutex _i		TaskID _{bit4}	TaskID _{bit3}	TaskID _{bit2}	TaskID _{bit1}	TaskID _{bit0}
grMutex ₀	0/1(Mutex ₀)						
grMutex ₁	0/1(Mutex ₁)						
...
grMutex _{m-1}	0/1(Mutex _{m-1})		TaskID _{bit4}	TaskID _{bit3}	TaskID _{bit2}	TaskID _{bit1}	TaskID _{bit0}

Implementarea Fișierului de Registri pentru Mutex-uri – FRM

Adresa	Registru	2nj+k+1	nj-1	...	0	nj-1	...	0	k-1	...	1	0
		Ev _i	sID _{nj-1}	...	sID ₀	dID _{nj-1}	...	dID ₀	Mess _{k-1}	...	Mess ₁	Mess ₀
Adresa ₀	grERF ₀	0/1										
Adresa ₁	grERF ₁	0/1										
...
Adresa _{e-1}	grERF _{e-1}	0/1										

Implementarea Fișierului de Registri pentru Evenimente

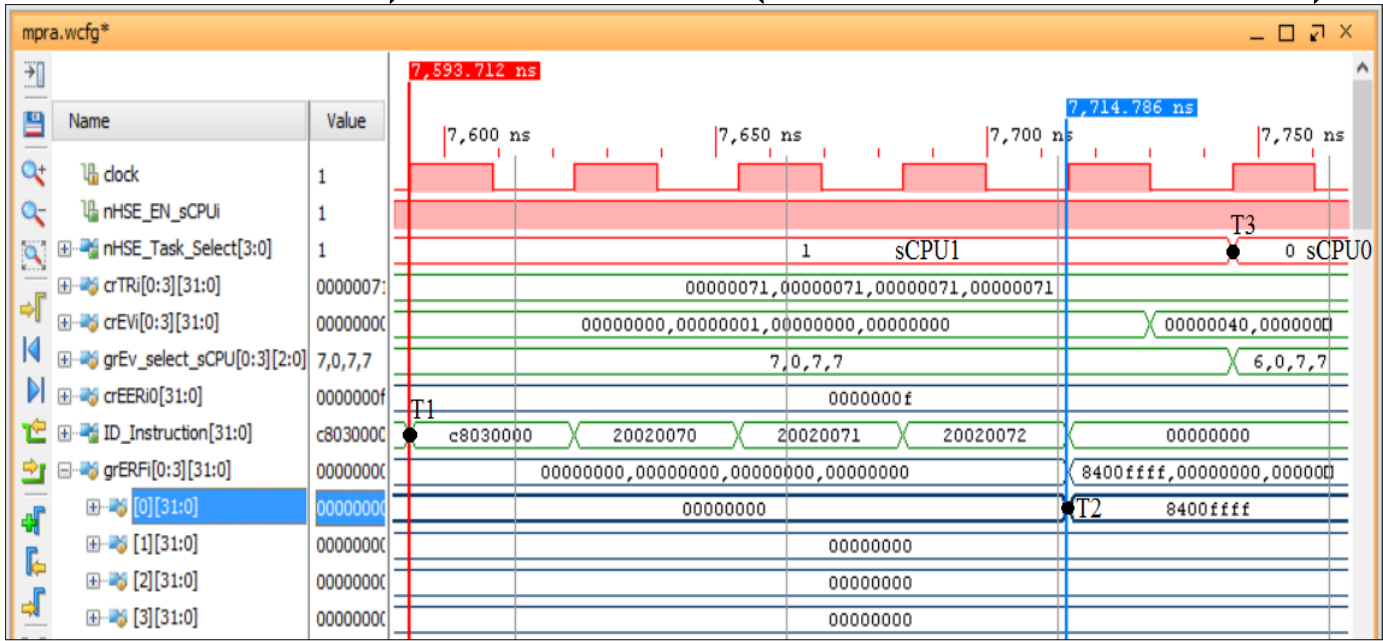
3.1. IMPLEMENTAREA MECANISMULUI DE SINCRONIZARE PRIN MUTEX-URI



Semnălele corespunzătoare mutex-urilor obținute prin intermediul simulatorului Vivado

3.2. IMPLEMENTAREA MECANISMULUI DE COMUNICAȚIE INTER-TASK

Pentru implementarea mecanismului de comunicație inter-task, după cum se poate vedea în tabelul următor, nMPRA folosește un număr de e regiștri globali grERFi fiecare având $2*n+k+1$ biți. Cuvântul $0x8400ffff$ (32'b **1_00001_00000_000001111111111111111111**) reprezintă operația de trimitere a mesajului **_00000111111111111111111111** de la sCPU1 (ID_sursa=00001) către sCPU0 (ID_destinatie=00000).

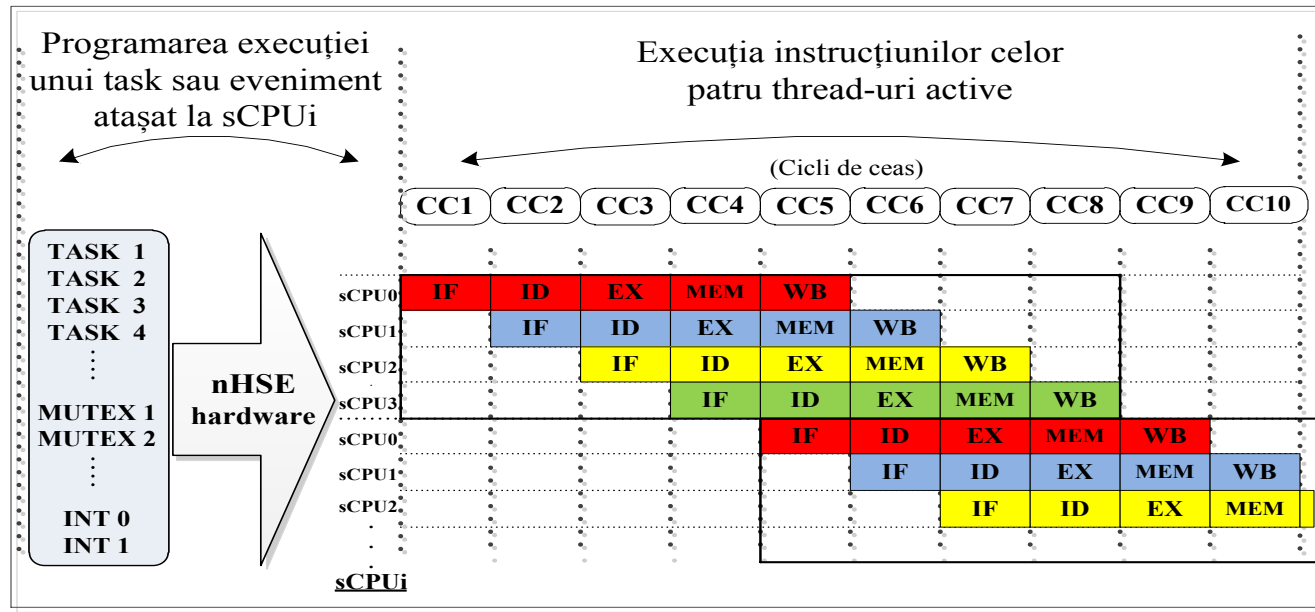


Semnalele corespunzătoare mecanismului de comunicare inter-task

4. PLANIFICAREA TASK-URILOR CU ARHITECTURA DE TIMP REAL NMPRA-MT

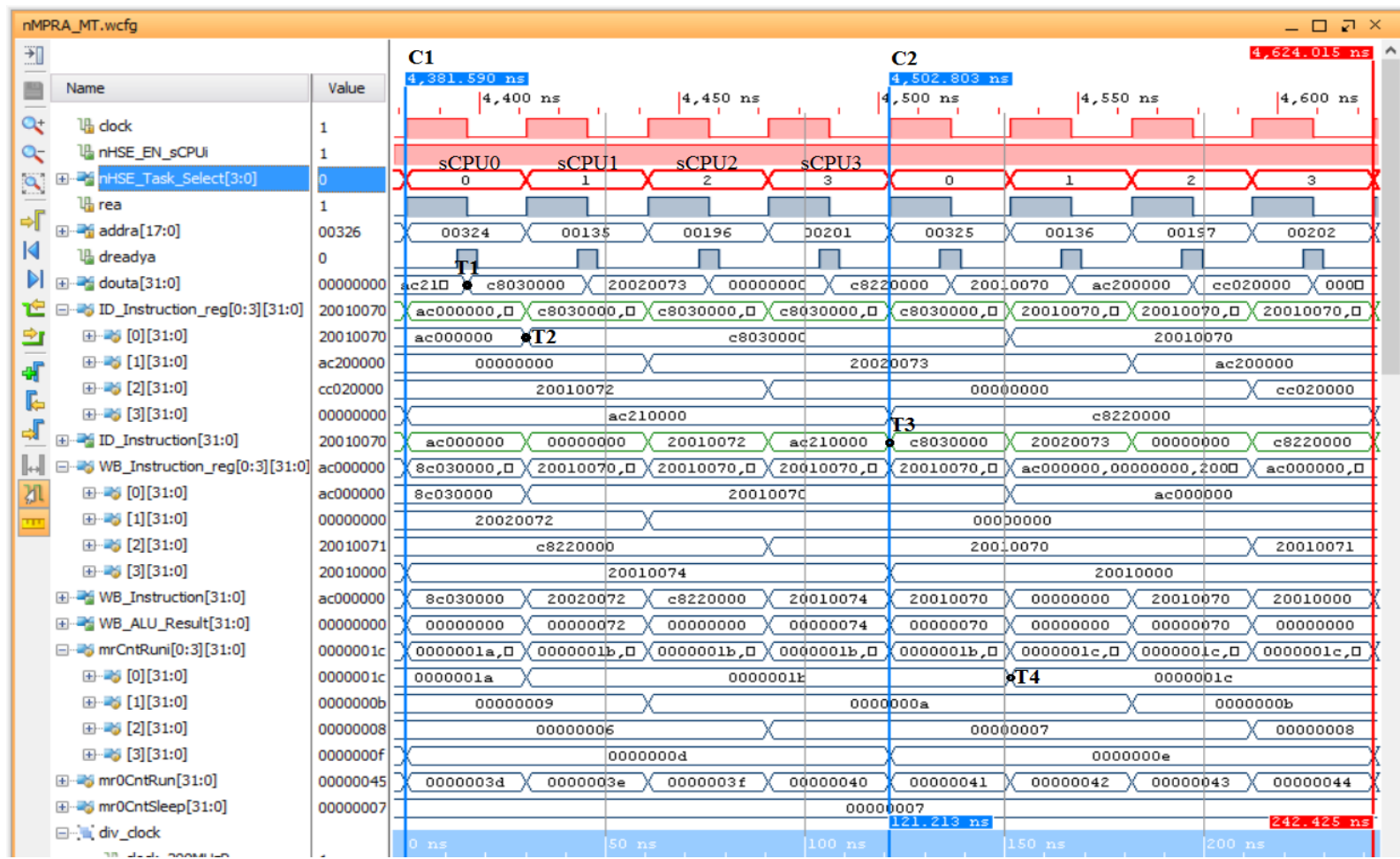
Arhitectura **nMPRA-MT** (Multi Pipeline Register Architecture - Multithreading) este o implementare multithreading a procesorului nMPRA.

După cum se poate observa în figura următoare, procesorul nMPRA-MT nu mai poate intra într-o situație de **hazard de date** datorită datelor consumate de o instrucțiune ce nu au fost încă produse de instrucțiunea precedentă.



Planificatorul nHSE și arhitectura nMPRA-MT

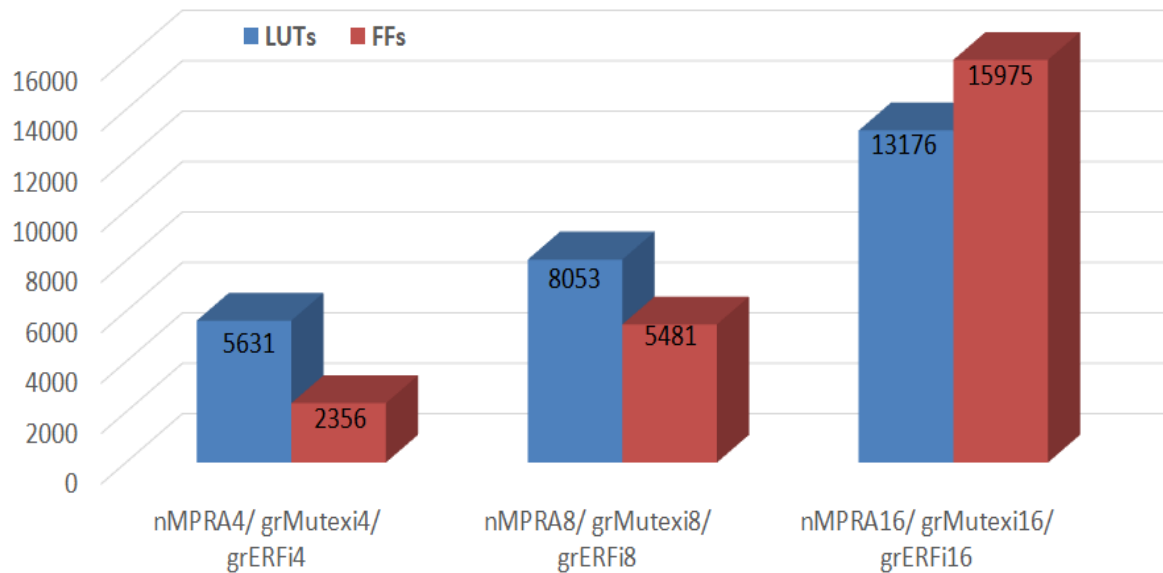
4. PLANIFICAREA TASK-URILOR CU ARHITECTURA DE TIMP REAL NMPRA-MT



Formele de undă corespunzătoare procesorului nMPRA-MT

5. Costuri de implementare ale arhitecturii nMPRA

Deși arhitectura MPRA este o arhitectură cu multiplexare de resurse, **costurile de implementare ale acesteia sunt avantajoase** față de alte arhitecturi comerciale. Implementarea acestui procesor pentru un număr mare de task-uri ar atrage după sine un necesar de resurse prea mare și timpi de propagare ale semnalelor nejustificați de mari.



Resursele necesare pentru implementarea arhitecturii nMPRA cu 4, 8 și 16 sCPUi

5.1. Necesarul de resurse pentru implementarea arhitecturii nMPRA și a proiectului SoC

În tabelul următor este prezentat necesarul de resurse pentru implementarea întregului proiect System-on-Chip ce înglobează procesorul nMPRA. Aici sunt incluse și resursele utilizate pentru implementarea on-chip a **memoriei dual-port** pentru instrucțiuni și date, a mecanismului de **supra-eșantionare a datelor** recepționate prin UART și **modulul Bootloader** pentru reîncărcarea instrucțiunilor în memoria program.

Resurse utilizate de proiectul SoC	LUT	LUTRAM	FF	BRAM	IO	BUFG	MMCM
nMPRA4/ grMutexi4/ grERFi4	7170	898	3022	148	28	16	1
nMPRA8/ grMutexi8/ grERFi8	10436	1074	6151	148	28	16	1
nMPRA16/ grMutexi16/ grERFi16	15855	1402	4333	148	28	16	1
Resurse disponibile (Virtex-7)	303600	130800	607200	1030	700	32	14

Resursele necesare pentru implementarea proiectului SoC ce înglobează procesorul nMPRA

5.1. Necesarul de resurse pentru implementarea arhitecturii nMPRA și a proiectului SoC

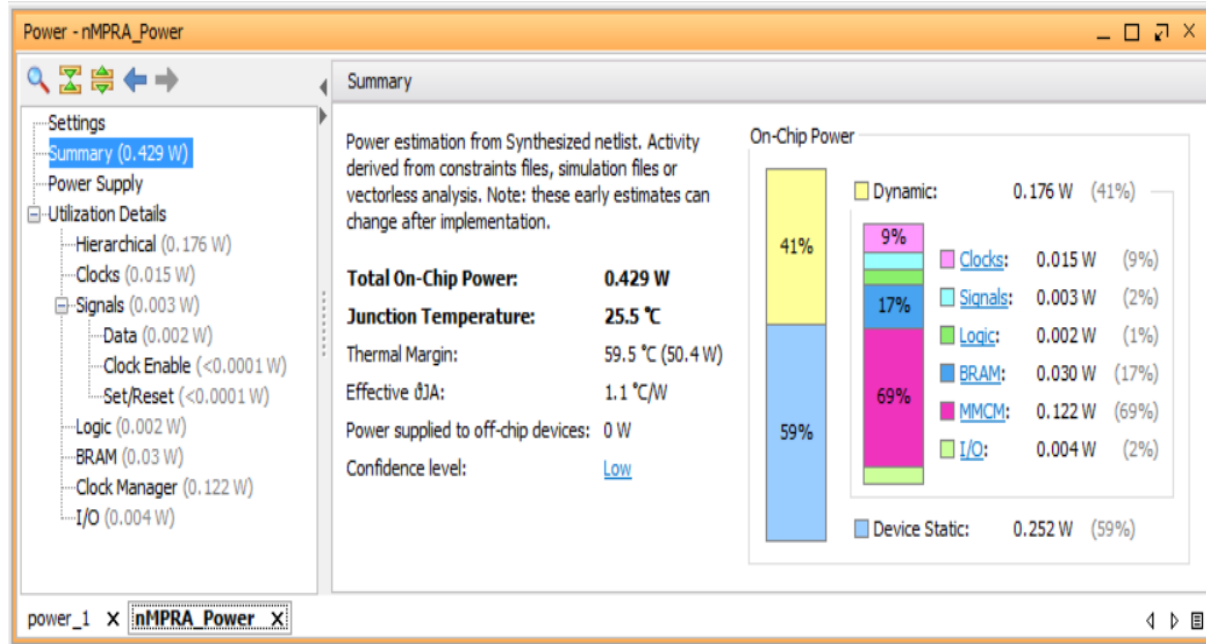
În cazul regiștrilor pipeline, registrul care consumă cea mai multă memorie este registrul pipeline ID/EX, adică 1748 LUTs.

Name	Slice LUTs (303600)	Slice Registers (607200)	F7 Muxes (151800)	F8 Muxes (75900)	Block RAM Tile (1030)	Bonded IOB (700)	BUFCTRL (32)	MMCME2_ADV (14)
Top	7170	3022	308	136	148	28	16	1
UART (uart_bootloader)	480	264	1	0	0	0	0	0
Switches (Switches)	0	1	0	0	0	0	0	0
Memory (BRAM_592KB_Wr...)	895	102	256	128	148	0	0	0
MIPS32 (Processor)	5631	2356	24	0	0	0	0	0
nHSE_inst (nHSE)	872	1879	0	0	0	0	0	0
WBMemtoReg_Mux (Mu...)	34	0	0	0	0	0	0	0
RegisterFile (RegisterFile)	242	0	0	0	0	0	0	0
PC_Add4 (Add_7)	29	0	0	0	0	0	0	0
PCSrcStd_Mux (Mux4_6)	32	0	0	0	0	0	0	0
PC (Register)	128	0	0	0	0	0	0	0
MWriteData_Mux (Mux2)	24	0	0	0	0	0	0	0
MEMWB (MEMWB_Stage)	259	0	0	0	0	0	0	0
IFID (IFID_Stage)	812	0	8	0	0	0	0	0
IDRtFwd_Mux (Mux4_5)	32	0	0	0	0	0	0	0
IDRsFwd_Mux (Mux4_4)	34	0	0	0	0	0	0	0
IDEX (IDEX_Stage)	1748	0	15	0	0	0	0	0
EXRtRdLnk_Mux (Mux4_...	5	0	0	0	0	0	0	0
EXRtFwdLnk_Mux (Mux4_...	32	0	0	0	0	0	0	0
EXRsFwd_Mux (Mux4_2)	48	0	0	0	0	0	0	0
EXMEM (EXMEM_Stage)	643	0	0	0	0	0	0	0
EXALUImm_Mux (Mux4)	38	0	0	0	0	0	0	0
DataMem_Controller (M...	6	32	0	0	0	0	0	0
Controller (Control)	55	132	1	0	0	0	0	0
CP0 (CPZero)	116	144	0	0	0	0	0	0
BranchAddress_Add (Add)	30	0	0	0	0	0	0	0
ALU (ALU)	406	168	0	0	0	0	0	0
LEDs (LED)	0	9	0	0	0	0	0	0
LCD_Screen (LCD)	160	289	27	8	0	0	0	0
Clock_Generator (PLL_200...	0	0	0	0	0	0	4	1

Resursele necesare implementării proiectului SoC ce include procesorul nMPRA cu 4 sCPUi

5.2. Analiza puterii consumate

Figura următoare prezintă în detaliu puterea consumată de implementarea nMPRA cu 4 sCPUi, având la bază suportul hardware pentru tratarea întreruperilor și implementarea mecanismelor de sincronizare și comunicație inter-task în hardware. Valorile obținute pot varia în funcție de versiunea de nHSE și platforma utilizată.



Puterea consumată de implementarea nMPRA cu 4 sCPUi

6. Comparație cu planificatoarele software și alte planificatoare hardware existente

	Tipul arhitecturii	Implementare planificator	Frecvența	Platforma utilizată	Pipeline/ Implementare
nMPRA4 [6]	MIPS32	Static și dinamic în HW	33MHz	Xilinx Virtex-7 XC7VX485T-2FFG1761C	5 etaje pipeline/ Single-core
FlexPret [20]	RISC-V	Static și dinamic (EDF, rate-monotonic)/ SW	80MHz	Xilinx Virtex-5 XC5VLX110T	5 etaje pipeline fine-grained multithreaded/ Single-core
Merasa [21]	MERASA bazat pe SMT cores	Round robin implementat în HW/SW	25 MHz (real-time bus)	Altera Stratix II EP2S180F1020C3	Două benzi de asamblare cu 5 etaje pipeline/ Multi-core
MicroBlaze [22]	MicroBlaze Architecture (RISC)	SW	Max 396Mhz (Virtex-7)	Xilinx Virtex-7 XC7VX485T-ffg1761-3	3 etaje pipeline/ Single-core
ARPA-MT [26]	Implementare SMT bazată pe MIPS32	HW	42.8MHz (4 contexts)	Spartan-3 FPGA	5 etaje pipeline/ Single-core
Amber 23 [23], [24]	ARM-compatible RISC 32 biti	SW	40MHz - Spartan-6 /80MHz - Virtex-6	Xilinx Spartan-6 SP605 xc6slx45t- fgg484-3/ Virtex-6	Amber 23 - 3 etaje pipeline/Single-core, Amber 25 - 5 etaje pipeline/Single-core

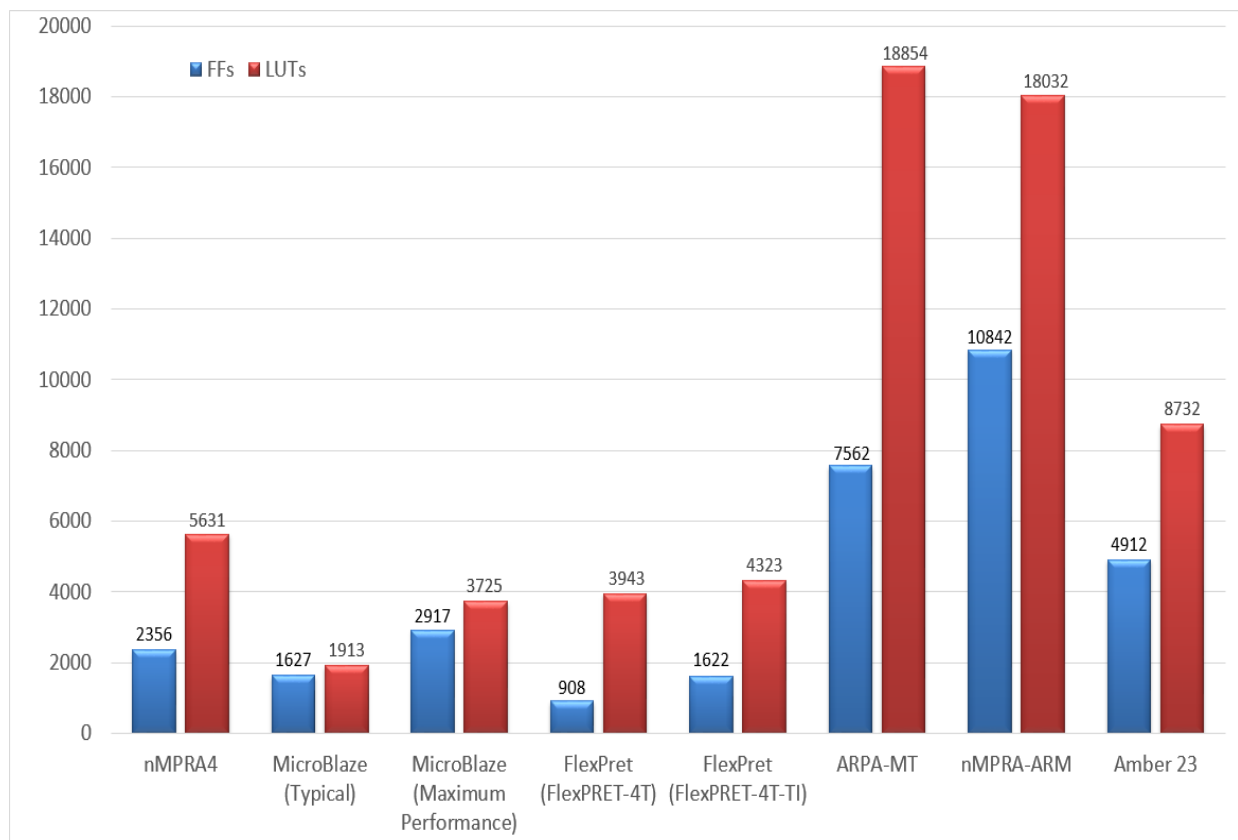
Comparație cu alte implementări de procesor

6. Comparație cu planificatoarele software și alte planificatoare hardware existente

Implementare/Resurse utilizate/Platforma	FFs	LUTs	Platforma utilizată
<u>nMPRA4</u> [6] (4 contexte, 33MHz)	2356	5631	<u>Xilinx</u> Virtex-7 XC7VX485T-2FFG1761C
<u>MicroBlaze</u> [22] (Typical, 255MHz)	1627	1913	<u>Xilinx</u> Virtex-7 XC7VX485T-ffg1761-3
<u>MicroBlaze</u> [22] (Maximum Performance, 231MHz)	2917	3725	<u>Xilinx</u> Virtex-7 XC7VX485T-ffg1761-3
<u>FlexPret</u> [20] (FlexPRET-4T, 80MHz)	908	3943	<u>Xilinx</u> Virtex-5 XC5VLX110T
<u>FlexPret</u> [20] (FlexPRET-4T-TI, 80MHz)	1622	4323	<u>Xilinx</u> Virtex-5 XC5VLX110T
<u>ARPA-MT</u> [26] (4 contexte, 42.8MHz)	7562	18854	Spartan-3 FPGA XC3S1500
<u>nMPRA-ARM</u> [25] (4 fire de execuție)	10842	18032	<u>Xilinx</u> Spartan-6 SP605 xc6slx45t-fgg484-3
<u>Amber 23</u> [23],[24] (40MHz)	4912	8732	<u>Xilinx</u> Spartan-6 SP605 xc6slx45t-fgg484-3

Resursele utilizate pentru implementarea diferitelor arhitecturi de procesor

6. Comparație cu planificatoarele software și alte planificatoare hardware existente



Resursele utilizate pentru implementarea proiectului nMPRA4 cât și a altor procesoare descrise în literatura de specialitate

8. CONCLUZII

Abordarea acestui subiect din punct de vedere științific, tehnologic și economic se datorează **evoluțiilor spectaculoase** din domeniul sistemelor înglobate și aria de aplicabilitate care poate fi **automotive, avionică** sau **robotică**.

Elementele de dificultate ale proiectului de cercetare sunt date de obținerea unei **arhitecturi predictibile și performante**, prin înglobarea funcțiilor specifice SOTR în hardware și evidențierea îmbunătățirii performanțelor prin programe de test adecvate.

Procesoarele pipeline sunt dificil de proiectat datorită dimensiunilor foarte mari și multitudinea de fire de legătură și module diverse conectate împreună. De asemenea a fost foarte important să se decidă care regiștri sa fie de tip '**registered**' și care componente sa fie conectate la **semnalul de ceas** (clocked).

9. PUBLICAȚII

1. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. "CPU architecture description based on fine-grained multithreading and hardware scheduler engine". *Sisteme Distribuite*, Vol: XII, Suceava, Romania, **2014**, ISSN/ISBN: 1842-6808.
2. GAITAN, Nicoleta Cristina; **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. "Predictable CPU Architecture Designed for Small Real-Time Application – Concept and Theory of Operation". In: International Journal of Advanced Computer Science and Applications IJACSA, 6(4), **2015**, pp. 47-52.
3. GAITAN, Nicoleta Cristina; **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. "IMPROVING THE PREDICTABILITY OF NMPRA AND NHSE ARCHITECTURE". In: *Bulletin of the Polytechnic Institute of Iasi, Automatic Control and Computer Science Section*, fasc. 1/**2015**, pp. 27-38.
4. GAITAN, Nicoleta Cristina; GAITAN, Vasile Gheorghita; UNGUREAN, Ioan; **ZAGAN, Ionel**. "Methods to improve the performances of the real-time operating systems for small microcontrollers". In: *20th International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, Romania, 27-29 May **2015**, pp. 261-266.
5. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. "Predictable CPU Architecture Designed for Small Real-Time Applications – Implementation Results". In: *3rd International Conference on Advances in Computing, Electronics and Communication (ACEC)*, 10 - 11 October **2015** / Zurich, Switzerland. ISBN: 978-1-63248-064-4, pp. 143-150.
6. **ZAGAN, Ionel**. "Improving the performance of CPU architectures by reducing the Operating System overhead". In: *The 3rd IEEE Workshop on Advances in Information, Electronic and Electrical Engineering AIEEE'2015*, Riga, Latvia, pp.1-6, 13-14 Nov. **2015**.
7. **ZAGAN, Ionel**. "Real-time evaluation of nMPRA CPU Architecture based on Multithreaded Execution", *8th International Conference on Computer Science and Information Technology*, 10 - 11 December **2015**, Amsterdam, Netherlands.
8. **ZAGAN, Ionel**; GAITAN, Nicoleta Cristina; GAITAN, Vasile Gheorghita. "Scheduling real-time tasks with nMPRA architecture for embedded applications". *Sisteme Distribuite*, Vol: XIII, 16 December **2015**, Suceava, Romania, ISSN/ISBN: 1842-6808.

9. PUBLICAȚII

9. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. “Schedulability Analysis of nMPRA Processor based on Multithreaded Execution”, 13rt International Conference on DEVELOPMENT AND APPLICATION SYSTEMS (DAS), Suceava, Romania, May 19–21, **2016**.
10. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. “Improving the performance of CPU architectures by reducing the Operating System overhead (Extended Version)”, In: The Scientific Journal of Riga Technical University – Electrical, Control and Communication Engineering, **2016**.
11. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. “Improving the Performances of the nMPRA Processor using a Custom Interrupt Management Scheduling Policy”, In: Advances in Electrical and Computer Engineering (AECE), Volume 16, Issue 4, 30/11/2016, pp. 45–50.
12. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita. “Improving the Performances of the nMPRA Architecture by Implementing Specific Functions in Hardware”, accepted for presentation: 19th International Conference on Digital Circuits and Microarchitecture Technologies (ICDCMT 2017), Berlin, Germany, May 21–22, **2017**.
13. **ZAGAN, Ionel**; GAITAN, Vasile Gheorghita, “CPU Architecture Based on Static Hardware Scheduler Engine and Multiple Pipeline Registers”, Accepted for publication: 19th International Conference on Advanced Computing Systems and Microarchitecture (ICACSM), Zurich, Switzerland, September 15 – 16, **2017**.
14. **ZAGAN, Ionel**; GAITAN, Nicoleta Cristina; GAITAN, Vasile Gheorghita. “An Approach of nMPRA Architecture using Hardware Implemented Support for Event Prioritization and Treating”, Accepted for publication: International Journal of Advanced Computer Science and Applications (IJACSA), 2016.



Universitatea “Ștefan cel Mare” Suceava

Facultatea de Inginerie Electrică și Știința Calculatoarelor



Craciun Fericit!

