

## Anexa 12. R14 – F11

## Informații post

<b>Universitatea</b>	UNIVERSITATEA "ȘTEFAN CEL MARE" DIN SUCEAVA
<b>Facultatea*</b>	INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR
<b>Departament*</b>	CALCULATOARE
<b>Poziția în statul de funcții*</b>	28
<b>Funcție*</b>	Asistent
<b>Disciplinele din planul de învățământ*</b>	Microcontrolere, Arhitecturi paralele, Programarea calculatoarelor și limbaje de Programare III-2, Programarea calculatoarelor și limbaje de programare II, Programarea calculatoarelor și limbaje de programare II, Programarea în limbaj de asamblare, Programarea calculatoarelor și limbaje de programare II, Programarea calculatoarelor și limbaje de programare II
<b>Domeniu științific*</b>	Calculatoare și tehnologia informației
<b>Descriere post*</b>	<p>Postul conține o normă universitară de 40 ore / săptămână cu o normă didactică de 13 ore convenționale, din care: 13 ore de lucrări practice de laborator și proiect cu următoarea distribuție semestrială pe discipline:</p> <ul style="list-style-type: none"> <li>• disciplina: <b>Microcontrolere</b>, efectuată în semestrul 1 cu studenții din anul 2 de la programul de studii <b>Calculatoare</b>: - 8 ore laborator semestrul 1, cu 4 semigrupe;</li> <li>• disciplina: <b>Arhitecturi Paralele</b>, efectuată în semestrul 2 cu studenții din anul 4 de la programul de studii <b>Calculatoare</b>; - 3 ore laborator semestrul 2, cu 2 semigrupe;</li> <li>• disciplina: <b>Programarea Calculatoarelor și Limbaje de Programare III-2</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Calculatoare</b>: - 4 ore laborator semestrul 2, cu 2 semigrupe;</li> <li>• disciplina: <b>Programarea calculatoarelor și limbaje de programare II</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Electronică Aplicată</b> - 2 ore laborator semestrul 2, cu o semigrupa;</li> <li>• disciplina: <b>Programarea calculatoarelor și limbaje de programare II</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Electronică Aplicată</b> - o ora proiect semestrul 2, cu o semigrupa;</li> <li>• disciplina: <b>Programare în Limbaj de Asamblare</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Automatica și Informatica Aplicată</b>: - 2 ore laborator semestrul 2, cu o semigrupa;</li> <li>• disciplina: <b>Programarea calculatoarelor și limbaje de programare II</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Electronică Aplicată</b> - 4 ore laborator semestrul 2, cu 2 semigrupe;</li> <li>• disciplina: <b>Programarea calculatoarelor și limbaje de programare II</b>, efectuată în semestrul 2 cu studenții din anul 2 de la programul de studii <b>Electronică Aplicată</b> - 2 ore proiect semestrul 2, cu 2 semigrupe;</li> </ul> <p>Activitatea se va desfășura în Universitatea "Ștefan cel Mare" din Suceava și în posibilele extensii ale acesteia.</p>
	Activități de predare Activități de seminar / proiecte de an / lucrări de laborator / lucrări practice Activități de evaluare Alte activități: Consultații Participare în comisii de admitere Coordonare cercuri științifice studențești Activități de pregătire științifică și metodică și alte activități în interesul învățământului Activități de cercetare științifică, de dezvoltare tehnologică, activități de proiectare, de creație potrivit specificului Activități administrative
<b>Salariul minim de încadrare</b>	minim ..... lei – maxim ..... lei

	(Salariul va fi stabilit în limitele minime – maxime în funcție de vechimea și performanța candidatului)	
<b>Calendarul concursului</b>		
Data publicării anunțului în Monitorul Oficial	<b>Monitorul Oficial Nr. 569/28.04.2017</b>	
Perioadă înscriere	Început	Sfârșit
Data susținerii prelegerii		
Ora susținerii prelegerii		
Locul susținerii prelegerii		
Perioadă susținere a examenelor	Început	Sfârșit
Perioadă comunicare a rezultatelor	Început	Sfârșit
Perioadă de contestații	Început	Sfârșit
<b>Tematica probelor de concurs*</b>	<p><b>Microcontrolere</b> – Laborator: 1. Elemente de tehnica securității muncii în laborator și organizarea activităților. 2. Prezentarea mediului de dezvoltare ARM Developments Tools și a kit-urilor MCBSTR7 și MCBSTR9. 3. Utilizarea, programarea și testarea funcționării porturilor de intrare/ ieșire. 4. Utilizarea, programarea și testarea funcționării întreruperilor. 5. Utilizarea, programarea și testarea funcționării întreruperilor externe. 6. Utilizarea, programarea și testarea funcționării timer-elor. 7. Utilizarea, programarea și testarea funcționării porturilor seriale. 8. Utilizarea, programarea și testarea funcționării convertoarelor A/D, D/A și a PWM-ului. 9. Utilizarea, programarea și testarea funcționării controlerului CAN. 10. Utilizarea, programarea și testarea funcționării controlerului USB. 11. Programarea PLC-urilor. Aplicația I – monitorizarea unei mărimi analogice. Programarea PLC-urilor . 12. Aplicația II – Implementarea unui segment de rețea industrială locală cu PLC-ul MASTER CANOPEN și Module CANOpen SLAVE, utilizând server și clienți OPC. 13. Recuperări. Bibliografie selectiva: 1. Sloss, et. Al. - "ARM System Developer's Guide", 2004, ISBN 1-55860-874-5. 2. David Seal – "ARM Architecture Reference Guide ", 2003, ISBN 0-201-737191. 3. Steve Furber – "ARM system on chip architecture ", 2000, ISBN 0-201-67519-6. 4. Vasile GAITAN - Studiu privind structura software a aplicațiilor pentru realizarea de sisteme și echipamente pentru conducere și monitorizare Vol I - Considerații generale Pag. 1-87. 2009. Raport de cercetare. 5. Vasile GAITAN - Studiu privind structura software a aplicațiilor pentru realizarea de sisteme și echipamente pentru conducere și monitorizare - Vol III - IEC 61131 – 3. Pag. 263-544.2009. Raport de cercetare. 6. www.arm.com (2014) 7. www.keil.com (2014) 8. www.st.com (2014) 9. <a href="http://www.eed.usv.ro/~cristinag/LabMC/labMC.html">http://www.eed.usv.ro/~cristinag/LabMC/labMC.html</a> (2014)</p> <p><b>Arhitecturi Paralele</b> - Laborator: 1. Elemente de tehnica securității muncii în laborator și organizarea activităților. 2. Implementarea VHDL a unui sistem de calcul MIPS pipeline (calea de date, unitatea de control și unitatea de control ALU). 3. Implementarea VHDL a unui sistem de calcul MIPS pipeline (registrii pipeline, unitatea Branch Control și unitatea Forward Control). 4. Construirea ununi sistem de calcul paralel MIMD folosind o structura FPGA. 5. Introducere în arhitectura CELL. 6. Prezentarea arhitecturii CELL, a mediului de dezvoltare și realizarea unui program multi-threaded pentru CBE. 7. Mecanisme de comunicare PPU – SPU – Mailbox. 8. Mecanisme de comunicare PPU – SPU – DMA. 9. Implementarea de calcul polinomial I. Aliniere vectori pentru utilizare eficienta a DMA. Impartirea problemei pentru 8 SPU-uri. 10. Implementarea de calcul polinomial II. Utilizarea librariilor de SIMD-izare. "Derularea" buclelor (loop unrolling). 11. Ascunderea duratei transferurilor DMA (dubla buffer-are). 12. Recuperări. Bibliografie selectivă: 1. Parralel Algorithms and Architectures - Cosnard M. Trystram D. – International Thomson Computer Press, 1995. 2. Parallel Computer Architecture - Culler D. Singh, Morgan Kaufmann, 1998. 3. Parallel Processing: From Application to Systems - Moldovan D. - Morgan Kaufmann, 1995. 4. Structured Computer Organization - Tanenbaum A. - Prentice-Hall International, 1999. 5. Aplicații Software Distribuite - Dan Cosma, Stejărel Vereș, Adrian Petru Mierluțiu - Editura de Vest, 2003.</p>	

**Programarea Calculatoarelor și Limbaje de Programare III-2** - Laborator: 1. Introducere în programarea sub Assembler. 2. Familiarizare cu mediul de lucru. Dezasamblarea și studiul unor programe de test. 3. Prezentarea depanatorului DOS Debugger. 4. Realizarea primelor programe realizate sub debugger. 5. Prezentarea TASM și realizarea unor programe simple. 6. Utilizarea instrucțiunilor logico-aritmetice. 7. Implementarea structurilor de control din limbaje de nivel înalt (IF-THEN-ELSE). 8. Implementarea structurilor de control din limbaje de nivel înalt ( FOR ). 9. Implementarea structurilor de control din limbaje de nivel înalt ( WHILE, DO ). 10. Construcția și apelul subrutinelor. 11. Utilizarea sistemului de întreruperi. 12. Apelul / utilizarea funcțiilor BIOS și DOS sub sistemul MS-DOS. 13. Utilizarea macroinstrucțiunilor și a asamblării condiționate pentru programarea avansată sub MASM. 14. Recuperări

Bibliografie selectivă: 1. *Practica dezvoltării software în limbaje de asamblare* – Ion Ivan, Paul Pocatilu, Doru Cazan - Editura Economica, 2002. 2. *Totul despre Z80* – M. Patrubany – Editura Tehnică, 1989. 3. *Bazele Microprocesoarelor* – Liviu Kreindler – Matrix Rom București, 1997. 4. *Familia de Microcontrolere MCS51* - Vasile Găitan, 1997. 5. *Microprocesoarele 8086,286,386* - Athanasiu I., Panoiu Al. - Ed. TEORA, 1992. 6. *Procesoare Intel, Programare în Limbaj de Asamblare* - Vasile Lungu, ediția II a, 2004. 7. *Fundamentals of Embedded Software with the ARM Cortex-M3* - Daniel W. Lewis –Ediția a 2-a, 2013.

**Programarea calculatoarelor si limbaje de programare II** - Laborator: 1. Introducere în programarea sub Assembler. 2. Familiarizare cu mediul de lucru. Dezasamblarea și studiul unor programe de test. 3. Prezentarea depanatorului DOS Debugger. 4. Realizarea primelor programe realizate sub debugger. 5. Prezentarea TASM și realizarea unor programe simple. 6. Utilizarea instrucțiunilor logico-aritmetice. 7. Implementarea structurilor de control din limbaje de nivel înalt (IF-THEN-ELSE). 8. Implementarea structurilor de control din limbaje de nivel înalt ( FOR ). 9. Implementarea structurilor de control din limbaje de nivel înalt ( WHILE, DO ). 10. Construcția și apelul subrutinelor. 11. Utilizarea sistemului de întreruperi. 12. Apelul / utilizarea funcțiilor BIOS și DOS sub sistemul MS-DOS. 13. Utilizarea macroinstrucțiunilor și a asamblării condiționate pentru programarea avansată sub MASM. 14. Recuperari

Bibliografie selectiva: 1. *Practica dezvoltării software in limbaje de asamblare* – Ion Ivan, Paul Pocatilu, Doru Cazan - Editura Economica, 2002. 2. *Totul despre Z80* – M. Patrubany – Editura Tehnica, 1989. 3. *Bazele Microprocesoarelor* – Liviu Kreindler – Matrix Rom Bucuresti, 1997. 4. *Familia de Microcontrolere MCS51* - Vasile Gaitan, 1997. 5. *Microprocesoarele 8086,286,386* - Athanasiu I., Panoiu Al. - Ed. - TEORA 1992 6. *Procesoare Intel, Programare în Limbaj de asamblare* - Vasile Lungu - ediția II a, 2004.

**Programarea calculatoarelor si limbaje de programare II** – Proiect: 1. Familiarizare cu mediul de lucru. 2. Prezentarea depanatorului DOS Debugger. 3. Utilizarea instrucțiunilor logico-aritmetice. 4. Implementarea structurilor de control din limbaje de nivel înalt (IF-THEN-ELSE). 5. Implementarea structurilor de control din limbaje de nivel înalt ( FOR ). 6. Implementarea structurilor de control din limbaje de nivel înalt ( WHILE, DO ). Construcția și apelul subrutinelor. 7. Utilizarea sistemului de întreruperi. Apelul / utilizarea funcțiilor BIOS și DOS sub sistemul MS-DOS. Utilizarea macroinstrucțiunilor și a asamblării condiționate pentru programarea avansată sub MASM.

Bibliografie selectiva: 1. *Practica dezvoltării software in limbaje de asamblare* – Ion Ivan, Paul Pocatilu, Doru Cazan - Editura Economica, 2002. 2. *Totul despre Z80* – M. Patrubany – Editura Tehnica, 1989. 3. *Bazele Microprocesoarelor* – Liviu Kreindler – Matrix Rom Bucuresti, 1997. 4. *Familia de Microcontrolere MCS51* - Vasile Gaitan, 1997. 5. *Microprocesoarele 8086,286,386* - Athanasiu I., Panoiu Al. - Ed. - TEORA 1992 6. *Procesoare Intel, Programare în Limbaj de asamblare* - Vasile Lungu - ediția II a, 2004.

**Programare in Limbaj de Asamblare** - Laborator: 1. Introducere în programarea sub Assembler. 2. Familiarizare cu mediul de lucru. Dezasamblarea și studiul unor programe

	<p>de test. 3. Prezentarea depanatorului DOS Debugger. 4. Realizarea primelor programe realizate sub debugger. 5. Prezentarea TASM și realizarea unor programe simple. 6. Utilizarea instrucțiunilor logico-aritmetice. 7. Implementarea structurilor de control din limbaje de nivel înalt (IF-THEN-ELSE). 8. Implementarea structurilor de control din limbaje de nivel înalt ( FOR ). 9. Implementarea structurilor de control din limbaje de nivel înalt ( WHILE, DO ). 10. Construcția și apelul subrutinelor. 11. Utilizarea sistemului de întreruperi. 12. Apelul / utilizarea funcțiilor BIOS și DOS sub sistemul MS-DOS. 13. Utilizarea macroinstrucțiunilor și a asamblării condiționate pentru programarea avansată sub MASM. 14. Recuperari</p> <p>Bibliografie selectiva: 1. <i>Practica dezvoltării software în limbaje de asamblare</i> – Ion Ivan, Paul Pocatilu, Doru Cazan - Editura Economica, 2002. 2. <i>Totul despre Z80</i> – M. Patrubby – Editura Tehnica, 1989. 3. <i>Bazele Microprocesoarelor</i> – Liviu Kreindler – Matrix Rom Bucuresti, 1997. 4. <i>Familia de Microcontrolere MCS51</i> - Vasile Gaitan, 1997. 5. <i>Microprocesoarele 8086,286,386</i> - Athanasiu I., Panoiu Al. - Ed. - TEORA 1992 6. <i>Procesoare Intel, Programare în Limbaj de asamblare</i> - Vasile Lungu - ediția II a, 2004.</p>
--	--