

FIȘA DISCIPLINEI

1. Date despre program

Instituția de învățământ superior	Universitatea <i>Lucian Blaga</i> din Sibiu
Facultatea	Facultatea de Inginerie
Departament	Departamentul de <i>Calculatoare și Inginerie Electrică</i>
Domeniul de studiu	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
Ciclul de studii	Studii de licență
Specializarea	Electronică Aplicată

2. Date despre disciplină

Titlul cursului/ Course title	ARHITECTURA MICROPROCESOARELOR				
Codul cursului/ Course code	Tipul cursului/ Type of course	Nivelul cursului/ Level of course	An de studiu/ Year of study	Semestrul/ Semester	Număr de credite/ Number of credits
39045.503.1218 DO40	Obligativu	Studii de licență	3	1	5
Tipul de evaluare	Categoria formativă a disciplinei (DF=fundamentală.; DD=domeniu; DS=specialitate; DC=complementară)				
EXAMEN	DD				
Titular activități curs/ Course titular	Prof.univ.dr.ing. Adrian FLOREA				
Titular activități seminar/ laborator/ proiect Seminar/lab/proiect titular	Prof.univ.dr.ing. Adrian FLOREA				

Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână				
Curs/ Lecture	Seminar	Laborator/ Lab	Proiect/ Project	Total
2		2	1	5
Extinderea disciplinei în planul de învățământ – Total ore din planul de învățământ				
Curs/ Lecture	Seminar	Laborator/ Lab	Proiect/ Project	Total(<i>NOAD_{sem}</i>)
28		28	14	70
Bugetul de timp pentru studiul individual				
Denumirea activității	Nr. ore	Denumirea activității	Nr. ore	
1. Studierea notițelor de curs	4	5. Pregătirea seminariilor/ laboratoarelor	6	
2. Studiul după suport de curs	6	6. Elaborarea referatelor, temelor individuale	8	
3. Studierea bibliografiei minimale	6	7. Pregătirea pentru evaluările periodice	8	
4. Documentare suplimentară (internet, bibliotecă, etc..)	4	8. Participarea la consultații	13	
Total ore alocate studiului individual <i>NOSI_{sem}</i>			55	
Total ore pe semestru (<i>NOAD_{sem} + NOSI_{sem}</i>)			125	

<p>Obiectivele cursului/ Course Objectives</p>	<p>1. Dobândirea principalelor cunoștințe și înțelegerea conceptelor de bază referitoare la sistemele cu microprocesoare, abordate prin prisma unui sistem unitar interactiv la nivelul hardware-software.</p> <p>2. Cunoașterea principiilor de proiectare, modelare și simulare în vederea optimizării microarhitecturilor de procesare a informației.</p> <p>3. Abordarea microarhitecturilor de calcul atât din punct de vedere formativ cât și informativ bazat pe dezvoltarea de aplicații.</p>		
<p>Discipline precursore obligatorii/ Prerequisites</p>	<p>Computing Hardware (Basic Digital Logic, Computer Arithmetic), C Programming, LINUX, Compilers.</p>		
<p>Conținutul cursului/ Course contents</p>	<p>Curs 1</p>	<p>Introducere în problematica microarhitecturilor de procesare a informației. Evoluția și clasificarea sistemelor de calcul. Conceptul de calculator omniprezent. Structura unui microsistem de calcul. Rolul blocurilor componente și interacțiunea acestora: microprocesor (decodificatoare / multiplexoare / sumatoare / unități aritmetico-logice, regiștrii), memorii, interfețe, periferice, bus-uri de interconectare, semnale de comenzi și stări, întreruperi.</p>	<p>2</p>
	<p>Curs 2</p>	<p>Instrucțiuni, cicli canonici, stări. Probleme ale comunicării microprocesor – memorie. Memorii DRAM, SRAM, EPROM. Moduri de transfer între microprocesor și dispozitivele periferice (interogare / intrerupere / DMA). Rolul stivei în tratarea întreruperilor imbricate.</p>	<p>2</p>
	<p>Curs 3</p>	<p>Modalități de evaluare a sistemelor de calcul. Metrici de evaluare și alegerea metricilor. Rolul performanței. Simulatoare software dedicate microarhitecturilor de calcul. Metodologii de simulare. Platforme de simulare monolitice respectiv modulare (SimpleScalar / UniSim). Clasificare, caracteristici, utilitate (Execution Driven respectiv Trace Driven). Simularea la nivel tranzacțional în sistemele many-core. Benchmark-uri standardizate (SPEC, EEMBC). Instrumente software utile în analiza și proiectarea microarhitecturilor (asambleare, link-editoare, debuggere, cross-compilatoare). Proiectare simulatoare (interfața cu utilizatorul și crearea resurselor; nucleul funcțional al simulatorului; optimizatoare software de cod obiect). Exemple, aplicații.</p>	<p>4</p>

	Curs 4	Proiectarea sistemului ierarhizat de memorii într-o microarhitectura de calcul. Necesitatea ierarhizării sistemelor de memorii. Localități temporale și spațiale: caracteristici intrinseci ale programelor în execuție (<i>regula 90-10</i>). Memoria cache – definiție, metrici folosite în caracterizare (Rhit, Taces, Rata de procesare). Tipuri arhitecturale de memorii cache: definiție, structură, regula de mapare, principii de proiectare / implementare. Funcționare dinamică – algoritmi de înlocuire a blocurilor conflictuale, strategia de scriere. Folosirea instrumentelor software în evaluarea complexității și a consumului de putere în microarhitecturi; evaluare bazată pe necesitatea simulării consumului energetic aferent unui cache (simulatorul CACTI : http://quid.hpl.hp.com:9081/cacti/detailed.y)	2
	Curs 5	Metode de îmbunătățire a performanței cache-urilor. Optimizări software în vederea reducerii ratei de miss în cache. Coerența cache-urilor în sistemele multiprocesor. Evidențierea conceptelor legate de cache-uri – modul de organizare, regulile de mapare, algoritmi de înlocuire a blocurilor conflictuale, strategia de scriere, optimizări software – folosind simulatorul PCSPIM-CACHE . Insistat pe organizare simulator (ferestre Registru / Sesiune / Cache_Instructiuni_Date), vizualizare metrici, limbaj asamblare MIPS. Implementarea a două aplicații: prima (g_cache.s) are un caracter explicativ cu rolul de a evidenția eficiența memoriilor cache din prisma celor două <i>principii de localitate: temporală și spațială</i> . De asemenea, sunt ilustrate interferențele produse de accesul în aceeași zonă a cache-ului de date din partea a două tablouri (aflate evident la adrese diferite în memoria principală). A doua aplicație se referă la implementarea unei metode software de optimizare care să determine creșterea ratei de hit, și anume <i>interschimbarea buclelor</i> .	2
	Curs 6	Testare parțială din materia parcursă. Seminar de aplicații. (I) Interogare/Intrepereri/DMA – probl. 42, 36, 30 cap.11 [3]; Metrici Cache-uri (rata de hit, timp mediu de acces la memorie) – probl. 40, 64 cap.11 [3], probl. cache - <i>Structura sistemelor de calcul</i> [8].	2
	Curs 7	Microcontrollere și aplicații hard-soft. Introducere. Caracteristici arhitecturale specifice. Programarea microcontrollerului: alegerea limbajului; depanarea, testarea și încărcarea codului în memoria EPROM.	2

	Curs 8	Microcontrollerul Philips 80C51. Organizarea memoriei. Schema bloc de principiu a unui microcontroller. Setul de instrucțiuni, moduri de adresare. Structura mecanismelor de întreruperi hardware. Structura interfețelor de I/O. Magistrala de interconectare I2C. Magistrala ACCES.bus. Protocolul CAN (Control Area Network). Placi și softuri de dezvoltare a aplicațiilor.	2
	Curs 9	Arhitectura microprocesoarelor RISC. Arhitectura setului de instrucțiuni (ISA): regiștrii procesorului, modurile de adresare, sintaxa asamblor, utilizarea memoriei și convenții de apel, formatul instrucțiunilor și setul de instrucțiuni. Optimizarea ISA în vederea facilitărilor compilărilor HLL: formate de instrucțiuni de lungime fixă, opcode de lungime fixă, permiterea adnotarilor pentru creșterea performanței de procesare prin interacțiunea software – hardware, determinarea variabilelor de memorie (frecvent folosite) care pot fi stocate în regiștrii. Simbioza procesor – compilator. Generarea codului obiect pentru o arhitectură dată. Implementarea în hardware a funcțiilor din programele de nivel înalt. Stiva de date aferentă funcțiilor. Transferul parametrilor în cazul apelului. Traducerea codului de la nivel <i>high (C)</i> la nivel <i>low (asamblare)</i> . (Cap. 1 [3], Cap. 10 [2])	2

	Curs 10	<p>Principiile arhitecturale ale microprocesoarelor moderne. Caracteristicile de bază ale modelului RISC. Setul optimizat de instrucțiuni: diferențe MIPS – DLX (instrucțiuni pentru rest, instrucțiuni de transfer și de salt condiționat). Structura pipeline a microprocesorului (fazele). Avantajele procesării pipeline. Implementarea non-pipeline respectiv pipeline a arhitecturii „hardware” a microprocesoarelor RISC (probl. 24, 29 (a, b) și 50 cap.11 [3]). Proiectarea caii de date și a caii de control a microprocesorului. Problemele hazardurilor în microprocesoarele RISC: definire, clasificare (structurale, de date, de ramificații), soluții de evitare a efectelor defavorabile. Principii de proiectare a unității de control în vederea detectării / eliminării hazardurilor (<i>forwarding, renaming</i>). Vizualizarea hazardurilor cu ajutorul simulatorului WinDLX. Optimizarea unităților secvențiale de program și predicția ramificațiilor (probl. <i>Branch prediction Example</i>, pg. 263, cap.3.5 [4], Probl. <i>3.7, 3.8</i>, pg. 348, cap. 3.15 [4]). Aplicații:</p> <ul style="list-style-type: none"> • Utilizarea simulatorului WinDLX: configurarea WinDLX, încărcarea programelor de test, simularea benchmark-urilor, sistemul de ferestre. Investigații arhitecturale utilizând simulatorul DLX: benchmark - urile Fact.s, Invers.s, rutina Input.s, probleme propuse spre rezolvare. Evidențierea efectului defavorabil al hazardurilor RAW în cadrul arhitecturilor pipeline scalare asupra performanței de procesare. Modificarea programului de test fact.s și simularea hazardurilor structurale și a conflictelor de nume (WAW). Vizualizarea tehnicii de forwarding – câștigul de performanță obținut. 	2
--	---------	---	---

	Curs 11	<p>Principiile mașinilor cu execuție multiplă. Considerații generale. Taxonomii (procesoare superscalare și VLIW). Tehnici de optimizare statică a programelor (locale – <i>List Scheduling</i>; <i>Loop Unrolling</i> și <i>Software Pipelining</i>, globale – <i>Trace Scheduling</i>). Avantaje / Dezavantaje față de procesoarele superscalare. Aplicabilitate – sisteme dedicate (procesoare de semnal, procesoare multimedia). Aplicații:</p> <ul style="list-style-type: none"> • Utilizarea simulatorului VLIW-DLX: ilustrarea principiilor fundamentale ale procesării VLIW (very long instruction word). Rolul software-ului în detecția și eliminarea hazardurilor RAW. Relația dintre instrucțiunea multiplă și instrucțiunile RISC primitive și independente, care vor fi alocate unităților de execuție în conformitate strictă cu poziția lor în instrucțiunea multiplă (număr/latențe). Optimizări software în procesoarele VLIW (loop unrolling, software pipelining). Avantaje / Dezavantaje față de procesoarele superscalare. Aplicabilitate – sisteme dedicate (procesoare de semnal, procesoare multimedia). Aplicații rezolvate: (i) execuția pe un procesor VLIW a unui program care înmulțește două matrici pătratice. (ii) să se execute pe procesorul VLIW-DLX bucla de program care realizează translatarea unui tablou de valori flotante cu o constantă dată, în circumstanțele: a) program neoptimizat, b) program optimizat cu <i>loop-unrolling</i> (2 iterații), c) program optimizat cu <i>loop-unrolling</i> (4 iterații) și d) program optimizat cu <i>software-pipelining</i>. 	2
	Curs 12	<p>Principiile mașinilor cu execuție multiplă. Considerații generale. Modele și algoritmi de procesare a instrucțiunilor în microprocesoarele superscalare (în order / out of order). Resurse arhitecturale implicate. Influența caracteristicilor arhitecturale specifice procesoarelor RISC superscalare asupra consumului de putere și respectiv asupra performanței. Predictorul de salturi – una din principalele cauze care determină consum ridicat de putere în arhitecturi. Evidențierea efectului negativ al predicției greșite a instrucțiunilor de ramificație atât asupra vitezei de execuție a procesorului cât și asupra energiei consumate (<i>simulatorul PSATSim</i>).</p>	2
	Curs 13	<p>Seminar de aplicații. (II) Problematika hazardurilor în microprocesoarele pipeline. Graficul dependentelor de date, timpul de execuție al unui program înainte și după optimizare (renaming, forwarding) – probl. 7, 8, 2, 23, 28 cap.11 [3]. Recapitulare finală. Parcurgerea conceptelor esențiale.</p>	2
Total ore curs:			28

Conținut laborator/ Lab contents	Lab 1	Exemplificare de sisteme cu microprocesoare / microcontrollere folosite în industrie. Avantajele utilizării sistemelor dedicate față de cele universale. Prezentare a conceptelor fundamentale din teoria sistemelor cu microprocesoare. Asemănări / deosebiri microprocesoare vs. Microcontrollere. Noțiuni introductive și prezentarea generală structurii laboratorului.	2
	Lab 2	Prezentare generală a mediului și metodologiei de lucru, a resurselor utilizate. Instalarea instrumentelor hardware / software folosite în dezvoltarea de proiecte cu microprocesoare. Prezentare placă de dezvoltare EasyPic Fusion v7, Microcontroller-ul PIC32MX795F512L. Instalarea și prezentarea mediului de programare MicroC Pro for PIC32 IDE. Compilarea și depanarea programelor de test aferente plăcii; Prezentarea temelor de proiecte; Conversii de numere dintr-o baza în alta.	2
	Lab 3	Introducere în programarea microcontrollerelor (low level). Structura generală a unui program embedded. Aplicație „Hello PIC”: Fișierele „.c” și „.h”; Funcția main; Bucla infinită. Implementarea, compilarea și încărcarea programului de test “Hello PIC” pe microcontrollerul PIC32MX795F512L. Testarea funcționării. Utilizarea bibliotecilor din MicroC. Definiere funcții și prototipuri. Funcții inline; Directive: „#define” și „#include”; Tipuri de date. Variabile externe; Operatori C exemplificați pe exemple practice. Repartizare temelor de proiect.	2
	Lab 4	Porturi I/O. Convertor Analog Digital. Porturile I/O digitale și analogice. Scriere respectiv citire; Conversia analog-digitală (ADC); Aplicații propuse: a. Utilizând ledurile de pe placa de dezvoltare și butoanele, configurați microcontrollerul astfel încât să răspundă la două butoane, când unul este apăsat să se aprindă toate ledurile, iar când celalalt este apăsat să se stingă; b. Alegeți un PORTx și creați efectul de shiftare (deplasare) de la stânga la dreapta și invers. c. Citirea temperaturii folosind senzor-ul LM35;	2

	Lab 5	Ecran TFT (<i>Thin Film Transistor</i>). Touch panel. Utilizare API. Aplicații propuse: a. Citirea temperaturii folosind senzor-ul LM35 și afișarea valorii pe ecranul TFT; b. Determinarea distanței folosind senzor-ul IR Sharp și afișarea valorii pe ecranul TFT; c. Desenarea unor figuri geometrice care se modifică în timp.	2
	Lab 6	Prezentare Timere: noțiuni generale de funcționare a timerelor, configurarea și utilizarea acestora. Prezentare mod de funcționare intern al timerelor în cadrul arhitecturii sistemului cu microcontrolere și exemplificarea utilității acestora în proiecte viitoare. Utilizarea întreruperilor generate de timere și folosirea lor pentru gestiunea task-urilor într-un sistem. 1. Timer. Factorul de postscalare respectiv prescalare; 2. Polling (înterogare); 3. Întreruperi; 4. Aplicații propuse: a. Led blinking folosind întreruperi (overflow / underflow la 1 secundă). b. Ajustarea volumului buzzer-ului de pe placă utilizând joystick-ul. Buzzer-ul se comandă cu un semnal PWM cu factor de umplere variabil, frecvența = 3kHz.	2
	Lab 7	Evaluare parțială a cunoștințelor.	2
	Lab 8	Introducere în programarea de nivel înalt. Diferența dintre programarea de nivel înalt și cea de nivel jos (low level). Instalare Visual Studio 2010. Introducere în C#, lucrul în consolă, exerciții de conversie binară, blocuri de decizie. Aplicații de tip consolă. 1. Crearea unui proiect de tip consolă „Hello Word” utilizând mediul de dezvoltare Microsoft Visual Studio; 2. Aplicații: a. Scrieți un program de tip consolă care citește de la tastatură numele dumneavoastră și îl afișează pe ecran; b. Scrieți un program de tip consolă care citește de la tastatură o listă de numere și calculează media lor aritmetică; c. Scrieți un program de tip consolă care vă permite să vă jucați cu calculatorul Rock, Paper or Scissors. Calculatorul va alege aleator una din cele trei variante.	3

	Lab 9	Introducere în programarea orientată obiect (POO), lucrul cu clase și obiecte, necesare pentru crearea de interfețe vizuale (WPF). C# - aplicații de tip Windows Forms. 1. Crearea unui proiect de tip Windows Forms utilizând mediul de dezvoltare Microsoft Visual Studio; 2. Prezentare celor mai uzuale componente vizuale din toolbox (button, list, edit box, etc). Utilizarea lor în practică;	3
	Lab 10	Realizarea comunicării între interfața grafică și microcontroller prin interfața serială. Vizualizare process și crearea unui protocol de comenzi. Detaliere și proiectarea unui sistem complex de mesaje pe LCD ce ilustrează starea unui PC. Prezentarea noțiunilor de <i>socket</i> , accesul la interfața serială din C# și transferul datelor dinspre microcontroller către PC. Această lucrare va presupune îmbinarea tuturor cunoștințelor de până acum în crearea unei monitorizări simpliste a unui sistem. Generare de date și feedback.	2
	Lab 11	Aplicații cu LCD-ul grafic: desenare forme. Se va crea o interfață grafică în C# de desenare, ce va transmite desenul prin interfața serială microcontroller-ului. Acesta din urmă va afișa desenul pe LCD-ul plăcii de dezvoltare.	2
	Lab 12	Recapitulare urmată de evaluare practică privind materia studiată la laborator.	4
Total ore laborator			28

<p>Conținut proiect/ Project contents</p>	<p>Pro 1</p>	<p>Alegerea temei de proiect. Conceperea planului de cercetare-dezvoltare pentru proiect. Aflarea bibliografiei necesare.</p> <p>Teme disponibile:</p> <ul style="list-style-type: none"> • Colecție de instrumente de simulare folosind animație interactivă pentru: analiza sistemului ierarhic de memorie în arhitecturile multicore. • Simulator grafic pentru microarhitectura LC-3. • Simularea interfeței procesor - cache pentru o arhitectură risc superscalară parametrizabilă • Implementarea schemelor clasice de predicție în procesoarele superscalare avansate: BTB, Gag • Optimizarea schemelor de predicție pentru ramificațiile de program în procesoarele superscalare avansate: markoviene, PPM, neuronale (perceptron simplu și multistrat), O-GEHL • Implementarea unui Automatic Design Space Exploration aferent unei arhitecturi superscalare (Hatfield), pentru determinarea optimului din punct de vedere a ratei de procesare folosind algoritmi de căutare locali (în vecinătatea unei configurații microarhitecturale) de tip hill climbing, simulated annealing. • Implementarea unui Automatic Design Space Exploration aferent unei arhitecturi superscalare (vezi simulatorul PSATSim – arhitectura PowerPC), pentru analiza multiobiectiv (performanță, consum de energie) folosind tehnici de optimizare non-Pareto (suma ponderată a obiectivelor, algoritmul VEGA (Vector Evaluated Genetic Algorithm)). • Implementarea unui Automatic Design Space Exploration aferent unei arhitecturi superscalare (vezi simulatorul PSATSim – arhitectura PowerPC), pentru analiza multiobiectiv (performanță, consum de energie) folosind tehnici de optimizare Pareto: NSGA-II (Non-dominated Sorting Genetic Algorithm), SPEA2 (Improving the strength Pareto evolutionary algorithm) 	<p>2</p>
---	--------------	---	----------

	Pro 2	Prezentarea și discutarea planului de realizare a proiectului. Documentarea în vederea implementării. Proiectul trebuie să demonstreze cunoașterea științifică a temei abordate, să conțină elemente de originalitate în dezvoltarea sau soluționarea temei, precum și modalități de validare științifică a acestora. Stabilirea sarcinilor în cadrul echipei de cercetare / realizare a proiectului.	2
	Pro 3	Structura proiectului: a) Introducere (în care se motivează abordarea temei și se justifică modul în care se integrează tema în problematica specifică arhitecturii microprocesoarelor); b) Analiza critică a stadiului actual privind tema abordată (bazată pe interpretarea personală a autorului lucrării de proiect a informațiilor din literatura de specialitate referitoare la tema lucrării); c) Contribuțiile studentului la rezolvarea unor aspecte ale problematicii implicate de tema proiectului (este partea ce prezintă realizările proprii ale autorului lucrării de proiect: cercetări experimentale, dezvoltări teoretice noi, proceduri și metodologii de proiectare, produse informatice noi, interpretări originale ale unor date din literatura de specialitate etc.); d) Concluzii (se prezintă principalele concluzii desprinse pe baza studiilor și cercetărilor realizate cu ocazia elaborării lucrării de disertație și se subliniază contribuțiile personale, aplicabilitatea și utilitatea lor și modul în care contribuie acestea la dezvoltarea cunoașterii în domeniul temei); e) Bibliografie.	2
	Pro 4	Întocmirea lucrării. Stabilirea metodologiei, a tools-urilor necesare (mediu de dezvoltare, simulatoare, compilatoare, debugger-e, link-editoare, benchmark-uri standardizate).	2
	Pro 5	Implementarea software a algoritmilor și metodelor în vederea testării ipotezelor și a verificării experimentale.	2
	Pro 6	Realizarea documentatiei (raportul) de cercetare. Accentuarea laturii practice a cercetării întreprinse în cadrul proiectului. Prezentarea succintă a noțiunilor teoretice care vizează exclusiv cercetarea practică realizată.	2
	Pro 7	Fiecare student va prezenta un raport tehnic (TR), sursele aplicației și prezentarea PowerPoint (PPT) în care va descrie soluția propusă. Validarea raportului se face de către titularul disciplinei.	2
	Total ore proiect:		14

Metode de predare/ Teaching methods	Predare cu videoprojectorul direct din calculator		Limba de predare/ Language of instruction	Română
Sisteme de evaluare/ Assessment methods	Activități aplicative	10%	1. Teme de curs / pondere = 10% (nCPE) 2. Referate de disciplină = 20% (nCPE) 3. Lucrări practice = 70% (CPE)	
	Proiect	25%	CPE	
	Examen parțial	10%	nCPE	
	Examen de semestru	55%	CEF	
CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală;				

Competențe specifice disciplinei	
Competențe profesionale	C3 – Aplicarea cunoștințelor, conceptelor și metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje și tehnici de programare C4 – Proiectarea și utilizarea unor aplicații hardware și software de complexitate redusă specifice electronicii aplicate
Competențe transversale	CT2 - Definierea activităților pe etape și repartizarea acestora subordonațiilor cu explicarea completă a îndatoririlor, în funcție de nivelurile ierarhice, asigurând schimbul eficient de informații și comunicarea interumană. CT3 - Adaptarea la noile tehnologii, dezvoltarea profesională și personală, prin formare continuă folosind surse de documentare tipărite, software specializat și resurse electronice în limba română și, cel puțin, într-o limbă de circulație internațională.
Competențele generale sunt menționate în Fișa specializării	

Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)	
Obiectivul general al disciplinei	Înșușirea cunoștințelor legate de funcționarea sistemelor cu microprocesoare de tip "desktop", "notebook", server sau încorporate, de proiectarea și simularea funcționării acestora, diagnosticarea și depanarea sistemelor numerice cu microprocesor și de realizarea unor aplicații scrise în limbaje de asamblare specifice unor microprocesoare cunoscute, atât de uz general cât și încorporate utilizând sisteme de programe specializate.
Obiectivele specifice	<ul style="list-style-type: none"> • Prezentarea generațiilor actuale de microprocesoare, precum și un set cât mai complet de tehnici pentru a interfața aceste microprocesoare cu componentele unui sistem (memorie) și, implicit, cu lumea reală (periferice) • Diagnosticarea și depanarea unui sistem cu microprocesor • Cunoașterea și utilizarea unor componente arhitecturale specializate pe aplicații (microprocesoare CISC și RISC, interfețe specializate și controllere de magistrale, procesoare superscalare și VLIW) • Proiectarea, dezvoltarea și implementarea unor scheme hardware și

	aplicații software în vederea realizării unor sisteme cu microprocesoare • Dezvoltarea unor simulatoare software care să modeleze și să descrie funcționarea unei arhitecturi de microprocesor
--	---

Referințe bibliografice recomandate/ Recommended reading	1. Vințan Lucian – Organizarea și proiectarea microarhitecturilor, http://webspaces.ulbsibiu.ro/lucian.vintan/html/Organizarea.pdf .
	2. Florea Adrian – Introducere în Știința și Ingineria Calculatoarelor. Interfața Hardware – Software, Editura Matrix ROM, București, ISBN 978-973-755-264-8, 2007 (313 pg.).
	3. Florea Adrian, Vințan Lucian – Simularea și optimizarea arhitecturilor de calcul în aplicații practice, Editura Matrix ROM, București, ISBN 973-685-605-4, 2003 (443 pg. + CD atasat).
	4. Patterson David, Hennessy John – Computer Architecture: A quantitative Approach, Morgan Kaufmann Publishers, Fourth Edition, 2006.
	5. Florea Adrian – Predicția dinamică a valorilor în microprocesoarele generației următoare, Editura MatrixROM, București, ISBN 973-685-980-0, 2005, (413 pg. + CD atasat).
Referințe bibliografice suplimentare/ More references	6. Patt Yale, Patel Sanjay – „Introduction to Computing Systems: from bits & gates to C & beyond”, McGraw-Hill Higher Education, 2001.
	7. Patterson David, Hennessy John – Computer Organisation and Design: The Hardware/Software Interface, Third Edition, Elsevier, 2005, ISBN: 1-55860-604.
	8. Baruch Zoltan Francisc – Structura sistemelor de calcul, Editura Albastra Cluj Napoca, 2005, ISBN 973-650-143-4.
	9. http://www.ecs.umass.edu/ece/koren/architecture/
Regulamentul disciplinei	

Data completării: 26.10.2019

Data avizării în Departament:.....

	Grad didactic, titlul, prenume, numele	Semnătura
Titular disciplină	Prof.univ.dr.ing. Adrian FLOREA	
Director de departament	Prof. dr. ing. Daniel VOLOVICI	