

FIȘA DISCIPLINEI

1. Date despre program

Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
Facultatea	Facultatea de Inginerie
Departament	Departamentul de Calculatoare și Inginerie Electrică
Domeniul de studiu	Calculatoare și Tehnologia Informației
Ciclul de studii	Licență
Specializarea	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	Bazele logice ale calculatoarelor			
Codul cursului	Tipul cursului	An de studiu	Semestrul	Număr de credite
390451060611FO06	Obligatoriu	1	1	5
Tipul de evaluare	Categororia formativă a disciplinei (DF=fundamentală.; DD=domeniu; DS=specialitate; DC=complementară)			
Examen	DF			
Titular activității curs	Prof. dr. ing. Adrian FLOREA			
Titular activității seminar / laborator/ proiect	Prof. dr. ing. Adrian FLOREA			

3. Timpul total estimat

Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână				
Curs	Seminar	Laborator	Proiect	Total
2	2	-	-	4
Extinderea disciplinei în planul de învățământ – Total ore din planul de învățământ				
Curs	Seminar	Laborator	Proiect	Total (<i>NOAD_{sem}</i>)
28	28	-	-	56

Distribuția fondului de timp pentru studiu individual		Nr.ore
Studiul după manual, suport de curs, bibliografie și notițe		45
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren		8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri		10
Tutoriat:		4
Examinări:		2
Total ore alocate studiului individual (<i>NOSI_{sem}</i>)		69
Total ore pe semestru (<i>NOAD_{sem} + NOSI_{sem}</i>)		125

4. Precondiții (acolo unde este cazul)

De curriculum	Cunoștințe de bază de matematică. Nu sunt necesare alte cunoștințe prealabile. Nu sunt obligatorii cunoștințe de programare.
De competențe	

5. Condiții (acolo unde este cazul)

De desfășurare a cursului	Participare activă, lectura suportului de curs Tablă, videoproiector
De desfășurare a sem/lab/pr	Elaborarea și susținerea lucrărilor planificate. Sală dotată cu calculatoare cu sistem de operare Windows XP.

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> • Operarea cu fundamente științifice, ingineresti și ale informaticii • Proiectarea componentelor hardware, software și de comunicații • Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor • Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații
Competențe transversale	<ul style="list-style-type: none"> • Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate • Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Cunoașterea și înțelegerea principiilor generale ale disciplinei • Cunoașterea și operarea adecvată cu noțiunile specifice disciplinei • Identificarea principalelor surse de informare • Analiza critică a modelelor teoretice, ideilor și a abordărilor consacrate • Aptitudini de realizare a unei teme și a unui raport aferent • Dezvoltarea abilităților de cercetare individuală
Obiectivele specifice	<ul style="list-style-type: none"> • Înțelegerea principiilor de bază ale structurii sistemelor de calcul • Proprietățile algoritmilor, reprezentarea acestora prin scheme logice și prin pseudocod, respectarea principiilor programării structurate în procesul de elaborare a algoritmilor • Identificarea tipurilor de informații reprezentate într-un calculator; reprezentarea informațiilor; conversii de valori între sisteme de numerație; aritmetică binară; operații logice, legile lui De Morgan • Identificarea circuitelor digitale combinaționale și secvențiale care compun o microarhitectură generică (LC-3): tranzistoare MOS, porți logice (fundamentale), bistabili, structuri logice (Multiplexoare, Decodificatoare, Sumatoare), registre, memorii. • Înțelegerea modelului de execuție von Neumann, (componentele modelului și ciclul instrucțiunii) aferent unei arhitecturi simple LC-3, a limbajului de asamblare propriu. Cunoașterea interfeței cu tastatura și monitorul, a apelurilor sistem (întreruperi software reprezentând servicii ale sistemului de operare prin instrucțiuni dedicate), modurile de adresare pentru instrucțiunile cu referire la memorie (load / store), mecanismul de apel și revenire din subrutină.

	<ul style="list-style-type: none"> Exemplificarea la nivelul limbajului C și implicațiile în hardware a structurilor de control, a diverselor tehnici de alocare a memoriei, lucrul cu stiva, apel de funcții și revenire, stivele de date aferente funcțiilor, transmiterea parametrilor în cazul apelului, recursivitate, pointeri, tablouri și structuri elementare de date.
--	--

8. Conținuturi

Curs		Nr. ore
Curs 1	Introducere în sistemele de procesare a informației. Calculatorul – dispozitiv universal de calcul. Structură. Nivele de transformare. Nevoia de abstractizare și "Când eșuează abstractizarea?"	2
Curs 2	Reprezentarea informațiilor. Entropia informațională. Exemplificare diferența <i>dată – informație</i> . Sisteme de numerație. Conversii de valori între sisteme de numerație. Coduri pentru detectarea și corectarea erorilor. Aritmetică binară. Operații logice. Depășirea domeniului de reprezentare. Extensia semnului.	2
Curs 3	Algoritmi. Caracteristicile algoritmilor. Reprezentarea algoritmilor prin scheme logice și prin pseudocod.	2
Curs 4	Structuri logice digitale. Tranzistori. Porți logice. Circuite logice combinaționale (Decodificatorul, Multiplexorul, Sumatorul). Proiectarea unei unități aritmetico-logice.	2
Curs 5	Elemente primare de memorare. Bistabili. Regiștrii. Circuite numărătoare. Memorii. Configurarea unor memorii folosind circuite primare. Automate secvențiale și programabile. Automate cu număr finit de stări.	2
Curs 6	Modelul arhitectural <i>von Neumann</i> . Componente de bază. Principiile procesării instrucțiunilor. Ciclul instrucțiunii. Ciclul instrucțiunii văzut ca un automat cu număr finit de stări. Exemplificare pe simulatorul <i>LC3 Microarchitecture Simulator</i> („ <i>a graphical simulator for the LC3 processor, concentrating on the micro-architecture level</i> ”). Tipuri de instrucțiuni. Ceasul procesorului. Oprirea sistemului de calcul.	2
Curs 7	LC-3 – Arhitectura setului de instrucțiuni. Calea fluxului de date. Organizarea memoriei la LC-3.	2
Curs 8	Limbajul de asamblare aferent arhitecturii LC-3. Asamblorul. Etapele generării codului mașină. Tabela de simboluri.	2
Curs 9	Apeluri de subrutine – directe și indirecte. Reveniri. Întreruperi software la nivel <i>low</i> . Salvarea și restaurarea regiștrilor. Strategiile " <i>caller-save</i> " respectiv " <i>callee save</i> ".	2
Curs 10	Stiva – structură. Principiu de funcționare. Operații aferente (<i>Push & Pop</i>). Implementare hardware și software. Modul de lucru prin întreruperi hardware. Operații aritmetice folosind stiva. Rolul stivei în tratarea întreruperilor hardware imbricate.	2
Curs 11	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. Translatarea codului de la nivel <i>high (C)</i> la nivel <i>low (asamblare – LC3)</i> .	2
Curs 12	Introducere în <i>recursivitate</i> . Recursivitate vs. Recurență. Rolul stivei în implementarea recursivității. Compararea dintre <i>recursiv</i> și <i>iterativ</i> în alegerea algoritmului de rezolvare a problemelor. Avantaje / dezavantaje. Implementarea	2

	recursivității la nivelul stivei de date. Eroarea <i>Stack Overflow</i> . Tipuri de funcții recursive. Eliminarea recursivității.	
Curs 13	Pointeri. Tablouri. Transferul parametrilor prin adresă. Pointeri spre funcții. Legătura dintre nivelul <i>high</i> și <i>low</i> văzută prin intermediul modurilor de adresare. Alocarea și accesarea indirectă de variabile.	2
Curs 14	Recapitulare finală. Parcurgerea conceptelor esențiale.	2
Total ore curs:		28
Seminar		Nr. ore
Sem 1	Reprezentarea numerelor întregi: fără semn, cu semn (MS, C1, C2). Conversii din diverse baze. Reprezentarea numerelor fracționare: standardul IEEE 754. Operații aritmetico-logice în binar. Coduri ASCII și Unicode. Conversia din caracter în numeric și invers.	4
Sem 2	Algoritmi. Caracteristicile algoritmilor. Reprezentarea algoritmilor prin scheme logice și prin pseudocod.	2
Sem 3	Proiectarea circuitelor digitale folosind tranzistoare. Operații logice folosind porți logice fundamentale (NAND, NOR). Implementarea multiplexoarelor, sumatoarelor, decodificatoarelor.	2
Sem 4	Proiectarea unor circuite de memorie folosind circuite primare. Aplicații cu automate secvențiale și programabile. Utilizarea simulatorului <i>memory.exe</i> , care descrie operațiile de citire – scriere dintr-o memorie cu 4 locații având 3 biți de date fiecare.	2
Sem 5	Modelul de procesare „von Neumann”. Codificarea instrucțiunilor. Determinarea numărului de instrucțiuni executate pe un anumit microprocesor într-o secundă. Determinarea numărului de accese la memorie pentru fiecare tip de instrucțiune.	2
Sem 6	Examinare parțială din materia parcursă.	2
Sem 7	Limbajul de asamblare aferent arhitecturii LC-3. Etapele generării codului mașină. Tabela de simboluri.	2
Sem 8	Aplicații în LC-3 cu întreruperi software și apeluri imbricate de subrutine.	2
Sem 9	Aplicații cu stiva și operațiile <i>Push / Pop</i> . Codificarea ASCII to Binary și Binary to ASCII în LC-3.	2
Sem 10	Aplicații scrise <i>recursiv</i> și <i>iterativ</i> . Avantaje / dezavantaje. Implementarea recursivității la nivelul stivei de date. Evidențierea erorii <i>Stack Overflow</i> .	2
Sem 11	Aplicații cu pointeri și tablouri. Transferul parametrilor prin adresa. Pointeri spre funcții. Legătura dintre nivelul <i>high</i> și <i>low</i> văzută prin intermediul modurilor de adresare. Alocarea și accesarea indirectă de variabile.	2
Sem 12	Examinare parțială din materia parcursă.	2
Sem 13	Ghid de utilizare <i>LC-3 Edit</i> și <i>LC-3 Simulate</i> . Implementarea software a aplicațiilor anterior prezentate la curs și seminar. Depanarea și simularea folosind cele două utilitare.	2
Sem 14		-
Total ore Seminar		28

Metode de predare

Prelegeri, problematizări, studii de caz, exerciții, conversații, explicații, demonstrații și dezbateri. Metodele de predare la curs se bazează pe folosirea videoprojectorului, dar în	Limba de predare	Română
--	------------------	--------

anumite cazuri în care se impune se folosește creta și tabla pentru diverse demonstrații sau rezolvări de exemple / probleme.

La seminar sunt rezolvate o serie de probleme la tabla, se propun și se rezolvă (fie ca temă, fie ca test, fie ca pure exerciții) teste grilă. De asemenea, sunt folosite calculatoarele pentru rezolvarea unor probleme de programare în limbaj de asamblare folosind simulatorul LC-3 și *memory* (vezi mai jos).

Simulatorul LC-3 (for *Little Computer 3*) – disponibil atât sub sistem de operare Windows cât și Linux. LC-3 descrie funcționarea unei arhitecturi pe 16 biți; arhitectura înglobează cele mai importante caracteristici ale procesoarelor Intel 8088 – folosit în primul sistem IBM PC în 1981, Motorola 68000 – folosit la Apple Macintosh 1984, sau Intel Pentium III integrat în sistemele anilor 2000.

Simulatorul arhitecturii LC-3 permite testare / depanare a programelor scrise în limbaj de asamblare, stabilire de puncte de întrerupere. De la adresa <http://users.ece.utexas.edu/~ambler/ee306/software&doc.htm> pot fi descărcate în mod gratuit pentru utilizare în scop didactic atât simulatorul LC-3 (LC301.exe), un simulator realizat în Macromedia Flash Player versiunea 6, care descrie operațiile de citire – scriere dintr-o memorie cu 4 locații având 3 biți fiecare (*memory.exe*), precum și ghidul de utilizare al simulatorului LC-3. La adresa <http://webspace.ulbsibiu.ro/adrian.florea/html/simulatoare/alu/index.html> se găsesc aplicații *Flash* realizate sub îndrumarea titularului de disciplină care descriu funcționarea circuitelor logice digitale combinaționale (Sumatorul complet, Unitate Aritmetico-Logică pe 1 bit și un circuit de deplasare).

Bibliografie

Referințe bibliografice recomandate	1. Patt Yale, Patel Sanjay – „ <i>Introduction to Computing Systems: from bits & gates to C & beyond</i> ”, McGraw-Hill Higher Education, 2003.
	2. Patterson David, Hennessy John – <i>Computer Organisation and Design: The Hardware/Software Interface</i> , Third Edition, Elsevier, 2005, ISBN: 1-55860-604.
	3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. – <i>Introduction to Algorithms</i> . McGraw-Hill, New York, 1990.
	4. Pop Vasile – <i>Analiza și sinteza dispozitivelor numerice</i> . Curs litografiat, Institutul Politehnic Timișoara, 1986, vol. I și II.
	5. Vințan Lucian – <i>Organizarea și proiectarea microarhitecturilor</i> , http://webspace.ulbsibiu.ro/lucian.vintan/html/Organizarea.pdf .
	6. Mârșanu Radu – „ <i>Sisteme de calcul</i> ”, Manual pentru licee de informatică, cls. IX-a, Editura Didactică și Pedagogică, București, 1996.
	7. Florea Adrian – <i>Introducere în Știința și Ingineria Calculatoarelor. Interfața Hardware – Software</i> , Editura Matrix ROM, București, ISBN 978-973-755-264-8, 2007 (313 pg.).
Referințe bibliografice suplimentare	8. Zaharia Mihai, Leon Florin – „ <i>Limbajul C de la Zero la Student</i> ”, Editura Politehnicum Iasi, 2004.
	9. Dana Lica, ș.a. – <i>Fundamentele programării</i> , Editura L&S Soft, București.
	10. Florea Adrian – <i>Predicția dinamică a valorilor în microprocesoarele generației următoare</i> , Editura MatrixROM, 2005.
	11. Golometry A. – <i>Proiectarea Translatoarelor</i> , Editura Universității "Lucian Blaga", Sibiu, 1997.

12. Gorgan D., Sebestyen-Pal Gh., <i>Computer Design</i> , Editura albastra, Cluj-Napoca,. ISBN 973-650-123-X, 2005.
13. Katz R. H., Borriello G., <i>Contemporary Logic Design (2nd Edition)</i> , Prentice Hall, 2005.

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

--

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Ponderea în nota finală	Obs.*
Curs	Teste pe parcursul semestrului	Lucrare scrisă	20%	CPE
	Examen de semestru	Examen scris	55%	CEF
	Alte activități: prezența la curs	-	5%	nCPE
Seminar	Activități aplicative	Evaluare orală, aplicații realizate Fișă de evaluare seminar	10%	nCPE
	Teme / referate	Verificare săptămânală	10%	nCPE

Standard minim de performanță

- Cunoașterea, înțelegerea și explicarea noțiunilor elementare specifice științei și ingineriei calculatoarelor.
- Interes constant manifestat pentru însușirea disciplinei.
- Îndeplinirea condițiilor minime obligatorii (50%) în privința temelor de casă, referatelor și a testelor date pe parcursul semestrului.

(*) CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală;

Data completării: ...10/09/2016..

Data avizării în Departament:.....

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