

FIȘA DISCIPLINEI

Denumirea disciplinei :		BAZELE LOGICE ALE CALCULATOARELOR			
Codul disciplinei:		390451071117FO06			
Programul de studii:		INGINERIA SISTEMELOR MULTIMEDIA			
Catedra:		CALCULATOARE SI AUTOMATIZARI			
Facultatea:		DE INGINERIE „Hermann Oberth”			
Universitatea:		„Lucian Blaga” Sibiu			
Anul de studiu:	I	Semestrul	I	Tipul de evaluare finală	C
Regimul disciplinei (DI=obligatorie/ DO=opțională/DF=liber aleasă):			DI	Numărul de credite:	5
Categorია formativă a disciplinei (DF=fundamentală.; DI=ingineresci; DS=specialitate; DC=complementară)					DF
Total ore din planul de învățământ	56			Total ore pe semestru:	56
Titularul disciplinei: Conf.univ.dr.ing. Adrian FLOREA					

Numărul total de ore (pe semestru) din planul de învățământ					
Total ore/ semestru	C	S	L	P	Total
	28	28			56

Obiective:	<p>Acest curs reprezintă o introducere axată pe înțelegerea principiilor fundamentale ale științei și ingineriei calculatoarelor. Abordare <i>bottom-up</i> centrată în principal pe două componente:</p> <ul style="list-style-type: none"> • Structura de bază a calculatoarelor folosind simulatorul LC-3 (<i>Little Computer</i> – www.ece.utexas.edu/~ambler/ee306/Software&Doc/LC3WinGuide.pdf) • 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. <p>Aspectul novator al abordării constă în prezentarea interfeței arhitectură (ISA, limbaj de asamblare specific) – aplicație de nivel înalt (incluzând compilatorul ca și generator de cod pentru respectiva arhitectură).</p> <p>Relativ la interfata Hardware – Software, cu toate ca putem fi specialiști într-una din cele doua, trebuie intelese capabilitatile si limitarile ambelor componente ale sistemului de calcul.</p>
-------------------	---

Competențe specifice disciplinei	<p>1. Cunoaștere și înțelegere:</p> <ul style="list-style-type: none"> • Înțelegerea principiilor de bază ale structurii sistemelor de calcul • Proprietățile algoritmilor, Reprezentarea acestora prin <i>scheme logice</i> și prin <i>pseudocod</i>, 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 <i>von Neumann</i>, (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. • 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.
	<p>2. Explicare și interpretare:</p> <ul style="list-style-type: none"> • Cunoașterea și utilizarea adecvată a noțiunilor specifice științei calculatoarelor, explicarea și interpretarea conținuturilor teoretice și practice ale disciplinei • Explicarea unor noțiuni din discipline sau domenii nestudiate încă și care vor fi introduse la un nivel sumar • Explicații la întrebările studenților • Indicații pentru întocmirea referatelor și temelor de casă
	<p>3. Instrumental – aplicative:</p> <ul style="list-style-type: none"> • Scrierea de programe în limbaj de asamblare LC-3, realizarea de scheme logice / pseudocod pentru orice problemă propusă. • Rezolvarea unor probleme de proiectare folosind circuite de memorie, regiștrii, automate programabile.
	<p>4. Atitudinale:</p> <ul style="list-style-type: none"> ♦ Exprimarea unui mod de gândire creativ, în structurarea și rezolvarea problemelor ♦ Conștientizarea impactului social, economic al științei calculatoarelor ♦ Formarea obișnuințelor de a recurge la concepte și metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme. ♦ Manifestarea unor atitudini favorabile față de știință și de cunoaștere în general ♦ Manifestarea unei atitudini responsabile și pozitive față de profesia didactică

Conținutul tematic (descriptori)	TEMATICA CURSURILOR		
	Nr. ct.	Denumirea temei	Nr. ore
	1.	Introducere în sistemele de procesare a informației. Calculatorul – dispozitiv universal de calcul. Structură. Nivele de transformare (abstractizare).	2
	2.	Reprezentarea informațiilor. Entropia informațională. Exemplificare diferența <i>dată</i> – <i>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
	3.	Algoritmi. Caracteristicile algoritmilor. Reprezentarea algoritmilor prin scheme logice și prin pseudocod.	2
	4.	Structuri logice digitale. Tranzistori. Porți logice. Circuite logice combinaționale (Decodificatorul, Multiplexorul, Sumatorul). Proiectarea unei unități aritmetico-logice.	2
	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
	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. Tipuri de instrucțiuni. Ceasul procesorului. Oprirea sistemului de calcul.	2
	7.	LC-3 – Arhitectura setului de instrucțiuni. Călea fluxului de date. Organizarea memoriei la LC-3.	2
	8.	Limbajul de asamblare aferent arhitecturii LC-3. Asamblorul. Etapele generării codului mașină. Tabela de simboluri.	2
	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
	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 intreruperilor hardware imbricate.	2
	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

12.	Introducere în <i>recursivitate</i> . Recursivitate vs. Recurență. Rolul stivei în implementarea recursivității. Comparația dintre <i>recursiv</i> și <i>iterativ</i> în alegerea algoritmului de rezolvare a problemelor. Avantaje / dezavantaje. Implementarea recursivității la nivelul stivei de date. Eroarea <i>Stack Overflow</i> . Tipuri de funcții recursive. Eliminarea recursivității.	2
13.	Pointeri. 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
14.	Recapitulare finală. Parcurgerea conceptelor esențiale.	2
TEMATICA SEMINARIILOR/LABORATOARELOR/PROIECTULUI		
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
2.	Algoritmi. Caracteristicile algoritmilor. Reprezentarea algoritmilor prin scheme logice și prin pseudocod.	2
3.	Proiectarea circuitelor digitale folosind tranzistoare. Operații logice folosind porți logice fundamentale (NAND, NOR). Implementarea multiplexoarelor, sumatoarelor, decodificatoarelor.	2
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
5.	Modelul de procesare „ <i>von Neumann</i> ”. 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
6.	Examinare parțială din materia parcursă.	2
7.	Limbajul de asamblare aferent arhitecturii LC-3. Etapele generării codului mașină. Tabela de simboluri.	2
8.	Aplicații în LC-3 cu întreruperi software și apeluri imbricate de subrutine.	2
9.	Aplicații cu stiva și operațiile <i>Push / Pop</i> . Codificarea ASCII to Binary și Binary to ASCII în LC-3.	2
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
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

	12.	Ghid de utilizare <i>LC-3 Edit</i> și <i>LC-3 Simulate</i> . Implementarea software a aplicațiilor anterior prezentate la curs și seminar. Depanare și simularea folosind cele doua utilitare.	4
	13.		
	14.		

Metode de predare / seminarizare	<p>Metodele de predare la curs se bazeaza pe folosirea videoproietorului, dar in anumite cazuri in care se impune se foloseste creta si tabla pentru diverse demonstratii sau rezolvari de exemple / probleme.</p> <p>La seminar sunt rezolvate o serie de probleme la tabla, se propun si se rezolva (fie ca tema, fie ca test, fie ca pure exercitii) teste grila. De asemenea, sunt folosite calculatoarele pentru rezolvarea unor probleme de programare in limbaj de asamblare folosind simulatorul LC-3 si memory (vezi mai jos).</p> <p>Simulatorul LC-3 (for <i>Little Computer 3</i>) – 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.</p> <p>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.</p>
----------------------------------	---

Stabilirea notei finale (procentaje)*)	- răspunsurile la examen/colocviu(evaluare finală)	60
	- teste pe parcursul semestrului	20
	- răspunsurile finale la lucrările practice de laborator	
	- activități gen teme/referate/eseuri/traduceri/proiecte etc.	20
	- teme de control	
	- alte activități(precizați).....	
	- TOTAL	100%

*) Condiții de promovare: minim nota 5 pe toate componentele (Examen / Teste din timpul semestrului / Teme de casa).

<p>Descrieți modalitatea practică de evaluare finală, E/V (de exemplu: lucrare scrisă (descriptive și/sau test grilă și/sau probleme etc.), examinare orală cu bilete, colocviu individual ori în grup, proiect etc)</p> <p>Evaluarea finală va cuprinde lucrare scrisă compusă din (partial) teste grilă și probleme.</p>	
<p>Cerințe minime pentru nota 5</p> <ul style="list-style-type: none"> Cunoașterea, înțelegerea și explicarea noțiunilor elementare specifice științei și ingineriei calculatoarelor. 	<p>Cerințe pentru nota 10</p> <ul style="list-style-type: none"> Cunoașterea, înțelegerea și explicarea noțiunilor specifice științei și ingineriei calculatoarelor.

<ul style="list-style-type: none"> • Interes constant manifestat pentru însușirea disciplinei. • Îndeplinirea condițiilor minime obligatorii în privința temelor de casă, referatelor și a testelor date pe parcursul semestrului. 	<ul style="list-style-type: none"> • Punctaj maxim pentru activitatea din timpul semestrului (seminar) care include temele de casă, referatele și a testele parțiale date. • Punctaj maxim pentru lucrarea scrisă la examenul final. • Utilizarea logică și creativă a noțiunilor disciplinei.
TOTAL ore studiu individual (pe semestru) =4ore x 14 sapt. = 56ore	

Bibliografia	<p>Minimală obligatorie:</p> <ol style="list-style-type: none"> 1. Patt Yale, Patel Sanjay – „<i>Introduction to Computing Systems: from bits & gates to C & beyond</i>”, McGraw-Hill Higher Education, 2001. 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://webspaces.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.). <p>Complementară:</p> <ol style="list-style-type: none"> 8. Zaharia Mihai, Leon Florin – „<i>Limbaajul 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.
	<p>Lista materialelor didactice utilizate în procesul de predare: VideoProiector / RetroProiector, Calculatoare (simulatoare software), Tabla + Creta</p>

Coordonator de Disciplină	Grad didactic, titlul, prenume, numele	Semnătura
	Conf.univ.dr.ing. Adrian FLOREA	