

Numele disciplinei:	Arhitectura calculatoarelor						
Facultatea:	FH		Domeniu:		ISI		
Program de studii	AIA						
Cod plan:	21		Cod disciplină:		774		
Anul de studiu:	1	Forma de examinare: (E-Examen; C- Colocviu; P-Proiect; A/R- Admis/ Respins)		E	Credite ECTS (CR):	E(C)	4
Semestrul:	2					P	0
Categoria disciplinei: (DF - Fundamentală; DD - Domeniu; DS -Specialitate; DC - Complementară; DA - Aprofundare)							DD
Tipul disciplinei: (OB-Obligatorie; OP-Opțională; FC-Facultativă)							OB
Numărul de ore de activitate didactică pe semestru:		Activitate didactică neasistată		44	Activitate didactică asistată parțial		0
Activitate didactică asistată integral:	56	Din care, săptămânal:					
		Curs		Seminar		Laborator	Proiect
		2	0		2	0	

Departament	DIH
Cadru didactic titular:	Conferențiar Virginia Cristiana Săndulescu

Obiectivele disciplinei (descrierea rezultatelor învățării):	
Cunoștințe: Înțelegerea principiilor fundamentale ale calculatoarelor; proiectare hardware; programare la nivel de asamblare. Abilități: Aplicarea cunoștințelor în proiectarea și implementarea hardware; scrierea și interpretarea codului în limbaj de asamblare; proiectarea și simularea componentelor de bază ale unui sistem de calcul Responsabilitate și autonomie: proiectarea și implementarea calculatoarelor; configurarea unui calculator prin selecția componentelor adecvate	
Descrierea cursului:	
1. Curs	1. Concepte fundamentale legate de arhitectura calculatoarelor. Nivelurile conform structurării realizate de A. Tanenbaum. 2. Logică booleană Porțile NAND ca element de bază; Utilizarea unui limbaj de descriere hardware (HDL) pentru realizarea circuitelor elementare (AND, OR, NOT, MUX etc.). 3. Aritmetică booleană Reprezentarea numerelor semnate cu biți și realizarea operațiilor aritmetice prin porți logice. Construcția unei Unități Aritmetico-Logice (ALU). 4. Memorie Funcționarea logicii secvențiale, semnalele de ceas și flip-flop-urile pentru stocarea informațiilor. Construirea unei ierarhii de memorie de la celule de un bit; regiștrii și unități RAM. 5. Limbaj de mașină Definirea setului de instrucțiuni și a arhitecturii abstracte a calculatorului, care transformă programele de nivel înalt în cod binar. Realizarea programelor de asamblare și gestionarea dispozitivelor de intrare/ieșire. 6. Arhitectura de bază a unui calculator. Integrarea componentelor hardware dezvoltate anterior într-un sistem complet. Utilizarea conceptului de program

	stocat și a ciclului fetch-execute pentru a construi un calculator capabil să ruleze programe scrise în limbaj mașină. 7. Asamblare Transformarea instrucțiunilor simbolice în cod binar. Aplicarea tehnicilor de parsare, generare de cod și rezolvare a simbolurilor pentru realizarea unui translator de instrucțiuni. 8. Mașini virtuale I Conceptul de mașină virtuală ca intermediar între programele de nivel înalt și hardware, bazat pe automatul cu stivă. Crearea unui translator pentru comenzile VM (push/pop și operații aritmetice). 9. Mașini virtuale II Extinderea funcționalităților VM prin implementarea comenzilor de control al fluxului, precum sărituri și apeluri de subrutine. Dezvoltarea unui translator complet pentru generarea de cod de asamblare compatibil cu platforma Hack. 10. Limbaje de nivel înalt Introducerea limbajului Jack, un limbaj orientat pe obiecte similar cu Java. Realizarea unui proiect final (de exemplu, un joc) care rulează pe platforma Hack. 11. Compilatoare I Utilizarea gramaticilor context-free și a algoritmilor de parsare recursivă pentru construirea unui analizor sintactic (tokenizer și parser) pentru limbajul Jack. 12. Compilatoare II Extinderea analizorului sintactic într-un compilator complet capabil să transforme programele scrise în Jack în cod VM destinat mașinii virtuale. 13. Nivelul sistemului de operare Implementarea funcționalităților de bază ale unui sistem de operare, incluzând gestionarea memoriei, operații matematice, redarea grafică și manipularea șirurilor de caractere, prin cod scris în Jack pe platforma Hack. 14. Extensii și optimizări Idei pentru extinderea și îmbunătățirea sistemelor dezvoltate: implementarea unui sistem de fișiere, a unui shell de comandă, conectarea la Internet sau realizarea hardware-ului pe FPGA.
2. Seminar/ Laborator/ Proiect/ Practică	1. Structura unui sistem de calcul. Abstractizări. 2. Logică booleană Porțile NAND ca element de bază; Utilizarea unui limbaj de descriere hardware (HDL) pentru realizarea circuitelor elementare (AND, OR, NOT, MUX etc.). 3. Aritmetică booleană Reprezentarea numerelor semnate cu biți și realizarea operațiilor aritmetice prin porți logice. Construcția unei Unități Aritmetico-Logice (ALU). 4. Memorie Funcționarea logicii secvențiale, semnalele de ceas și flip-flop-urile pentru stocarea informațiilor. Construirea unei ierarhii de memorie de la celule de un bit; regiștrii și unități RAM. 5. Limbaj de mașină Definirea setului de instrucțiuni și a arhitecturii abstracte a calculatorului, care transformă programele de nivel înalt în cod binar. Realizarea programelor de asamblare și gestionarea dispozitivelor de intrare/ieșire. 6. Arhitectura de bază a unui calculator. Integrarea componentelor hardware dezvoltate anterior într-un sistem complet. Utilizarea conceptului de program stocat și a ciclului fetch-execute pentru a construi un calculator capabil să ruleze programe scrise în limbaj mașină. 7. Asamblare Transformarea instrucțiunilor simbolice în cod binar. Aplicarea tehnicilor de parsare, generare de cod și rezolvare a simbolurilor pentru realizarea unui translator de instrucțiuni. 8. Mașini virtuale I Conceptul de mașină virtuală ca intermediar între programele de nivel înalt și hardware, bazat pe automatul cu stivă. Crearea unui translator pentru comenzile VM (push/pop și operații aritmetice).

	9. Mașini virtuale II Extinderea funcționalităților VM prin implementarea comenzilor de control al fluxului, precum sărituri și apeluri de subrutine. Dezvoltarea unui translator complet pentru generarea de cod de asamblare compatibil cu platforma Hack. 10. Limbaje de nivel înalt Introducerea limbajului Jack, un limbaj orientat pe obiecte similar cu Java. Realizarea unui proiect final (de exemplu, un joc) care rulează pe platforma Hack. 11. Compilatoare I Utilizarea gramaticilor context-free și a algoritmilor de parsare recursivă pentru construirea unui analizor sintactic (tokenizer și parser) pentru limbajul Jack. 12. Compilatoare II Extinderea analizorului sintactic într-un compilator complet capabil să transforme programele scrise în Jack în cod VM destinat mașinii virtuale. 13. Nivelul sistemului de operare Implementarea funcționalităților de bază ale unui sistem de operare, incluzând gestionarea memoriei, operații matematice, redarea grafică și manipularea șirurilor de caractere, prin cod scris în Jack pe platforma Hack. 14. Extensii și optimizări Idei pentru extinderea și îmbunătățirea sistemelor dezvoltate: implementarea unui sistem de fișiere, a unui shell de comandă, conectarea la Internet sau realizarea hardware-ului pe FPGA.
3. Bibliografie	1. Noam Nisan, Shimon Schocken, The Elements of Computing Systems: Building a Modern Computer from First Principles, 2005, The MIT Press 2. A. S., Tanenbaum, „Organizarea structurală a calculatoarelor”, Ed. Byblos, 2004. 3. Mi, Lu, "Arithmetic and Logic in Computer Systems" Wiley - Interscience, 2004 4. Darren Ashby, Bonnie Baker - "Circuit Design", Editura Elsevier, Newnes 2008

Examinarea:	Ponderea fiecărui criteriu în nota finală
1. Examinarea finală	50%
2. Verificarea cunoștințelor pe parcurs	
Seminar	
Laborator	
Proiect (proiectul nu are notă distinctă)	
3. Evaluări periodice:	
3.1 Examinări scrise / orale	
3.2 Teme, rapoarte, etc.	50%
4. Alte criterii (se vor specifica)	10% (activitatea la orele de curs și aplicații - minim 5 puncte la minim 7 teste de la orele de curs)
Scurtă descriere a procedurii de examinare finală:	
Evaluarea finală se va realiza prin examen tip grilă, asistat de calculator pe platforma https://moodle.utcb.ro	

Estimarea numărului total de ore pe semestru pentru studiu individual			
Tipul de activitate individuală	Nr. ore	Tipul de activitate individuală	Nr. ore
1. Studiu notițe de curs	20	8. Studiu pentru examinarea finală	4
2. Studiu bibliografie obligatorie		9. Ședințe de consultații	4

3. Studiu bibliografie suplimentară		10. Documentare practică pe teren	
4. Pregătire activități specific disciplinei		11. Studiu la bibliotecă adițional	
5. Pregătire teme	16	12. Studiu resurse internet	
6. Studiu pentru evaluări periodice scrise		13. Alte activități (se enumeră)	
7. Studiu pentru evaluări periodice orale		Numărul total de ore:	44

Data:	Decan
01 octombrie 2024	Conf. dr. ing. Alexandru Dimache
	Director de departament:
	Conf. dr. ing. Neculoiu Giorgian
	Titular de disciplină:
	Conferențiar Virginia Cristiana Săndulescu