

FIȘA DISCIPLINEI

Arhitectura sistemelor de calcul

Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai din Cluj-Napoca
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Matematică și Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Matematică-Informatică (în limba engleză)
1.7. Forma de învățământ	Zi

2. Date despre disciplină

2.1. Denumirea disciplinei	Analiza și sinteza circuitelor			Codul disciplinei	MLE5004
2.2. Titularul activităților de curs	Prof. dr. ing. Octavian Creț				
2.3. Titularul activităților de seminar	Prof. dr. ing. Octavian Creț				
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligatoriu	2.8. Tipul disciplinei	Disciplină de domeniu		

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	1 lab 1 sem
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/ laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					21
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					21
Tutoriat (consiliere profesională)					4
Examinări					3
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				69	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	-
4.2. de competențe	-

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tablă și videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Laborator cu calculatoare/laptopuri

6.1. Competențele dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)

Competențe profesionale	
Codul competenței	Competență
CP3	Elaborarea și analiza unor algoritmi pentru rezolvarea problemelor

CP6	Analiza, testarea și utilizarea sistemelor informatice
Competențe transversale	
Codul competenței	Competență
CT1	Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactico-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor de etică profesională.
CT2	Capacitatea de a se integra în medii variate din domeniul învățământului, al cercetării și al economiei.

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1	Absolventul cunoaște și înțelege conceptele, teoriile și metodele de bază ale domeniului Matematică și Informatică și este capabil să le utilizeze în mod adecvat în comunicarea profesională.	<ul style="list-style-type: none"> • Luarea de notițe: rezumarea și înregistrarea informațiilor cheie din cursuri sau texte. • Gândire critică: Analizarea, evaluarea și sintetizarea informațiilor, în loc de a le accepta pur și simplu. • Înțelegerea textului citit: Citirea activă, parcurgerea rapidă, citirea pe diagonală și înțelegerea textelor complexe.
CP2	Absolventul este capabil să proiecteze / implementeze componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii, și să le evalueze caracteristicile funcționale și nefuncționale pe baza unor metrici.	<ul style="list-style-type: none"> • Gestionarea timpului: Stabilirea priorităților, respectarea termenelor limită și organizarea. • Redactare academică: Construirea unor argumente clare, structurate și bazate pe dovezi. • Abilități de studiu: Tehnici de revizuire, strategii de susținere a examenelor și ajutoare pentru memorie.
CP3	Absolventul este capabil să realizeze testarea și evaluarea calitativă a caracteristicilor funcționale și nefuncționale ale sistemelor informatice, pe baza unor criterii specifice	<ul style="list-style-type: none"> • Rezolvarea problemelor: Aplicarea raționamentului logic pentru a rezolva probleme academice.
CP5	Absolventul este capabil să utilizeze instrumente electronice pentru a caracteriza și pentru a evalua performanțele circuitelor electronice.	<ul style="list-style-type: none"> • Competențe digitale: Utilizarea eficientă a bazelor de date, a software-ului și a instrumentelor online. • Comunicare și prezentare: vorbirea în public, prezentarea argumentelor și munca în grup. • Colaborare: Lucrul eficient în echipă și oferirea de critici constructive.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Cunoașterea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, utilizarea sistemelor de reprezentare a informației în sistemele de calcul
2. Însușirea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, a utilizării sistemelor de reprezentare a informației în calculator.
3. Inițiere în programarea în limbaj de asamblare, ceea ce asigură înțelegerea arhitecturii și funcționării unui microprocesor.
4. Înțelegerea impactului arhitecturii procesoarelor 80x86 asupra sistemului de operare.
5. Cunoașterea metodelor de evaluare a performanțelor sistemelor de calcul
6. Cunoașterea metodelor arhitecturale de creștere a performanțelor sistemelor de calcul
Abilități academice specifice (Specific academic skills)
1. Absolventul este capabil să proiecteze și să optimizeze un sistem de calcul la nivel arhitectural

2. Absolventul este capabil să testeze și să evalueze caracteristicile funcționale și nefuncționale ale sistemelor de calcul pe baza unor criterii specifice.

8. Conținuturi



















8.1 Curs	Metode de predare - învățare	Observații
<ol style="list-style-type: none"> 1. Introducere. Sisteme de numerație și coduri 2. Reprezentarea numerelor în calculatoare 3. Reprezentări binare și ordinea pozițiilor 4. Organizarea unui sistem de calcul 5. Unitatea centrală de procesare – UCP 6. Memoria 7. Perifericele 8. Arhitectura microprocesoarelor x86 (IA-32) 9. Arhitecturile RISC și CISC. Evaluarea performanței sistemelor informatice. Legea lui Amdahl 10. Noțiuni de bază despre limbajul de asamblare (pe 32 de biți) 11. Instrucțiuni în limbajul de asamblare 12. Operații în limbajul de asamblare 13. Programarea multi-modul în limbajul de asamblare 14. Recapitulare. Întrebări și răspunsuri finale 	Prezentări, discuții	N/A
<p>Bibliografie</p> <ol style="list-style-type: none"> 1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014. 2. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005. 3. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003. (http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html) 4. Irvine, K.R., 2015. Assembly language for x86 processors. 5. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer. 6. Carter, P.A., 2004. PC Assembly Language. Github: (http://pacman128.github.io/static/pcasm-book.pdf) 7. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press. 8. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer’s manual. Volume 3B: System programming Guide, Part, 2, p.11. (http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf) 9. Bartlett, Jonathan. "Nasm (Intel) Assembly Language Syntax." In Learn to Program with Assembly: Foundational Learning for New Programmers, pp. 271-273. Berkeley, CA: Apress, 2021. 		
8.2 Seminar / laborator	Metode de predare - învățare	Observații
<ol style="list-style-type: none"> 1. Conversii și operații în diferite baze de numerație. Familiarizarea cu instrumentele de lucru pentru laborator 2. Instrucțiuni aritmetice (1). Declararea variabilelor / constantelor 3. Instrucțiuni aritmetice (2). Instrucțiuni de conversie cu semn 4. Operații pe biți. Instrucțiuni de comparare, salt condiționat și de ciclare. Operații pe șiruri. 5. Operații pe șiruri de bytes/words/doublewords/quadwords 6. Apeluri de funcții sistem. Programare multi-modul (asm+asm) 7. Test de laborator 	<p>Lucrări practice pe panouri didactice de testare, plăci FPGA, software specializat (simulatoare), prezentări la tablă, explicații suplimentare și discuții</p>	N/A
<p>Bibliografie</p> <ol style="list-style-type: none"> 1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014. 2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005. 3. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005. 4. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003. (http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html) 		

5. Irvine, K.R., 2015. Assembly language for x86 processors.
6. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer.
7. Carter, P.A., 2004. PC Assembly Language. Github: (<http://pacman128.github.io/static/pcasm-book.pdf>)
8. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press.
9. Guide, P., 2011. Intel® 64 and IA-32 architectures software developer's manual. Volume 3B: System programming Guide, Part, 2, p.11. (<http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf>)
10. Zhirkov, Igor, and Igor Zhirkov. "Assembly Language." Low-Level Programming: C, Assembly, and Program Execution on Intel® 64 Architecture, pp 17-38, 2017

9. Evaluare

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Abilități de rezolvare a problemelor. Prezență, (inter)activitate	Examen scris	70%
9.5 Seminar/laborator	Abilități de rezolvare a problemelor	Examen față în față și/sau scris sau prin intermediul platformei TEAMS, dacă este necesar	30%
9.6 Standard minim de promovare			
<ul style="list-style-type: none"> • Condiții pentru participarea la examenul scris final: nota la lucrările practice ≥ 5 • Condiții pentru promovarea examenului: nota la lucrările practice ≥ 5 ȘI nota la examenul scris ≥ 5; • Modelarea și rezolvarea problemelor tipice de proiectare logică folosind aparatul formal specific domeniului. 			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)

	Eticheta generală pentru Dezvoltare durabilă							
								
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
								Nu se aplică nici o etichetă
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Data completării:

09.05.2026

Semnătura titularului de curs

Oret

.....

Semnătura titularului de seminar

Oret

.....

Data avizării în departament:

...

Semnătura directorului de departament

.....