

D.O.007 ARHITECTURI DE CALCULATOARE
1. Date despre disciplină/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Ingineria Software și Automatică				
Ciclul de studii	Ciclul I				
Programul de studii	0613.2 Ingineria software				
Anul de studii	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
Anul II (<i>învățământ cu frecvență</i>)	4	E	D-disciplină de domeniu profesional	O-disciplină obligatorie	4

2. Timpul total estimat

Total ore în planul de învățământ	Din care				
	Ore auditoriale		Lucrul individual		
	Curs	Lucrări practice	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
120 (cu frecvență)	30	30	0	30	30

3. Precondiții de acces la disciplină/modul

Conform planului de învățământ	Programarea calculatoarelor, Structuri de date și algoritmi, Analiza și modelarea sistemelor, Matematica aplicată
Conform competențelor	Cunoștințe generale privind structura calculatoarelor, structuri de date și capacitatea de analiză și descompunere a problemelor în pași logici, structurarea algoritmică a programelor, modurile de funcționare a dispozitivelor numerice. Familiarizarea cu concepte de matematică aplicată, cum ar fi sistemele de numerație binare și hexazecimale, operații logice de bază și algebra booleană precum și concepte fundamentale de procesare a datelor.

4. Condiții de desfășurare a procesului educațional pentru

Curs	Pentru a îmbunătăți predarea conținutului teoretic în clasă, echipamentele esențiale includ un proiector, o conexiune la internet fiabilă, boxe și fie un flipchart, fie o tablă pentru scris. În plus, pentru lecțiile practice, pot fi necesare markere/markere pentru tablă albă etc.
Lucrări practice	Studentii vor perfecta rapoarte conform condițiilor impuse de indicațiile metodice. Termenul maximal de predare a lucrării de laborator – 2 săptămâni după finalizarea acesteia. Pentru predarea cu întârziere a lucrării, aceasta se depuncea cu 1pct./s. de întârziere.

5. Competențe specifice acumulate

Competențe profesionale	C5.1 Identificarea și definirea de componente hardware, software și de comunicații specifice arhitecturii calculatoarelor. <ul style="list-style-type: none"> ✓ Identificarea structurii de componente a sistemelor de calcul. ✓ Identificarea structurii și organizării memoriei, magistralelor, interfețelor calculatorului.
--------------------------------	---

	<p>C5.2 Explicarea interacțiunii și funcționării componentelor arhitecturale hardware și software ale sistemului de calcul.</p> <ul style="list-style-type: none"> ✓ Explicarea funcționării și interacțiunii a componentelor pe placa de sistem a calculatorului. ✓ Explicarea structurii interne, la nivel de module, ale microprocesoarelor și interacțiunea lor. <p>C5.3 Reprezentarea și procesarea corectă a datelor, crearea circuitelor logice și programarea în limbaj Assembler</p> <ul style="list-style-type: none"> ✓ Familiarizarea cu diferite sisteme de numerație și transformarea numerelor între acestea. ✓ Aplicarea codurilor directe, inverse și complementare în reprezentarea datelor. ✓ Capacitatea de a analiza și proiecta circuite logice prin utilizarea diagramei Karnaugh și a porților logice. ✓ Aplicarea conceptelor de bază pentru proiectarea circuitelor complexe utilizând multiplexoare. ✓ Înțelegerea limbajului Assembler și utilizarea NASM pentru crearea și gestionarea programelor complexe. ✓ Dezvoltarea de proceduri și meniuri funcționale în Assembler. <p>C5.4 Gestionarea componentelor hardware</p> <ul style="list-style-type: none"> ✓ Cunoașterea microprocesoarelor, arhitecturii setului de instrucțiuni și gestionarea memoriei (inclusiv memoria virtuală). ✓ Înțelegerea structurii plăcilor de sistem și utilizarea magistralelor și interfețelor pentru comunicație.
Competențe transversale	<p>Dezvoltarea abilității de a lua decizii tehnice bazate pe criterii logice și eficiente. Prezentarea clară și coerentă a ideilor și rezultatelor, atât în formă scrisă, cât și orală. Dezvoltarea unei atitudini proactive în dobândirea cunoștințelor și competențelor relevante. Organizarea eficientă a resurselor și a timpului pentru atingerea obiectivelor stabilite.</p>

6. Obiectivele disciplinei/modulului

Obiectivul general	<p>Inițierea în domeniul software (assembler) și hardware a calculatoarelor prin explorarea principiilor de bază, a sistemelor de numerație, a circuitelor logice și a limbajului Assembler.</p>
Obiectivele specifice	<ul style="list-style-type: none"> ✓ Înțelegerea componentelor fundamentale ale unui sistem de calcul (unitatea centrală de procesare, memoria, dispozitivele de intrare/ieșire) și a modului în care acestea interacționează. ✓ Dobândirea cunoștințelor despre structura și funcționarea unităților aritmetico-logice, a registrelor și a magistralelor. ✓ Familiarizarea cu conceptele de organizare și management al memoriei, incluzând memoria cache și memoria virtuală. ✓ Înțelegerea funcționării microprocesoarelor și a gestionării memoriei, inclusiv memoria virtuală. ✓ Explorarea structurii plăcilor de sistem, magistralelor și interfețelor utilizate în calculatoare. ✓ Stăpânirea conceptelor de conversie între sisteme de numerație și utilizarea codurilor binare pentru reprezentarea datelor. ✓ Familiarizarea cu principiile de funcționare ale circuitelor logice, optimizarea acestora prin diagrama Karnaugh și utilizarea porților logice.

	<ul style="list-style-type: none"> ✓ Dezvoltarea abilității de a scrie și optimiza coduri în Assembler NASM, utilizând definiții de date și structuri de instrucțiuni. ✓ Dezvoltarea și implementarea circuitelor logice complexe utilizând software precum Logisim. ✓ Crearea programelor și procedurilor în Assembler NASM, inclusiv realizarea unui meniu funcțional.
--	---

7. Conținutul disciplinei/modulului

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
Tematica cursurilor	
T1. Introducere în arhitectura calculatoarelor: Istoricul și evoluția arhitecturilor de calcul. Rolul arhitecturii în proiectarea și dezvoltarea software-ului.	2
T2. Sisteme de numerație. Reprezentarea datelor	2
T3. Bazele logice ale dispozitivelor numerice. Circuitelor logice și diagrama Karnaugh	4
T4. Structura și clasificarea generală a calculatoarelor. Clasificarea lui Flynn	2
T5. Definierea datelor în Assembler. NASM	2
T6. Arhitectura setului de instrucțiuni	4
T7. Microprocesoarele	6
T8. Gestionarea memoriei. Memoria virtuală	4
T9. Plăcile de sistem. Structura. Magistrale și interfețe	4
Total curs:	30
Tematica practicilor	
P1. Prezentarea calculatorului și a principiilor de funcționare.	2
P2. Conversia numerelor dintr-un sistem de numerație în altul. Codurile direct, invers și complementar	2
P3. Diagrama Karnaugh și circuitele logice	4
P4. Introducere în Logisim și exerciții de bază	4
P5. Porțile logice. Multiplexoare.	4
P6. Limbajul Assembler. NASM. Introducere.	2
P7. Crearea programelor complexe în Assembler NASM	4
P8. Crearea procedurilor în Assembler NASM	4
P9. Crearea unui meniu cu proceduri în Assembler NASM	4
Total practice:	30

8. Referințe bibliografice

Principale	<p>Materiale didactice ale cursului plasate pe platforma ELSE:</p> <ol style="list-style-type: none"> 1. https://else.fcim.utm.md/course/view.php?id=2293 <p>Resurse soft:</p> <ol style="list-style-type: none"> 1. Logisim http://www.cburch.com/logisim/docs/2.7/en/html/guide/index.html 2. NASM https://cs.lmu.edu/~ray/notes/nasmtutorial/ <p>Cărți:</p> <ol style="list-style-type: none"> 1. Fundamentals of Digital Logic with Verilog Design Stephen Brown and Zvonko Vranesic. Versiunea online: https://students.aiu.edu/submissions/profiles/resources/onlineBook/d6m3G9_Fundamentals_of_Digital_Logic_with_Verilog_Design.pdf
-------------------	--

	<p>2. D. Kusswurm, Modern x86 Assembly Language Programming: Covers x86 64-bit, AVX, AVX2 and AVX-512, 2018 , Geneva, IL, USA, ISBN-13: 978-1-4842-4062-5 Versiunea online: https://dokumen.pub/modern-x86-assembly-language-programming-covers-x86-64-bit-avx-avx2-and-avx-512-2nbsped-1484240626-9781484240625.html</p> <p>3. A. Tanenbaum, Structured Computer Organization, 6th ed, Pearson Education Limited, 2013, ISBN-13: 978-0132916523 Versiunea online: https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20-%20Structured%20Computer%20Organization.pdf</p> <p>4. Kip Irvine, Assembly language for x86 processors, 7ed , Florida International University, Computing and Information Sciences, Pearson Education, 2015, ISBN-13: 978-0-13-376940-1 Versiunea online: https://broman.dev/download/Assembly%20Language%20for%20x86%20Processors%207th%20Edition.pdf</p>
Suplimentare	<p>1. Randal Bryant Computer, David O'Hallaron, Systems: A Programmer's Perspective 3rd Edition, Publisher: Pearson; 3rd edition (March 2, 2015), Language: English, Hardcover: 1128 pages, ISBN-10: 013409266X, ISBN-13: 978-0134092669 http://54.186.36.238/Computer%20Systems%20-%20A%20Programmer%27s%20Persp.%202nd%20ed.%20-%20R.%20Bryant%2C%20D.%20O%27Hallaron%20%28Pearson%2C%202010%29%20BBS.pdf</p> <p>2. Intel 64 IA-32 Architectures Software Developer Manuals. Versiunea online: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html</p>

9. Utilizarea IA generativă

Permișiunea de utilizare	<p>Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli:</p> <ul style="list-style-type: none"> • IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice. • Orice utilizare a IA generative trebuie să fie declarată în secțiunea de apendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea acestui instrument/serviciu, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."
Restricții de utilizare	<p>Studenții nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate.</p> <ul style="list-style-type: none"> • Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară. • Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.

10. Evaluare

Periodică		Curentă	Studiu individual	Proiect/teză	Examen
EP 1	EP 2				
Învățământ cu frecvență					
15%	15%	15%	15%		40%

Standard minim de performanță

Prezența și activitatea la prelegeri și lucrări de laborator;

Obținerea notei minime de „5” la fiecare dintre evaluări periodice și lucrări de laborator;

Demonstrarea în lucrarea de examinare finală a cunoașterii conținutului cursului.

11. Criterii de evaluare

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Învățământ cu frecvență				
Evaluare periodică I	Conținut teoretic, teme 1-4	Test pe MOODLE	100%	15%
Evaluare periodică II	Conținut teoretic, teme 5-9	Test pe MOODLE	100%	15%
Evaluare curentă	Activitatea practică	Suștinerea cu succes (notă minimă „5”), a tuturor rapoartelor de laborator	100%	15%
Studiul individual	Activitatea practică	Finalizarea și suștinerea proiectului în cadrul orei, proiectul se încarcă pe MOODLE.	100%	15%
Evaluarea finală	Conținut teoretic și practic	Test pe MOODLE	100%	40%