



FIȘA DISCIPLINEI ARHITECTURA SISTEMELOR DE CALCUL

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA „OVIDIUS” DIN CONSTANȚA
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Matematică și Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studii	Informatică
1.7 Anul universitar	2025-2026

2. Date despre disciplină

2.1 Denumirea disciplinei	Arhitectura sistemelor de calcul					
2.2 Cod disciplină	Info.1.1.05					
2.3 Titularul activităților de curs	Lect. univ. dr. Anata-Flavia Ionescu					
2.4 Titularul activităților aplicative	Lect. univ. dr. Anata-Flavia Ionescu					
2.5 Anul de studii	1	2.6 Semestrul	1	2.7 Tipul de evaluare	Ex	2.8 Regimul disciplinei */** DF/DOB

* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

** DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

3. Timpul total (ore pe semestru)

3.1 Număr de ore activități directe pe săptămână	4	din care: 3.2 curs	2	3.3 aplicații (L) ***	2
3.4 Total ore activități directe pe semestru	56	din care: 3.5 curs	28	3.6 aplicații	28
3.7 Total ore de studiu individual					69
Distribuția fondului de timp					[ore]
Studiul cărților, manualelor, suportului de curs, notițelor, bibliografie minimală recomandată					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					5
Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri					20
Pregătire pentru prezentări sau verificări					10
Pregătire pentru examinarea finală					10
Alte activități: consultații					4
3.8 Total ore pe semestru	56 + 69 = 125				
3.9 Numărul de credite	5				

*** S - seminar; L - laborator; P - proiect

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de rezultate ale învățării	



5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1. de desfășurare a cursului	Sala de curs disponibilă cu videoproiector
5.2. de desfășurare a laboratorului	Sala de laborator disponibilă cu calculatoare, platformă online pentru susținerea activităților de laborator

6. Obiectivele disciplinei

6.1 Obiectivul general al disciplinei	Utilizarea bazelor teoretice ale informaticii și a modelelor formale.
6.2 Obiectivele specifice	Descrierea formală a arhitecturii unui sistem de calcul. Enunțarea unor principii fundamentale care stau la baza dezvoltării sistemelor de calcul. Enumerarea și descrierea unor modele arhitecturale pentru sisteme de calcul. Clasificarea arhitecturilor sistemelor de calcul. Utilizarea sistemelor de numerație binară și hexazecimal. Efectuarea de conversii între bazele de numerație. Efectuarea de operații în bazele de numerație 2 și 16. Programarea în limbaj de asamblare.

7. Rezultatele învățării

Cunoștințe	Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și ale arhitecturii calculatoarelor.
Aptitudini	Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare
Responsabilitate și autonomie	Studentul/absolventul coordonează echipe tehnice pentru dezvoltarea de aplicații informatice, asumând decizii responsabile legate de optimizarea și integrarea acestora



8. Conținuturi

8.1 Curs	Metode de predare	Număr ore alocate
1. Descrierea generală a sistemului de calcul, din punct de vedere conceptual. 1.1. Definiții: sistem de calcul, resurse 1.2. Modelul von Neumann 1.3. Conceptul de black-box. Conceptul de mașină virtuală. 1.4. Arhitectura stratificată a sistemului de calcul	Metode de predare- invatare interactive; Metode care implică activ studentii în învățare, punându-i în situația de a realiza conexiuni logice, de a produce idei și opinii proprii argumentate Problematizarea; Conversatia; Metodele active Sinteza/ esențializarea informațiilor Invățarea independentă și prin cooperare	3
2. Descrierea componentelor hardware. 2.1. Unitatea centrală de prelucrare (microprocesor) 2.2. Memoria și dispozitivele și unitățile de intrare/ieșire – noțiuni introductive		4
3. Microprocesoare CISC (Studiu de caz: Intel 8086) 3.1. Funcționarea Unității Centrale de Prelucrare. 3.2. Organizarea registrelor și a memoriei 3.3. Setul de instrucțiuni 3.3.1. Instrucțiuni de transfer 3.3.2. Instrucțiuni aritmetice și logice 3.3.3. Instrucțiuni pentru siruri de caractere 3.3.4. Instrucțiuni pentru controlul programului 3.3.5. Instrucțiuni pentru controlul procesorului 3.4. Întreruperi		6
4. Limbajul de asamblare 4.1. Tipuri de date 4.2. Instrumente utilizate în programarea în limbaj de asamblare 4.3. Structura programului sursă. Întreruperi. 4.4. Proceduri, funcții, macro-comenzi		2
5. Memoria și subsistemul de intrare/ieșire 5.1. Memoria internă 5.2. Memoria externă. Subsistemul de intrare/ieșire		4
6. Performanța sistemului de calcul. 6.1. Metrice de performanță 6.2. Paralelizarea execuției ca tehnică de creștere a performanței. Noțiunea de conductă (pipeline)		3
7. Arhitecturi RISC. 7.1. Caracteristici RISC 7.2. Microprocesoare RISC		1
8. Tipuri de paralelism 8.1. Definiție. Fluxuri. Clasificarea lui Flynn. 8.2. Dependente între instrucțiuni și hazarduri 8.3. Reordonarea instrucțiunilor și planificarea dinamică		3



9. Evoluția sistemelor de calcul		
9.1. Provocări actuale la adresa performanței		
9.2. Coprocesoare		2
9.3. Tendințe emergente în calcul și în arhitecturile multi-calculator		

Bibliografie

- [1]. Harris, S., & Harris, D. (2022). *Digital Design and Computer Architecture, RISC-V Edition*. Elsevier.
- [2]. Hennessy, J. L., & Patterson, D. A. (2019). *Computer Architecture: A Quantitative Approach*, 6th edition. Morgan-Kaufmann.
- [3]. Hwang, K., & Jotwani, N. (1993). *Advanced computer architecture: parallelism, scalability, programmability* (Vol. 199). New York: McGraw-Hill.
- [4]. Shen, J. P., & Lipasti, M. H. (2013). *Modern processor design: fundamentals of superscalar processors*. Waveland Press.
- [5]. Stallings, W. (2015). *Computer Organization and Architecture – Designing for Performance*, 10th edition. Pearson.
- [6]. Tanenbaum, A. S. (2016). *Structured computer organization*. Pearson Education India.

8.2 Aplicații (laborator)	Metode de predare	Număr ore alocate
1. Sisteme de numerație: binar, octal, hexazecimal. Codificarea numerelor. Conversii. Operații.	Dialogul;	4
2. Sistemul de operare. Utilizare. Organizarea informației pe disc. Operații de gestiune a informației stocate și a resurselor sistemului.	Problematizarea; Metodele active și interactive;	2
3. Utilizarea instrumentelor pentru calcul tabelar. Studiu de caz – Microsoft Office Excel.	Sinteza/esențializarea informațiilor;	2
4. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Mediul de dezvoltare (editor, asamblor, depanator) și funcții sistem.	Învățarea independentă și prin cooperare.	2
5. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Structura programului sursă. Apeluri sistem pentru operații de I/E.	Exercițiul	4



6. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Moduri de adresare.		2
7. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Utilizarea structurilor elementare de programare (liniară, alternativă, repetitivă).		4
8. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Proceduri și macroinstrucțiuni.		4
9. Programarea în limbaj de asamblare (studiu de caz: Intel 8086). Programarea dispozitivelor de intrare/ieșire: <ul style="list-style-type: none">• Programarea ecranului pentru modul grafic de lucru• Programarea mouse-ului• Programarea ceasului sistemului		4
Bibliografie [1]. Chelai, Ö., Șerban, C. (2016). <i>Bazele programării în limbaj de asamblare a microprocesoarelor Intel x86. Lucrări de laborator</i> . Editura Ovidius University Press, 2016, ISBN 978-973-614-943-6		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Participare activă la ore	Evaluare orală sau prin lucrări scrise la curs, pe parcursul semestrului	10%
	Definirea și explicarea conceptelor fundamentale, aplicarea corectă a regulilor și a formulelor de calcul predate la curs	Examen scris	50%
9.5 Aplicații (laborator)	Implementarea în limbajul de asamblare studiat a unor soluții pentru probleme simple de programare, utilizarea unor funcții pe întreruperile studiate la curs și laborator	Prezentarea temelor de laborator pe parcursul semestrului	20%
	Implementarea unor algoritmi în limbajul de asamblare studiat, integrarea diferitelor întreruperi și funcții într-o soluție complexă	Prezentarea orală a unui proiect final în limbaj de asamblare (evaluare pe parcursul semestrului, în cursul orelor de laborator)	20%

9.6 Standard minim de performanță / Condiții de promovare – nota 5 din 10
Cunoașterea arhitecturilor fundamentale de sisteme de calcul. Identificarea componentelor hardware și a caracteristicilor acestora. Implementarea și documentarea de unități de program în limbaj de asamblare și folosirea eficientă a mediilor de programare.



Observații:

Nota acordată pentru orice criteriu de evaluare va fi un număr cuprins între 1 și 10.

Componentele evaluării pe parcursul semestrului nu pot fi refăcute în sesiune. Această regulă se aplică și la reexaminări.

Nu se acceptă susținerea unei teme, a unui test sau a unui proiect după expirarea termenului prevăzut.

Data completării,

Titular activităților de curs,

Titular aplicații,

Lect. univ. dr. Ionescu Anata-Flavia

Lect. univ. dr. Ionescu Anata-Flavia

15.09.2025

Data avizării în Departament,

Director de Departament,
Conf. univ. dr. Pelican Elena

19.09.2025

Decan,

Conf. univ. dr. Nicola Aurelian