Anexa 1. Fișa disciplinei R40 – F01

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licenţă - Dual
Programul de studii/calificarea	Calculatoare / Inginer

2. Date despre disciplină

Denumirea disciplinei	**PROGRAMAREA IN LIMBAJ DE ASAMBLARE**				
Anul de studiu	II	Semestrul	IV	Tipul de evaluare	Examen
Regimul disciplinei	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoria de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				DOB

3. Timpul total estimat (ore alocate activităţilor didactice)

I a) Număr de ore pe săptămână	3	Curs	1	Seminar		Laborator IIS	2	Proiect IIS		Practică IIS	
						Laborator IM		Proiect IM		Practică IM	
I b) Totalul de ore pe semestru din planul de învățământ	42	Curs	14	Seminar		Laborator	28	Proiect		Practică	

(*IIS – instituție de învățământ superior; IM – învățare prin muncă*)

Distribuția fondului de timp pe semestru	Ore IIS	Ore IM
II.a) Studiu individual	5	47
II.b) Tutoriat (pentru ID)		
III. Examinări	3	
IV. Alte activități (precizați):		

Total ore studiu individual II + III	Ore IIS	8	Ore IM	
Total ore pe semestru (Ib+II+III+IV)	Ore IIS	50	Ore IM	
Numărul de credite	Credite IIS	2	Credite IM	

4. Competenţe specifice acumulate

Competenţe profesionale	CP02. aliniază software-ul la arhitecturile de sistem CP04. creează softuri CP16. utilizează instrumente de inginerie software asistata de calculator CP20. testează hardware
Competenţe transversale	

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare la nivel mașina (programare în limbaj de asamblare), medii de programare (DOS Debug sau Turbo-Assembler (TASM) și Macro-Assembler (MASM)), tehnici de programare,inginerie software și modul lor de aplicare în probleme concrete.	Studentul analizează, elaborează, dezvoltă și testează programe în limbaj de asamblare pentru diferite arhitecturi de calcul: I8086 și ARM.	Studentul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare la nivel mașina (programare în limbaj de asamblare), medii de programare (DOS Debug sau Turbo-Assembler (TASM) și Macro-Assembler (MASM)), tehnici de programare,inginerie software și modul lor de aplicare în probleme concrete.

6. **Obiectivele disciplinei** (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	● Dezvoltarea de aplicaţii utilizând limbajul de asamblare. ● Manifestarea interesului faţă de programarea în limbaj de asamblare. ● Însuşirea aspectelor fundamentale privind dezvoltarea de aplicaţii utilizând limbajul de asamblare, pentru diferite arhitecturi de calcul RISC/CISC cu creşterea interesului faţă de programarea low level.
Obiective specifice	● Prezentarea limbajului de asamblare aferente familiei de procesoare I8086.
	● În cadrul orelor de laborator studenţii sunt familiarizaţi cu diverse metode de programare, utilizând resursele şi particularităţile microprocesoarelor studiate.
	● Se urmăreşte însuşirea principalelor modalităţi de realizare practică a unor programe editate direct în Assembler sub diferite aplicaţii cum ar fi sub comanda DOS Debug sau Turbo-Assembler (TASM) şi Macro-Assembler (MASM).

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observaţii
1. Introducere. 1.1 Arhitectura procesorului I8086. 1.2 Unitatea de executie. 1.3 Unitatea de interfata cu magistrala. 1.4 Registre de uz general. 1.5 Moduri de adresare.	2h	expunerea, prelegerea-dezbatere, demonstraţia	
2. Setul de instructiuni 8086. 2.1 Instructiuni de transfer. 2.2 Instructiuni aritmetice si logice. 2.3 Instructiuni pentru operatii cu siruri de caractere/cuvinte. 2.4 Instructiuni de apel de procedura si de salt. 2.5 Instructiuni pentru controlul procesorului. 2.6 Dezvoltarea programelor in limbaj de asamblare.	4h	expunerea, prelegerea-dezbatere, demonstraţia	
3. Structura programelor. Definirea si initializarea datelor. Operatori. 3.1 Segmentare. Directive pentru definirea segmentelor. 3.2 Directive pentru legarea modulelor. 3.3 Definirea si initializarea datlor. 3.4 Definirea etichetelor. 3.5 Definirea structurilor. 3.6 Operatori in limbajul de asamblare. 3.7 Directive de asamblare conditionata.	2h	expunerea, prelegerea-dezbatere, demonstraţia	
4. Macroinstructiuni. 4.1 Scopul macroinstructiunilor. Definire si expandare. 4.2 Macroinstructiuni cu parametri. 4.3 Macroinstructiuni repetitive. 4.4 Invocarea recursiva. 4.5 Tehnici avansate de utilizare a macroinstructiunilor.	2h	expunerea, prelegerea-dezbatere, demonstraţia	
5. Tehnici de programare in limbaj de asamblare.	2h		

5.1 Decizia simpla si decizia compusa. 5.2 Cicluri cu test la partea superioara si inferioara. 5.3 Selectia. Tabele de salt sau de apel de proceduri. 5.4 Transferul parametrilor catre proceduri. 5.5 Intoarcerea datelor da catre proceduri.		expunerea, prelegerea-dezbatere, demonstraţia	
5.6 Proceduri cu zone de date proprii. 5.7 Proceduri recursive. 5.8 Proceduri cu număr variabil de parametri. 5.9 Tehnici avansate de programare.	1h		
7.3 Implementarea buclelor. 7.4 Implementarea functiilor.	3h		

Bibliografie minimală recomandată

1. Bazele Microprocesoarelor – Liviu Kreindler – Matrix Rom Bucuresti, 1997
2. Familia de Microcontrolere MCS51 - Vasile Gaitan, 1997
3. Athanasiu I, Panoiu Al - Microprocesoarele 8086,286,386 Ed. TEORA 1992
4. Procesoare Intel, Programare în Limbaj de asamblare, ediţia II a, - Vasile Lungu, 2004
5. Programare in limbaj de asamblare, - Gheorghe Musca. Editura Teora. 1998
6. Fundamentals of Embedded Software with the ARM Cortex-M3, Second Edition, Daniel W. Lewis, 2013
7. Modern Arm Assembly Language Programming, Covers Armv8-A 32-bit, 64-bit and SIMD, Daniel Kusswurm, 2020

Aplicaţii (laborator)	Nr. ore	Metode de predare	Observaţii
● **Laborator** 1. Norme de protecția muncii. Noțiuni introductive ale Programării în Limbaj de Asamblare 2. Reprezentarea datelor. Setul de instrucțiuni 8086. Aritmetica pe 16 biți 3. Aritmetica pe 32 biți. Conversii de lungime a datelor. Instrucțiuni condiționale 4. Tablouri. Operații pe biți 5. Stiva. Subrutine. Macroinstrucțiuni 6. Moduri de transmitere a parametrilor 7. Prelucrarea sirurilor de parametri 8. Structuri. Întreruperi. Evaluarea cunoștințelor 9. Setul de instrucțiuni ARM. 10. Aritmetica pe 16 si 32 de biți 11. Instrucțiuni condiționale ARM 12. Instrucțiuni repetitive ARM 13.Translatare cod C in cod ARM 14. Evaluarea cunoștințelor. Recuperări	2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h	lucrări practice, experimentul	

Bibliografie minimală recomandată

1. Bazele Microprocesoarelor – Liviu Kreindler – Matrix Rom Bucuresti, 1997
2. Familia de Microcontrolere MCS51 - Vasile Gaitan, 1997
3. Athanasiu I, Panoiu Al - Microprocesoarele 8086,286,386 Ed. TEORA 1992
4. Procesoare Intel, Programare în Limbaj de asamblare, ediţia II a, - Vasile Lungu, 2004
5. Programare in limbaj de asamblare, - Gheorghe Musca. Editura Teora. 1998
6. Fundamentals of Embedded Software with the ARM Cortex-M3, Second Edition, Daniel W. Lewis, 2013
7. Modern Arm Assembly Language Programming, Covers Armv8-A 32-bit, 64-bit and SIMD, Daniel Kusswurm, 2020

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	participarea activă în timpul cursurilor.	Evaluare continuă	**10 %**
	Insusirea notiunilor de baza si utilizarea corecta a limbajului de specialitate	Evaluare prin probă finală de tip test practic (oral) și scris	**40%**

		(Moodle) din problemele furnizate la curs.	
Seminar			
Laborator/ Lucrări practice	Finalizarea lucrările practice propuse	evaluare continuă (prin metode orale și probe practice)	**30%**
	Rezolvarea problemelor propuse in temele de acasă	evaluare sumativă (prin prezentarea orală publică, utilizând slide-uri, a rezolvării temei de casă).	**20%**
Proiect			

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
25.09.2025	Conf. dr.ing Cristian Andy TANASE	As. Ing. Dr. Anda TCACIUC

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
25.09.2025	Conf. dr. ing. Ioan UNGUREAN

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Prof. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. dr. ing. Laurentiu- Dan MILICI