FIŞA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul	Calculatoare
Domeniul de studii	Calculatoare şi tehnologia informaţiei
Ciclul de studii	Masterat
Programul de studii/calificarea	Masterat Stiinta si Ingineria Calculatoarelor / Inginer

2. Date despre disciplină

Denumirea disciplinei	**PROCESARE PARALELA**				
Anul de studiu	I	Semestrul	I	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				DOP

3. **Timpul total estimat** (ore alocate activităţilor didactice)

I a) Număr de ore pe săptămână	2	Curs	1	Seminar		Laborator/lucrări practice	1	Proiect	
I b) Totalul de ore pe semestru din planul de învăţământ	28	Curs	14	Seminar		Laborator/lucrări practice	14	Proiect	

Distribuția fondului de timp pe semestru	ore
II. Studiu individual	93
III. Examinări	4
IV. Alte activități (precizați):	

Total ore studiu individual (II+III)	97
Total ore pe semestru (I.b+II+III+IV)	125
Numărul de credite	5

4. Competențe specifice acumulate

Competențe transversale	CT1 -lucreaza în echipe
Competențe profesionale	CP1 – efectuează cercetare ştiinţifică CP2 – concepe designul produsului CP3 – proiectează prototipuri CP8 – stabilește procese de date CP12 – asigură managementul de proiect

5. Rezultatele învățării

Cunoștințe	**Aptitudini**	**Responsabilitate și autonomie**
Studentul / absolventul identifică, conştientizează critic, sumarizează concepte și metode avansate privitoare la algoritmi pentru aplicații de înaltă performanță și modul lor de aplicare în rezolvarea inovativă a unor probleme concrete .	Studentul / absolventul cercetează și proiectează aplicații care folosesc procesarea paralelă a datelor, gestionând probleme de sincronizare, partiționare și comunicare între procese si utilizează algoritmi paraleli avansați.	Studentul / absolventul îşi poate asuma responsabilitatea pentru a contribui la cunoştinţele şi practicile profesionale.

6. **Obiectivele disciplinei** (reieşind din grila competenţelor specifice acumulate)

Obiectivul general al disciplinei	• Obiectivele urmărite sunt: însuşirea de către studenţi a principiilor de bază referitoare arhitecturilor paralele; • Familiarizarea studenţilor cu sistemele de operare pentru multiprocesoare. • Inţelegerea de către studenţi a modului de exploatare a concurenţei la multiprocesoare, detectarea paralelismului în programe şi câteva mecanisme de sincronizare a proceselor. • Stimularea studenţilor pentru orientarea lor profesională către acest domeniu.
Obiective specifice	• Cunoaşterea problematicii calcului paralel;
	• Cunoaşterea noţiunilor fundamentale utilizate în procesarea paralelă
	• Cunoaşterea arhitecturilor pentru paralelismul la nivel de instrucţiune.
	• Cunoaşterea arhitecturilor pentru paralelismul la nivel de date.
	• Cunoaşterea arhitecturilor pentru paralelismul la nivel de proces.
	• Explicarea noţiunilor fundamentale utilizate în procesarea paralelă.
	• Explicarea arhitecturilor folosite în paralelismul la nivel de instrucţiune.
	• Explicarea arhitecturilor folosite în paralelismul la nivel de date.
	• Explicarea arhitecturilor folosite în paralelismul la nivel de proces.

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observaţii
1. Arhitectura ZYNQ (SoC-uri FPGA cu procesor pentru DSP și AI) 1.1. Vedere de ansamblu: PS + PL 1.2. Sistemul de Procesare (PS) 1.3. Programarea Logică (PL) 1.4. Interconectarea PS–PL: AXI 1.5. Tact, I/O și boot 1.6. ZYNQ pentru DSP 1.7. ZYNQ pentru AI 1.8. Flux de proiectare și comparație 1.9. Următoarea generație: ACAP (Versal)	5h	expunerea, prelegerea-dezbatere, demonstraţia	
2. Kitul PYNQ-Z2 (Python + FPGA pentru DSP și IA practice) 2.1. Kitul PYNQ-Z2 (descriere) 2.2. Framework-ul PYNQ 2.3. Python ↔ hardware 2.4. DSP & IA pe PYNQ-Z2 2.5. Predare & primii pași	4h	expunerea, prelegerea-dezbatere, demonstraţia	
3. Lanțul de proiectare pentru PYNQ (De la un algoritm în C/C++, prin Vivado, până în Jupyter) 3.1. Panorama fluxului 3.2. Etapa 1: Vivado / Vitis HLS (sinteză de nivel înalt) 3.3. Etapa 2: Vivado (schematic / block design) 3.4. Etapa 3: Jupyter / PYNQ 3.5. Economia iterării 3.6. O rețetă generică	5h	expunerea, prelegerea-dezbatere, demonstraţia	

Bibliografie minimală recomandată

1. BALBO G. , VANNESCHI M. – General Purpose Parallel Computrs, Edizioni ETS, 1995
2. COSNARD M. TRYSTRAM D. – Parralel Algorithms and Architectures, International Thomson Computer Press, 1995
3. CULLER D. SINGH J. Parallel Computer Architecture, Morgan Kaufmann Publishers, 1998
4. MOLDOVAN D. Parallel Processing: From Application to Systems, Morgan Kaufmann Publishers, 1995
5. TABAK D. Multiprocessors , Prentice-Hall International 1990
6. TANENBAUM A. – Structured Computer Organization, Prentice-Hall International 1999
7. Programare concurentă pe platforme Unix, Windows, Java - Florian Mircea Boian, Corina Ferdean, s.a Editura Albastră 2002

8. APLICAŢII SOFTWARE DISTRIBUITE Dan Cosma, Stejărel Vereş, Adrian Petru Mierluţiu, Editura de Vest, 2003
9. Parallel Processing, 1980 to 2020 by Robert Kuhn, David Padua, 2020

Aplicaţii (Seminar/laborator/proiect)	Nr. ore	Metode de predare	Observaţii
Prin intermediul acestor lucrări de laborator, studenții parcurg un flux complet de Hardware/Software Co-Design pe platforme heterogene de tip System-on-Chip, învățând să transforme algoritmi din C/C++ în acceleratoare hardware dedicate prin High-Level Synthesis (Vivado HLS). Aceștia deprind abilitatea de a asambla arhitecturi complexe în Vivado, integrând procesorul ARM cu module de memorie DMA, magistrale AXI și IP-uri speciale pentru procesarea semnalelor (filtre FIR și calcul spectral FFT) sau procesare de imagine (detecția de muchii Sobel). De asemenea, studenții stăpânesc interfațarea cu placa reconfigurabilă PYNQ-Z2, învățând să controleze acceleratoarele hardware direct din Python prin mediul Jupyter Notebook. Nu în ultimul rând, aceștia înțeleg cum să depășească limitele de resurse ale plăcii FPGA prin tehnici de tiling aplicate pe cadre video de rezoluție mare, dobândind o viziune practică end-to-end asupra proiectării și optimizării sistemelor digitale moderne.			
1. Partea 1 — Vivado HLS (Proiectarea și verificarea nucleului FIR, pas cu pas) • Ce este un filtru FIR? • Unde se încadrează în ansamblul proiectului • Crearea proiectului • Nucleul • Simularea C • Sinteza și exportul ca IP	1h	lucrări practice, experimentul	
2. Partea 2 — Design-ul de Bloc în Vivado (De la un IP din HLS la un overlay hardware complet) • Unde se încadrează în ansamblul proiectului • Proiect nou • Asamblarea blocurilor IP • Conectarea blocurilor • Validare, generare, finalizare	2h		
3. Partea 3 — Python pe PYNQ-Z2 (Rularea overlay-ului real, pas cu pas) • Unde se încadrează în ansamblul proiectului • Transferul overlay-ului pe placă • Notebook-ul • Rezultatul real	1h		
4. Partea 4 — Un bloc FFT în schema Vivado (Spectrul semnalului calculat de FPGA, afișat în Python) • Ideea: FFT ca IP hardware • Configurarea IP-ului FFT • Schema completă: FIR + FFT • Codul Python • Rezultatul real	1h		
5. Partea 5 — Filtre diferite, văzute prin FFT (Un semnal cu mai multe frecvente, filtrat și analizat spectral pe FPGA) • Semnalul și filtrele • Codul Python • Rezultatul real	1h		
6. Filtrul Sobel pe PYNQ-Z2 — Partea 1 (Proiectarea și verificarea nucleului de delectație a marginilor, în Vivado HLS) • Bazele algoritmului Sobel • Crearea proiectului • Nucleul • Simularea C • Sinteza și exportul ca IP	2h		

7. Filtrul Sobel pe PYNQ-Z2 — Partea 2 (Design-ul de bloc în Vivado: de la IP-ul HLS la un bitstream complet) • Proiect nou • Procesorul și automatizarea • Asamblarea blocurilor IP • Conectarea blocurilor • Validare, generare, finalizare	2h		
8. Partea 3 — Python pe PYNQ-Z2 (Rularea overlay-ului Sobel real, pas cu pas) • Transferul overlay-ului pe placă • Notebook-ul • Rezultatul real	2h		
9. Partea 4 — Sobel pe rezoluție completă, prin tiling (Un accelerator fix de 64×64 procesează un video 640×360, mini-cadru cu mini-cadru, pe hardware real) • Ideea • Geometria tiling-ului • Codul Python • Rezultatul real • Alternativa: hardware nativ	2h		

Bibliografie minimală recomandată

1. BALBO G. , VANNESCHI M. – General Purpose Parallel Computrs, Edizioni ETS, 1995
2. COSNARD M. TRYSTRAM D. – Parralel Algorithms and Architectures, International Thomson Computer Press, 1995
3. CULLER D. SINGH J. Parallel Computer Architecture, Morgan Kaufmann Publishers, 1998
4. MOLDOVAN D. Parallel Processing: From Application to Systems, Morgan Kaufmann Publishers, 1995
5. TABAK D. Multiprocessors , Prentice-Hall International 1990
6. TANENBAUM A. – Structured Computer Organization, Prentice-Hall International 1999
7. Programare concurentă pe platforme Unix, Windows, Java - Florian Mircea Boian, Corina Ferdean, s.a Editura Albastră 2002
8. APLICAŢII SOFTWARE DISTRIBUITE Dan Cosma, Stejărel Vereş, Adrian Petru Mierluţiu, Editura de Vest, 2003
9. Parallel Processing, 1980 to 2020 by Robert Kuhn, David Padua, 2020

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Nota acordată pentru participarea activă în timpul cursurilor.	Evaluare continuă	**10 %**
	Nota acordată la examinarea finală.	Evaluare prin probă finală de tip test practic (oral) și scris din problemele furnizate la curs.	**40%**
Evaluare finală	Media notelor acordate la lucrările practice	evaluare continuă (prin metode orale şi probe practice)	**30%**
	Evaluarea temelor de acasă	evaluare sumativă (prin prezentarea orală publică, utilizând slide-uri, a rezolvării temei de casă).	**20%**

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

Data completării	Grad didactic, nume, prenume,	Grad didactic, nume, prenume,

	semnătura titularului de curs	semnătura titularului de aplicație
23.09.2026	Conf. dr.ing Cristian Andy TANASE	Conf. dr.ing Cristian Andy TANASE

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
24.09.2026	prof. dr. ing. Stefan-Gheorghe PENTIUC

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Prof. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof. dr. ing. Laurentiu- Dan MILICI