



### FIȘA DISCIPLINEI

#### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie Politehnica București
1.2 Facultatea	Electronică, Telecomunicații și Tehnologia Informației
1.3 Departamentul	Dispozitive, Circuite și Arhitecturi Electronice
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclul de studii	Masterat
1.6 Specializarea	Calcul Avansat în Sisteme Embedded

#### 2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Calcul paralel					
(en)		Parallel Computing					
2.2 Titularul activităților de curs		S.I./Lect. Dr. George Valentin STOICA					
2.3 Titularul activităților de seminar / laborator		S.I./Lect. Dr. George Valentin STOICA					
2.4 Anul de studiu	2	2.5 Semestrul	I	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei	DA	2.9 Codul disciplinei	UPB.04.M3.O.26-24	2.10 Tipul de notare	Nota		

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2.00	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56.00	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					65
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					
Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					
Tutorat					0
Examinări					4
Alte activități (dacă există):					0
3.7 Total ore studiu individual	69.00				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

#### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Nu se plica
4.2 de rezultate ale învățării	Nu se plica

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

5.1 Curs	Cursul se va desfășura într-o sală dotată cu videoproiector.
----------	--



5.2 Seminar/ Laborator/Proiect	Laboratorul va avea loc într -o sală dotată cu echipamente specifice, care trebuie să includă: PC -uri, mediu de dezvoltare software, cum ar fi MS Visual Studio, CPU multicore și/sau GPU, cum ar fi NVIDIA GTX
-----------------------------------	--

**6. Obiectiv general** (*Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățați în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului științific abordat, precum și la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învățământ al specializării studiate etc.)*)

Obiectivul cursului este ca studenții să înțeleagă conceptele referitoare la arhitecturile de calcul de performanță utilizate în domeniul tehnologiei informației. Cursul prezintă sintetic limbajele folosite în platformele de procesare a informațiilor precum C/C++/Java/Python/.Net și continuă cu calculul paralel: arhitecturi multicore CPU și GPU. A doua parte a cursului aprofundează arhitectura GPU și mai precis arhitectura CUDA ca platformă de calcul de înaltă performanță.

Aplicațiile de laborator au ca scop evidențierea conceptelor și notiunilor practice specifice calculului și prelucrării informațiilor. Folosind dezvoltarea aplicațiilor CUDA ca metode de experimentare sau folosind tehnologii precum OpenCL, studenții pot experimenta singuri cunoștințele dobândite în timpul cursului. Întregul set de noțiuni dobândite pe parcursul cursului poate fi experimentat în cadrul aplicațiilor de laborator

**7. Competențe** (*Capacitatea dovedită de a utiliza cunoștințe, aptitudini și abilități personale, sociale și/sau metodologice în situații de muncă sau de studiu și pentru dezvoltarea profesională și personală. Reflectă cerințele angajatorilor.*)

<b>Specifice</b>	După finalizarea acestui curs, studenții dobândesc noțiunile de bază și avansate de programare folosind arhitecturi GPU/CUDA. Aceste cunoștințe combinate cu parcurgerea aplicațiilor de laborator oferă posibilitatea de a dezvolta aplicații paralele de procesare a informațiilor. De asemenea, studenții vor fi capabili să dezvolte tehnici și metode de programare, optimizare și eficiență a algoritmilor paraleli.
<b>Transversale (generale)</b>	Îndeplinirea sarcinilor profesionale cu identificarea exactă a obiectivelor de atins, a potențialilor factori de risc, a resurselor disponibile, a aspectelor economico-financiare, a condițiilor de îndeplinire a acestora, a etapelor de lucru, a timpului de lucru și a termenelor aferente.

**8. Rezultatele învățării** (*Sunt enunțuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învățării reflectă realizările studentului și mai puțin intențiile profesorului. Rezultatele învățării informează studenții despre ceea ce se așteaptă de la ei din punct de vedere al performanței, pentru a obține notele și creditele dorite. Sunt definite în termeni concreți, folosind verbe similare exemplurilor de mai jos și indică ceea ce se va urmări prin evaluare. Rezultatele învățării vor fi astfel redactate încât să fie evidențiată clar relația față de competențele definite la punctul 7.)*



<b>Cunoștințe</b>	<p><i>Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice.</i></p> <p>După finalizarea acestui curs, studenții dobândesc noțiunile de bază și avansate de programare folosind arhitecturi paralele, cum ar fi arhitecturi CPU multicore și GPU/CUDA. Aceste cunoștințe combinate cu parcurgerea aplicațiilor practice oferă posibilitatea de a dezvolta aplicații eficiente de procesare a informațiilor.</p> <p>De asemenea, studenții vor fi capabili să dezvolte tehnici și metode de programare, optimizare și eficiență a algoritmilor de procesare a informațiilor.</p>
<b>Aptitudini</b>	<p><i>Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente).</i></p> <p>Selectați și grupați informații relevante într-un context dat.</p>
<b>Responsabilitate și autonomie</b>	<p><i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale. Demonstrează receptivitate la noi contexte de învățare.</i></p>

**9. Metode de predare** (Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămăneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)

În cadrul acestei discipline, ținând cont atât de orele de curs, cât și de cele de aplicații, se vor utiliza atât metode de predare expositive (prelecție, expunere), cât și conversațional-interactiv, bazate pe modele de învățare prin descoperire facilitate de explorarea directă și indirectă a realității (experiment, demonstrație, modelare), dar și pe metode bazate pe acțiune, precum exerciții, activități practice și rezolvarea de probleme.

În activitatea didactică se vor folosi prelegeri, pe baza unor prezentări Power Point. Prezentările folosesc imagini și diagrame astfel încât informațiile prezentate să fie ușor de înțeles și asimilat.

Prezentările pot fi realizate și prin intermediul canalelor de comunicare online, pe platforma Teams.

Vor fi luate în considerare abilitățile de ascultare activă și de comunicare asertivă, precum și mecanismele de construcție a feedback-ului. Abilitățile de lucru în echipă vor fi exersate pentru a rezolva diferite sarcini de învățare.

## 10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore
1	Introducere în calculul paralel 1.1. Paralelism motivant: aplicații intensive de calcul 1.2. Introducerea paralelismului în arhitecturile computerelor	2



2	Arhitecturi GPU 2.1. Arhitecturi GPU, GPGPU 2.2. Arhitectura CUDA	8
3	Arhitectura CUDA 3.1. Istorie, evoluție 3.2. Descriere 3.3. Concepte specifice CUDA 3.4. Modele de programare CUDA 3.5. Modele de programare. Paralelism la nivel de date. Tipuri de memorie CUDA. 3.6. Aplicații. Analiza și optimizare. Considerații de performanță.	10
4	Biblioteci bazate pe CUDA 4.1. Prezentarea bibliotecilor bazate pe arhitectura CUDA 4.2. cuBLAS, OpenACC, OpenCL, DNN	4
5	Aplicații 5.1. Aplicații de prelucrare a informațiilor cu implementări GPU/CUDA: procesare imagini, inteligență artificială, viziune computerizată, grafică 3D	4
	<b>Total:</b>	28

**Bibliografie:**

David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013

nVidia CUDA: [www.developer.nvidia.com](http://www.developer.nvidia.com)

Matt Pharr, Randima Fernando : GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pearson Education 2005,

[http://download.nvidia.com/developer/GPU\\_Gems\\_2/CD/Index.html](http://download.nvidia.com/developer/GPU_Gems_2/CD/Index.html)

Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010,

[http://www.mat.unimi.it/users/sansotte/cuda/CUDA\\_by\\_Example.pdf](http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf)

**LABORATOR**

Nr. crt.	Conținutul	Nr. ore
1	Biblioteci GPU/CUDA	2
2	Dezvoltarea unei aplicații CUDA	2
3	Dezvoltarea unei aplicații CUDA	2
4	Analiza/profilarea aplicațiilor CUDA	2
5	Optimizare	2
6	Aplicații: procesare imagini	2
7	Aplicații: prelucrarea datelor	2
	<b>Total:</b>	



**Bibliografie:**

CUDA C Programming Guide, <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>  
CUDA C Best Practices Guide, <http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html>  
David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013  
Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010,  
[http://www.mat.unimi.it/users/sansotte/cuda/CUDA\\_by\\_Example.pdf](http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf)

**11. Evaluare**

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	<ul style="list-style-type: none"><li>- cunoașterea termenilor teoretici fundamentali;</li><li>- cunostinte tehnici si arhitecturi de programare;</li><li>- cunoașterea platformei GPU/CUDA;</li><li>- cunoașterea unei biblioteci bazata pe GPU/CUDA</li></ul>	Teste de verificare scrise pe parcursul semestrului, susținute la date fixe la finalul cursului; subiectele acoperă întreaga materie, realizând o sinteză între cursul teoretic comparativ al materiei și explicarea prin exerciții a tehnicilor specifice programării platformelor GPU.	50%
11.5 Seminar/laborator/proiect	<ul style="list-style-type: none"><li>- cunoașterea modului de proiectare a unui algoritm de analiză a imaginii pentru a rezolva o problemă dată;</li><li>- cunoașterea modului de transpunere a unui algoritm în cod;</li><li>- demonstrarea funcționării unui algoritm implementat.</li></ul>	Proiectul final de laborator, cuprinzând o componentă teoretică și o componentă practică. Componenta teoretică se verifică prin întocmirea unui raport; componenta practică se evaluează prin verificarea modului de rezolvare (implementare, testare, operare) de către elev a unor probleme practice.	50%
11.6 Condiții de promovare			
Proiectarea, implementarea și demonstrarea de soluții practice pentru procesarea imaginilor, viziunea computerizată, inteligența artificială folosind biblioteci accelerate CPU sau GPU/CUDA. Obținerea a cel puțin 50% din punctajul total.			

**12. Coroborarea conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)**



**Universitatea Națională de Știință și Tehnologie Politehnica București**

**Facultatea de Electronică, Telecomunicații și  
Tehnologia Informației**



Prelucrarea informației necesită cunoașterea arhitecturilor disponibile în prezent pe piață, a limbajelor de programare, a structurilor de date necesare precum și a bibliotecilor existente adaptate diverselor domenii precum procesarea imaginilor, viziunea computerizată, inteligența artificială, grafica 3D, calculele matematice, algebra. Înțelegerea acestor aspecte împreună cu particularitățile fiecărei arhitecturi hardware creează premisele pentru implementări eficiente, optimizate și adaptate la cerințele de calcul din ce în ce mai mari odată cu complexitatea tot mai mare a aplicațiilor și a datelor acestora. Procesarea rapidă a informațiilor necesită utilizarea unor platforme adaptate, indiferent dacă sunt folosite arhitecturi cloud, arhitecturi GPU sau platforme mobile.

Data completării

Titular de curs

Titular(i) de aplicații

09.09.2022

Sl. Dr. Ing. George Valentin  
STOICA

Sl. Dr. Ing. George Valentin  
STOICA

Data avizării în departament

Director de departament

31.10.2024

Prof. Dr. Claudiu DAN

Data aprobării în Consiliul  
Facultății

Decan

01.11.2024

Prof. Dr. Mihnea Udrea