FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea[1] / Departamentul[2]	Robotică, Inteligență Artificială și Sisteme Autonome / Calculatoare și Tehnologia Informației
1.3 Domeniul de studii (denumire/cod [3])	Informatică / 40
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Informatică / 011 / Informatică

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Fundamentele calculatoarelor / DF				
2.1b Denumirea disciplinei în limba engleză			Fundamentals of Computing				
2.2 Titularul activităţilor de curs			Ș.l.dr.ing. Flavius-Gabriel OPRIȚOIU				
2.3 Titularul activităţilor aplicative[5]			Ș.l.dr.ing. Flavius-Gabriel OPRIȚOIU				
2.4 Anul de studii[6]	1	**2.5** Semestrul	1	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DI

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate) [8]

3.1 Număr de ore asistate integral/săptămână	4, format din:	**3.2** ore curs	2	**3.3** ore seminar/laborator/proiect	2
3.1* Număr total de ore asistate integral/sem.	56, format din:	**3.2*** ore curs	28	**3.3*** ore seminar/laborator/proiect	28
3.4 Număr de ore asistate parțial/săptămână	, format din:	**3.5** ore practică		**3.6** ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/ semestru	, format din:	**3.5*** ore practică		**3.6*** ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/ săptămână	4.9, format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1.6
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1.6
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1.7
3.7* Număr total de ore activități neasistate/ semestru	69, format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			23
		ore studiu individual după manual, suport de curs, bibliografie și notițe			23
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			23
3.8 Total ore/săptămână [9]	8.9				
3.8* Total ore/semestru	125				
3.9 Număr de credite	5				

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	•
4.2 de rezultatele învățării	• Cunoștințe elementare de fizică și matematică

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală de curs spațioasă, Materiale suport: laptop, proiector, tablă
5.2 de desfăşurare a activităţilor practice	• Sală de activități practice cu cel puțin 15-20 de stații de lucru: Mediu de proiectare și simulare; tablă de scris

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C3. Studentul descrie, identifică și explică funcționarea diverselor arhitecturi hardware și software de actualitate, sisteme incorporate si pe platforme reconfigurabile precum si elemente de automatica asociate sistemelor informatice industriale.
Abilități	• A3. Studentul selectează, evaluează și proiectează sisteme informatice care includ partea de hardware, software și elemente de sisteme incorporate.
Responsabilitate și autonomie	• RA3. Studentul construiește etic și responsabil soluții care îmbină componente hardware și software sigure și scalabile, colaborând cu specialiști din domenii conexe.

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Înțelegerea reprezentării numerelor, a algebrei booleene și a funcționării circuitelor logice combinaționale și secvențiale.
- Dobândirea capacității de a aplica aceste noțiuni în calcule cu sume ponderate, precum și de a identifica stările și tranzițiile unui automat finit pentru controlul accesului.

8. Conţinuturi[10]

8.1 Curs	Număr de ore	Metode de predare[11]
1. INTRODUCERE 1.1. Obiectivele disciplinei 1.2. Relații cu alte discipline: arhitectura calculatoarelor, inteligența artificială și securitatea cibernetică	3	Prelegere susţinută în manieră clasică cu multe exemplificări și un stil interogativ de predare cu discutarea amplă a aspectelor mai importante.
2. REPREZENTAREA NUMERELOR ÎN SISTEMELE DE CALCUL 2.1. Clasificarea informației; reprezentarea datelor numerice și a permisiunilor de acces prin biți 2.2. Reprezentarea numerelor în virgulă fixă 2.3. Reprezentarea numerelor în virgulă flotantă 2.4. Cuantizarea valorilor numerice: reprezentarea cu un număr limitat de biți 2.5. Precizie, rotunjire și depășirea domeniului de reprezentare; efecte asupra calculelor și verificării limitelor	6	
3. ELEMENTE FUNDAMENTALE DE ALGEBRĂ BOOLEANĂ 3.1. Funcții fundamentale ale algebrei booleene 3.2. Legile și postulatele algebrei booleene 3.3. Modelarea prin funcții booleene a unei reguli de acces: utilizator autentificat și permisiune acordată	5	
4. SCHEME LOGICE COMBINAȚIONALE 4.1. Etapele sintezei schemelor logice combinaționale 4.2. Metode de soluționare ale sintezei schemelor logice combinaționale 4.3. Metode de simplificare a schemelor logice combinaționale 4.4. Proiectarea tehnologică a schemelor logice combinaționale	8	

4.5. Codificatoare și decodificatoare 4.6. Multiplexoare și demultiplexoare 4.7. Unități aritmetice și logice; calculul unei sume ponderate, ca exemplu elementar din inteligența artificială 4.8. Comparatoare logice; compararea cu praguri și verificarea încadrării valorilor într-un interval admis 4.9. Circuite buffer/tri-state 4.10. Analiza hazardurilor la scheme logice combinaționale 4.11. Studiu de caz: circuit combinațional pentru acordarea accesului pe baza autentificării și a permisiunii		
5. SINTEZA SCHEMELOR LOGICE SECVENȚIALE 5.1. Paralelă între schemele logice combinaționale și secvențiale 5.2. Taxonomia schemelor logice secvențiale: automate Mealy și Moore 5.3. Elemente de memorare; stocarea valorilor de intrare, a ponderilor și a rezultatelor intermediare; controlul accesului la datele stocate 5.4. Etapele sintezei schemelor logice secvențiale; definirea stărilor, a tranzițiilor și a comportamentului la resetare 5.5. Studiu de caz: automat finit pentru controlul accesului, cu stările „în așteptare”, „acces permis” și „blocat” și reacție la o cerere de acces nevalidă	6	

Bibliografie[12]

1. G. Monkman: "Digital Electronics", Springer, ediția I, 2025, ISBN 978-3-031-69725-8
2. Mircea Vlăduțiu: “Computer Arithmetic. Algorithms and Hardware Implementations” Springer, 2012, ISBN 978-3-642-18314-0, ISBN 978-3-642-18315-7
3. M. M. Mano, M. D. Ciletti: "Digital Design, Global Edition", Pearson, ediția a VI-a, 2024, ISBN 978-1-292-47263-8
4. D. M. Harris, S. L. Harris: "Digital Design and Computer Architecture, RISC-V Edition", Morgan Kaufmann, ediția I, 2021, ISBN 978-0-12-820064-3
5. R. S. Sedha, L. Thakre: "A Textbook of Digital Electronics", S. Chand Publishing, 2025, ISBN 978-93-6958-020-0
6. D. R. Patrick, S. W. Fardo, V. Chandra, B. W. Fardo: "Electronic Digital System Fundamentals", River Publishers, ediția a II-a, 2023, ISBN 978-87-7022-739-1
7. S. Brown, Z. Vranesic: "Fundamentals of Digital Logic with Verilog Design: 2025 Release ISE", McGraw-Hill Education, 2025, ISBN 978-1-265-07067-0
8. T. L. Floyd: "Digital Fundamentals, Global Edition", Pearson, ediția a XI-a, 2025, ISBN 978-1-292-48526-3

8.2 Activităţi aplicative[13]	Număr de ore	Metode de predare
1. Introducere 1.1. Obiectivele disciplinei 1.2. Relații cu alte discipline 2. Reprezentarea informației în sistemele de calcul 2.1. Bit/Byte 2.2. Transformări din baza 10 în baza 2 și vice-versa	2	Introducerea temei, discuții și întrebări, rezolvarea ghidată la tablă a unor exemple, dezvoltarea și simularea în mediul ModelSim, urmate de activitate individuală pentru rezolvarea unor exerciții similare.
3. **Prezentarea limbajului Verilog 3.1. **Rolul limbajelor HDL 3.2. **Mediul de dezvoltare/simulare ModelSim 3.3. **Primul program în Verilog 4. Baze de numerație 4.1. Baza 8 4.2. Baza 16	3	
5. Reprezentarea numerelor cu semn 5.1. Reprezentarea în Semn-Mărime 5.2. Reprezentarea în Complement de 1 5.3. Reprezentarea în Complement de 2 5.4. Exerciții de reprezentare a numerelor cu semn folosite ca valori de intrare și ponderi într-o sumă ponderată simplă 6. **Funcții logice și operatori booleeni în Verilog 6.1. **Prezentarea noțiunilor și regulilor de algebră booleeană	4	

6.2. **Reprezentarea funcțiilor booleene în Verilog 6.3. **Operatori bitwise, logici și de reducere în Verilog 6.4. Exerciții în Verilog: implementarea unei reguli booleene simple de acordare a accesului		
7. **Module în Verilog 7.1. **Descrierea modulelor Verilog 7.2. **Intrări, ieșiri și porturi 7.3. **Tipuri de blocuri: initial, always 7.4. **Atribuiri în Verilog 7.5. Structuri repetitive în Verilog 8. **Verificarea modulelor Verilog 8.1. **Module testbench 8.2. **Componentele unui testbench; taskul de sistem $monitor 8.3. Generarea stimulilor de test pentru cereri de acces valide 8.4. **Interpretarea rezultatelor de simulare	4	
9. **Magistrale și vectori în Verilog 9.1. **Declararea semnalelor pe mai mulți biți 9.2. **Selectarea biților și a subvectorilor: part-select 9.3. Implementarea în Verilog a unui comparator pentru verificarea unui prag sau a unui interval permis 9.4. Concatenarea vectorilor 9.5. Operatorul de replicare 9.6. **Exerciții cu magistrale, selecții de biți și multiplexare 10. Coduri de reprezentare a numerelor zecimale în sistemele de calcul 10.1. Codul Binary Coded Decimal (BCD) 10.2. Exerciții de adunare a numerelor în BCD 10.3. Codul Exces de 3 (E3) 10.4. Adunarea numerelor în E3 10.5. **Implementări și conversii în Verilog pentru BCD și E3	6	
11. Reprezentarea numerelor fracționare 11.1. **Conversia părții fracționare din baza 10 în alte baze 11.2. Reprezentarea în virgulă fixă a valorilor de intrare și a ponderilor într-un exemplu elementar din inteligența artificială 11.3. **Reprezentarea numerelor în virgulă mobilă 11.4. Conversia în format IEEE pentru virgulă mobilă 12. Minimizarea funcțiilor logice 12.1. Diagrame Veitch-Karnaugh (VK) 12.2. Codul Gray în organizarea diagramelor Karnaugh 12.3. Minimizarea funcțiilor logice folosind diagrame VK 12.4. Exerciții rezolvate	6	
13. Metoda QUINE-MCCLUSKEY 13.1. Justificarea necesității metodei Quine-McCluskey 13.2. Reprezentarea mintermilor 13.3. Gruparea mintermilor după numărul de biți de 1 13.4. Construirea tabelelor succesive ale grupurilor de mintermi 13.5. Identificarea implicanților și obținerea expresiei minimizate	3	

Bibliografie[14]

1. S. Birla, B. P. Singh, N. K. Shukla: "Digital Design using VerilogHDL: VLSI Modeling, Coding and Verification", Elsevier, ediția I, 2026, ISBN 978-0-443-29088-6
2. Mircea Vlăduțiu: "Computer Arithmetic. Algorithms and Hardware Implementations" Springer, 2012, ISBN 978-3-642-18314-0, ISBN 978-3-642-18315-7
3. B. J. LaMeres: "Introduction to Logic Circuits & Logic Design with Verilog", Springer, ediția a III-a, 2024, ISBN 978-3-031-43948-3
4. S. Churiwala, U. Mehta, V. Dhare: "Digital Design and Verification using Verilog and SystemVerilog", Wiley, 2025, ISBN 978-93-6386-802-1
5. Sarah L. Harris, David Harris: "Digital Design and Computer Architecture, RISC-V Edition" Morgan Kaufmann, 2021, ISBN 978-0-12-820064-3
6. Shirshendu Roy: "Advanced Digital System Design. A Practical Guide to Verilog Based FPGA and ASIC Implementation" Springer, Cham, 2024, ISBN 978-3-031-41084-0, ISBN 978-3-031-41085-7
7. Vaibbhav Taraate: "Digital Logic Design Using Verilog. Coding and RTL Synthesis" Springer, ediția a II-a, 2022, ISBN 978-981-16-3198-6, ISBN 978-981-16-3199-3

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[15]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	1. Descrierea sintetică a unui concept din cele prezentate la curs 2. Aplicarea regulilor algebrei booleene pentru o problemă similară celor rezolvate la curs 3. Proiectarea unei scheme combinaționale, sau secvențiale similară celor exemplificate la curs	Examinare scrisă	22% 22% 22%
9.5 Activităţi aplicative	**S:**		
	L: 1. Soluționarea problemelor propuse în cadrul testelor la activitatea practică de laborator. 2. Prezentarea unor soluții proprii de rezolvare a exercițiilor propuse din timpul laboratorului destinat examinării fiecărui student; Evaluarea temelor de casă	Examinare orală, scrisă sau pe calculator Examinare orală, scrisă sau pe calculator	23% 11%
	P[16]**:**		
	Pr:		

9.6 Standard minim de performanţă (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor[17])

- Reprezentarea numerelor: conversii între bazele 10 și 2, reprezentarea unui număr cu semn în complement de 2 și identificarea unei depășiri a domeniului. Verificare: exercițiu scris.
  Algebră booleană și circuite combinaționale: construirea tabelului de adevăr, minimizarea și implementarea unei funcții prin porți logice. Verificare: problemă de proiectare.
  Circuite secvențiale: identificarea stărilor, tranzițiilor și comportamentului la inițializare ale unui automat finit. Verificare: diagramă sau tabel de stări.

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
21.09.2026	Ș.l.dr.ing. Flavius-Gabriel OPRIȚOIU	Ș.l.dr.ing. Flavius-Gabriel OPRIȚOIU
Director de departament (semnătura)	**Data avizării în Consiliul Facultăţii**[18]	**Decan (semnătura)**
Prof.univ.dr.habil.ing. Mihai MICEA		ș.l.univ.dr.ing. Robert KRISTOF