

## FIȘA DISCIPLINEI<sup>1</sup>

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Politehnica” din Timișoara
1.2 Facultatea <sup>2</sup> / Departamentul <sup>3</sup>	Automatică și Calculatoare / Calculatoare
1.3 Catedra	-
1.4 Domeniul de studii	Calculatoare și Tehnologia informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare / inginer

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Structuri tolerante la defecte</b>						
2.2 Titularul activităților de curs	Ș.I.dr.ing. Lucian PRODAN						
2.3 Titularul activităților de seminar	Asist.dr.ing. Flavius OPRÎȚOIU, drd.ing. Alexandru IOVANOVI						
2.4 Anul de studiu	4	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Opțională

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

3.1 Număr de ore pe săptămână	3	din care:3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	91	din care:3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					21
Tutoriat					7
Examinări					3
Alte activități					
<b>3.7 Total ore studiu individual</b>	49				
<b>3.8 Total ore pe semestru</b>	101				
<b>3.9 Numărul de credite</b>	4				

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

4.1 de curriculum	<ul style="list-style-type: none"> <li>Arhitectura calculatoarelor, Organizarea calculatoarelor, Fundamente de ingineria calculatoarelor</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li>Cunoștințe elementare de design hardware/software</li> <li>Cunoștințe elementare de matematică</li> </ul>

### 5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> <li>Sală de mărime medie, Materiale suport: laptop, proiector, tablă</li> </ul>
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>Laborator cu 15 calculatoare, 10 platforme reconfigurabile diferite (Altera, Xilinx), medii de dezvoltare Quartus și WebPack, tablă</li> </ul>

### 6. Competențe specifice acumulate

<sup>1</sup> Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

<sup>2</sup> Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;

<sup>3</sup> Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

Competențe profesionale <sup>4</sup>	<ul style="list-style-type: none"> <li>• Operarea cu fundamente științifice, ingineresti și ale informaticii</li> <li>• Proiectarea componentelor hardware, software și de comunicații</li> <li>• Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor</li> <li>• Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații</li> <li>• Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații</li> </ul>
Competențe transversale	<ul style="list-style-type: none"> <li>• Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura rezolvarea problemei</li> <li>• Demonstarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională</li> <li>• Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate</li> </ul>

## 7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Dobândirea noțiunilor de bază din domeniul toleranței la defecte și a principalelor metode de augmentare a fiabilității prin aplicarea redundanței la diverse niveluri.</li> <li>• Aprofundarea metodelor de codesign hardware/software în vederea îmbunătățirii unor indicatori de dependabilitate, cu exemplificare în limbajele VHDL/Verilog.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Construirea unei imagini de ansamblu asupra tipurilor de defecte și erori care pot afecta funcționarea sistemelor de calcul, precum și a tehnicilor de revenire din eroare</li> <li>• Consolidarea bazei teoretice însușite de studenți la disciplinele anterioare și dezvoltarea abilităților de operare cu concepte proprii redundanței și toleranței la defectare</li> <li>• Prezentarea atributelor strategiilor de reconfigurare și dezvoltarea abilităților de proiectare a componentelor hardware și software din cadrul sistemelor reconfigurabile în vederea creșterii fiabilității</li> <li>• Instruirea practica a studenților asupra tehnicilor de îmbunătățire a indicatorilor de dependabilitate prin studii de caz</li> <li>• Prezentarea metodelor de implementare a redundanței în scopul creșterii fiabilității la diverse niveluri în sisteme hardware-software</li> </ul>

## 8. Conținuturi

8.1 Curs	Număr de ore	Metode de predare
<b>1. Modelarea defectelor</b> 1.1 Definitii: fault, error, failure 1.2. Evitarea defectelor vs tolerarea defectelor 1.3 Modele de blocare 1.4. Detectabilitate si redundanta 1.5. Echivalenta si localizarea defectelor 1.6. Dominanta	6	Prelegere interactivă susținută de prezentări PPT, conversația, dezbateră, studiul de caz, brainstorming, analiza SWOT
<b>2. Analiza indicatorilor de fiabilitate, disponibilitate și mentenabilitate</b> 2.1 Definitii: rata de defectare, fiabilitate, mentenabilitate, timp mediu pina la defectare, disponibilitate 2.2. Fiabilitatea sistemelor serie, paralel și mixte 2.3. Disponibilitatea sistemelor de calcul 2.4. Mentenabilitatea sistemelor de calcul	4	
<b>3. Toleranța la defecte din perspectivă hardware</b> 3.1. Sisteme canonice serie/paralel. Sisteme non serie/paralel 3.2. Sisteme m-din-n și variante 3.3. Procese Markov. Procese Poisson 3.4. Tehnici de crestere a tolerantei la defectare la nivel de procesor. Procesor watchdog. SMT. 3.5. Defecte bizantine	8	
<b>4. Toleranța la defecte din perspectivă informațională</b> 4.1. Introducere în teoria codurilor. Coduri sistematice și nesistematice 4.2. Coduri Hamming detectoare și corectoare de erori 4.3. Sume de control 4.4. Coduri M-din-N, Berger și coduri ciclice	6	

<sup>4</sup> Aspectul competențelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS ([http://www.rncis.ro/portal/page?\\_pageid=117,70218&\\_dad=portal&\\_schema=PORTAL](http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL)) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fișă și materia în cauză

4.5. Coduri aritmetice 4.6. Sisteme de discuri tolerante la defecte RAID 4.7. Replicare de date. Votare în structuri ierarhice și non-ierarhice 4.8. Toleranța la defecte bazată pe algoritmi		
<b>5. Toleranța la defecte din perspectivă software</b> 5.1. Teste de acceptanță 5.2. Single version FT. Wrappers. Software rejuvenation 5.3. N-version FT. Comparare consistentă. Versiuni independente 5.4. Precondiții, postcondiții și aserțiuni 5.5. Gestionarea excepțiilor 4.6. Checkpointing	2	
<b>6. Studii de caz</b> 6.1. Sistemele NonStop de la TandemComputers. Arhitectura, software, mentenanța 6.2. Sisteme Stratus 6.3. Sisteme IBM: G5 și Sysplex 6.4. Intel Itanium 6.5. Subsistemele Cassini	2	
<b>Bibliografie</b> 1. I. Koren, C. Mani Krishna, Fault Tolerant Systems, Elsevier, 2007. 2. M.L. Shooman: Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design, John Wiley and Sons, 2002. 3. P. Lakey, A.M. Neufelder. System Software Reliability Assurance Guidebook. Rome Laboratory, Rome, NY, 1995. 4. D.K. Pradhan. Fault-Tolerant Computer System Design. Prentice-Hall, 1996. 4. D.P. Siewiorek, R. Swartz. The Theory and Practice of Reliable System Design. Digital Press, 1983. 5. K.S. Trivedi. Probability and Statistics with Reliability, Queueing, and Computer Science Applications. Wiley-Interscience, 2001.		
<b>8.2 Seminar/laborator</b>	<b>Număr de ore</b>	<b>Metode de predare</b>
1. Redundanță hardware: implementarea unui sistem simplu TMR, modulul de bază fiind reprezentat de un sumator paralel pe 4 biți (arhitectura la alegere). Modelare în VHDL/Verilog (sau vizual), simulare în VHDL/Verilog. Demonstrare pe Altera DE2.	2	Expunere temă, prelegerea interactivă, abordare euristică, dezbateri, controversă creativă, problematizare, algoritmizare, brainstorming.
2. Redundanță hardware: implementarea unei structuri simple procesor-watchdog pentru magistrala de 4 biți (arhitectura la alegere). Modelare în VHDL/Verilog (sau vizual), simulare în VHDL/Verilog. Demonstrare pe Altera DE2.	4	
3. Redundanță informațională: implementarea unui sistem de detectare și corecție a erorii simulare bazat pe coduri Hamming (7,4). Modelare în VHDL/Verilog (sau vizual), simulare în VHDL/Verilog. Demonstrare pe Altera DE2.	4	
5. Redundanță informațională: implementarea unui comparator pe k biți de tip totally self-checking dual rail. Particularizare pe 8 biți. Modelare în VHDL/Verilog (sau vizual), simulare în VHDL/Verilog. Demonstrare pe Altera DE2.	4	
<b>Bibliografie</b> 1. M.L. Shooman: Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design, John Wiley and Sons, 2002. 2. P.K. Lala. Self-checking and fault-tolerant digital design. Morgan-Kaufmann, 2001. 3. D.P. Siewiorek, R. Swartz. The Theory and Practice of Reliable System Design. Digital Press, 1983. 4. P.A. Lee, T. Anderson. Fault-tolerance principles and practices. Springer, 1990.		

### 9. Corelarea conținutului disciplinei cu cerințele specialiștilor din domeniu și cu așteptările angajatorilor reprezentativi

<ul style="list-style-type: none"> <li>Cunoștințele legate de estimarea fiabilității și aplicarea tehnicilor de redundanță în scopul obținerii toleranței la defecte constituie parte integrantă din profilul unui absolvent de studii superioare în domeniul IT</li> <li>O mare parte a angajatorilor din domeniul IT (atât software, cât și hardware) este direct interesată de creșterea fiabilității produselor oferite, ceea ce stimulează punerea în practică a cunoștințelor oferite de această disciplină. Mai mult, disciplina conferă și cunoștințe legate de estimarea perioadei până la defectare, aceasta fiind esențială în construirea strategiilor de marketing și stabilirea condițiilor de garanție și post-garanție.</li> </ul>
--

### 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	Prezentarea unor subiecte impuse sub formă de sinteză	Examinare scrisă	40 %
	Stilul de corelare a detaliilor tehnice corespunzătoare subiectelor	Examinare scrisă	10 %
	Gradul de aprofundare a subiectelor	Examinare scrisă	10 %
10.5 Seminar /laborator	Implementarea lucrărilor corespunzătoare de laborator	Prezentarea rezolvărilor, răspunsuri la întrebări. Demonstrarea rezolvării prin rularea pe platforma hardware	30 %
	Argumentarea deciziilor luate și stil de design	Examinare orală și pe platforma hardware	5 %
	Prezența	Evidența prezenței	5 %
10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)			
<ul style="list-style-type: none"> <li>• Proiectarea unei structuri de complexitate medie care să conțină cel puțin un tip de redundanță cu argumentarea deciziilor luate și prezentarea efectelor în creșterea fiabilității</li> <li>• Cunoștințe de bază legate de principalele tipuri de redundanță și tehnicile de implementare</li> <li>• Subiectele de examen trebuie să totalizeze cel puțin 5 puncte</li> </ul>			

### 11. Compatibilitate internațională

<ul style="list-style-type: none"> <li>• Colorado State University. CS530 – Fault-Tolerant Computing, CS635 – Advanced Fault-Tolerant Computing</li> <li>• University of Victoria. CSC 454 – Fault Tolerant Computing</li> <li>• University of Wisconsin-Madison. ECE753 – Fault-Tolerant Computing</li> </ul>
--

Data  
completării

Semnătura titularului de curs

Semnătura titularilor de seminar

Ș.I.dr.ing. Lucian PRODAN

Asist.dr.ing. Flavius OPRIȚOIU    Drd.ing. Alexandru IOVANOVICI

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Ing. Vladimir Ioan CREȚU