

FIȘA DISCIPLINEI

1. Date despre program

Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
Facultatea	Facultatea de Inginerie
Departament	Departamentul de Calculatoare și Inginerie Electrică
Domeniul de studiu	Calculatoare și Tehnologia Informației
Ciclul de studii	Studii de licență/master
Specializarea	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	Sisteme Încorporate (Embedded Systems)			
Codul cursului	Tipul cursului	An de studiu	Semestrul	Număr de credite
390458030611SA62	Optional	4	2	3
Tipul de evaluare	Categoriza formativă a disciplinei (DF=fundamentală.; DD=domeniu; DS=specialitate; DC=complementară)			
Examen	DS			
Titular activități curs	conf. dr. ing. Adrian FLOREA			
Titular activități seminar / laborator/ proiect	conf. dr. ing. Adrian FLOREA			

3. Timpul total estimat

Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână				
Curs	Seminar	Laborator	Proiect	Total
2		2		4
Extinderea disciplinei în planul de învățământ – Total ore din planul de învățământ				
Curs	Seminar	Laborator	Proiect	Total ($NOAD_{sem}$)
24		24		48

Distribuția fondului de timp pentru studiu individual		Nr.ore
Studiul după manual, suport de curs, bibliografie și notițe		7
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren		2
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri		4
Tutoriat:		4
Examinări:		2
Total ore alocate studiului individual ($NOSI_{sem}$)		19
Total ore pe semestru ($NOAD_{sem} + NOSI_{sem}$)		67

4. Precondiții (acolo unde este cazul)

De curriculum	Cunoștințe de bază de arhitectura calculatoarelor (microprocesoarelor), limbaje de programare de nivel înalt, sisteme de operare, algoritmi.
De competențe	

5. Condiții (acolo unde este cazul)

De desfășurare a cursului	Participare activă, lectura suportului de curs Tablă, videoproiector
De desfășurare a sem/lab/pr	Elaborarea și susținerea lucrărilor planificate Sală dotată cu calculatoare cu sistem de operare Ubuntu (Linux).

6. Competențe specifice acumulate

Competențe profesionale	C1 – Operarea cu fundamente științifice, ingineresti și ale informaticii C2 – Proiectarea componentelor hardware, software și de comunicații C4 – Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații
Competențe transversale	

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

Obiectivul general al disciplinei	Disciplina are rolul de a prezenta studenților într-o manieră integratoare noțiuni de specialitate (hardware și software) legate de un domeniu de mare interes la ora actuală utilizat în toate aplicațiile industriale sau casnice, și anume cel al sistemelor dedicate, cu accent pe elementul central al acestora și anume arhitecturile de calcul folosite, în special procesoare VLIW (very long instruction word). Din punct de vedere istoric, domeniul sistemelor embedded (dedicate) rareori se intersecta cu cel al arhitecturilor de calcul de tip VLIW. Dezvoltările tehnologice au reușit să înlăture această separare: tehnicile de procesare avansate de tip VLIW, care pareau scumpe pentru domeniul sistemelor embedded au devenit fezabile, pretabile, acceptabile ca și cost pentru acestea. Această schimbare a determinat trecerea într-o nouă epocă de proiectare a sistemelor de calcul dedicate, în care un procesor de înaltă performanță este elementul central, celelalte elemente (periferice, neprogramabile) de interconectare ale sistemului având un rol secundar. Proiectarea sistemului trebuie făcută ținând cont și de instrumentele software dedicate arhitecturii suport (compilatoare, debugger-e, simulatoare) necesare pentru dezvoltarea de aplicații.
Obiectivele specifice	<ol style="list-style-type: none"> 1. Înțelegerea conceptelor fundamentale referitoare la sistemele de calcul dedicat: hardware (arhitecturi), software (tehnologii și metodologii de evaluare) și aplicații. 2. Cunoașterea funcțiilor de bază ale sistemelor dedicate: capacitatea de a proiecta un sistem, o componentă sau proces care să corespundă unor constrângeri realiste, cum ar fi din punct de vedere economic, al mediului, etc, al sănătății și siguranței, al producției, și durabilității 3. Cunoașterea structurii sistemelor dedicate 4. Cunoașterea și dezvoltarea de aplicații cu sisteme dedicate 5. Dezvoltarea și înțelegerea tehnologiilor folosite ca suport pentru sistemele dedicate: capacitățile tehnologice și limitările hardware, componentele software, metode de evaluare a compromisurilor de proiectare între diferite opțiuni tehnologice 6. Dezvoltarea abilităților de programare pentru sisteme dedicate (C, C++ sub sistem de operare Linux Ubuntu, etc)

8. Conținuturi

Curs		Nr. ore
Curs 1	Organizarea cursului. Prezentarea structurii, bibliografiei, modului de desfășurare a activității și a evaluării. Introducere în procesarea dedicată. Legea lui Moore și calculatoare omniprezente. Ce sunt sistemele de calcul dedicat? Diferențe între procesoarele dedicate și cele aferente sistemelor de uz general.	2
Curs 2	Caracterizarea sistemelor dedicate: atribute și limitări fizice în domeniul embedded. Categorii de aplicații cu sisteme dedicate. Metodologii de proiectare. Productivitate versus complexitate.	2
Curs 3	Procesoare RISC cu paralelism la nivelul instrucțiunii (ILP). Mașini cu execuție multiplă: procesoare superscalare și VLIW. Asemănări și deosebiri. Rolul compilatorului în detecția hazardurilor de date. Procesoarele VLIW în calculul dedicat. Procesoare de semnal.	2
Curs 4	Problema hazardurilor în arhitecturile VLIW. Soluții software de eliminare a acestora. Tehnici de optimizare la nivelul compilatorului. Regula 90:10. <i>Loop-Based Strength Reduction, Induction Variable Elimination, Loop-Invariant Code Motion, Loop interchange, Merging array, Loop unswitching.</i>	2
Curs 5	Tehnici de optimizare la nivelul compilatorului. <i>Loop Unrolling, Software Pipelining, Enhance Percolation, Procedure Splitting, List Scheduling.</i> Optimizări globale de tip <i>Trace Scheduling, Function inlining.</i>	2
Curs 6	Analiza performanței programelor. <i>Control dependence graph (CDG). Worst-case execution time (WCET).</i> Considerații referitoare la consumul de putere și energie.	2
Curs 7	Verificare pe parcurs din materia studiată.	2
Curs 8	Procese și sisteme de operare în timp real. Planificarea proceselor care să permită o scalare dinamică a tensiunii de alimentare în vederea satisfacerii restricțiilor de timp (<i>deadline</i>) concomitent cu menținerea unei tensiuni joase satisfăcătoare din punct de vedere a puterii consumate.	2
Curs 9	Instrumente software pentru generarea codului specific arhitecturilor embedded. Compilatoare, Asamblatoare, Link-editoare, Debugger-e, Simulatoare. Compilatorul VEX (VLIW example).	2
Curs 10	Diferențe între compilatoare pentru sisteme embedded versus pentru sisteme de uz general: constrângeri de memorie, timp real și consum redus de putere. Profiling și granularitatea informațiilor colectate. Scheduling. Generarea codului mașină și compilare la nivel Back-End.	2
Curs 11	Acceleratoare pentru execuția pe arhitecturi dedicate a rețelelor neuronale.	2
Curs 12	Recapitulare finală. Parcurgerea conceptelor esențiale.	2
Total ore curs:		24
Laborator		Nr. ore
Lab 1	Organizarea laboratorului. Prezentarea structurii, bibliografiei, modului de desfășurare a activității și a evaluării.	2
Lab 2	Utilizarea simulatorului VLIW-DLX: ilustrarea principiilor fundamentale ale procesării VLIW (very long instruction word). Rolul software-ului în detecția și eliminarea hazardurilor RAW. Relația dintre instrucțiunea multiplă și instrucțiunile RISC primitive și independente, care vor fi alocate unităților de execuție în conformitate strictă cu poziția lor în instrucțiunea multiplă	2

	(număr/latențe). Optimizări software in procesoarele VLIW (loop unrolling, software pipelining). Avantaje / Dezavantaje față de procesoarele superscalare. Aplicabilitate – sisteme dedicate (procesoare de semnal, procesoare multimedia). Aplicație rezolvată – execuția pe un procesor VLIW a unui program care înmulțește două matrici pătratice și care translatează un vector de numere flotante. Comparatie timp de execuție față de varianta scalară.	
Lab 3	Instalare SO Linux Ubuntu. Descărcarea compilatorului VEX (http://www.hpl.hp.com/downloads/vex/vex-3.43.i586.tgz), instalarea și testarea pe un program simplu care determină maximul dintr-un șir de numere. Sesizarea diferențelor dintre compilatorul pentru arhitectura nativă (Intel) versus cel dedicat pentru arhitectura simulată VLIW.	2
Lab 4	Parcugerea manualului de utilizare a instrumentului VEX (http://home.deib.polimi.it/ashouri/courses/VEX_manual.pdf) și testarea fiecărei opțiuni de compilare (-mas_t, etc) pe diverse programe. Analizarea statisticilor privitoare la configurația cache-urilor și a ratelor de hit obținute (fișierele ta.log####). Prezentarea rezultatelor sub forma unui tutorial.	2
Lab 5	Aplicarea tehnicilor de optimizare la nivelul compilatorului: <i>Loop-Based Strength Reduction, Induction Variable Elimination, Loop-Invariant Code Motion, Loop interchange, Merging array, Loop unswitching</i> . Măsurarea timpului de execuție în varianta în care codul este neoptimizat (-O0) sau optimizat cu loop unrolling și trace scheduling (-O3).	4
Lab 6	Setul de instrumente VEX permite compilarea pentru arhitecturi de procesare diferite. Compilați și executați aplicația "decodorul MPEG2" pentru arhitectura scalară RISC, și pentru arhitecturi VEX2, VEX4 și VEX8. Comparați rezultatele în ceea ce privește timpul de execuție, IPC, ciclul în care procesarea stagnează și NOP. https://www.complang.tuwien.ac.at/cd/vliw/assignment2.pdf Implică modificări în Makefile și schimbată configurația în momentul compilării (risc.mm, etc).	2
Lab 7	Aplicarea unor informații de profiling (gprof fis1 gmon-icache.out). Analiza apelurilor de funcții.	2
Lab 8	Aplicarea tehnicilor de optimizare Loop unrolling și Software pipelining și determinați câștigul de performanță față de varianta neoptimizată pe un cod sursă similar lucrării Lab2.	2
Lab 9	Vizualizați influența parametrilor mașinii VLIW (latenței instrucțiunilor de salt) asupra performanței de procesare. Modificare în -fmm=config.mm flagurile posibile (DEL, RES, REG, etc).	2
Lab 10	Testarea aplicațiilor demonstrative de la adresa https://sites.google.com/site/alarivliw11/demo.tgz?attredirects=0 . Analiza grafic vizuală a rezultatelor – prezentare grafuri de control și de date.	2
Lab 11	Aplicații bazate pe modificarea parametrilor ierarhiei de memorie. Se modifică fișierul vex.cfg .	2
Lab 12	Evaluarea finală a activității de laborator	2
Total ore laborator		24

Metode de predare

Prelegeri, problematizări, studii de caz, exerciții, conversații, explicații, demonstrații și dezbateri. Metodele de predare la curs se bazează pe folosirea videoproietorului, dar în anumite cazuri în care se impune se folosește creta și tabla pentru diverse demonstrații sau rezolvări de exemple / probleme.	Limba de predare	Română
---	------------------	--------

Bibliografie

Referințe bibliografice recomandate	1. Joseph A. Fisher, Paolo Faraboschi, Cliff Young, <i>Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools</i> , 1st Edition, Morgan Kaufmann, 2005.
	2. https://sites.google.com/site/alarivliw11/
	3. Wayne Wolf, <i>High-Performance Embedded Computing: Architectures, Applications, and Methodologies</i> , 1st Edition, Morgan Kaufmann, 2007.
	4. http://www.ann.ece.ufl.edu/courses/eel6935_11fal/
	5. Vințan L. - <i>Arhitecturi de procesoare cu paralelism la nivelul instrucțiunilor</i> , Editura Academiei Române, București, ISBN 973-27-0734-8, 2000 (264 pg.) (http://webspace.ulbsibiu.ro/lucian.vintan/html/Organizarea.pdf).
	6. Vințan N. L., Florea A. – <i>Microarhitecturi de procesare a informației</i> , Editura Tehnică, București, ISBN 973-31-1551-7, 2000 (312 pg.) (http://webspace.ulbsibiu.ro/adrian.florea/html/docs/CARTEA_PDF.pdf).
Referințe bibliografice suplimentare	1. Peter Marwedel, <i>Embedded System Design</i> , Springer, 2006;
	2. John Catsoulis, <i>Designing Embedded Hardware</i> , O'Reilly, 2005.

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

--

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Ponderea în nota finală	Obs.*
Curs	Verificare pe parcursul semestrului	Examen scris	20%	nCPE
	Examen de semestru	Examen scris	50%	CEF
	Alte activități: prezența la curs	-	5%	nCPE
Laborator	Verificare finală	Aplicație practică	25%	CEF
Standard minim de performanță				
<ul style="list-style-type: none"> Cunoașterea, înțelegerea și explicarea noțiunilor elementare specifice sistemelor de calcul dedicate. Interes constant manifestat pentru însușirea disciplinei. Îndeplinirea condițiilor minime obligatorii (50%) în privința temelor de casă, referatelor și a testelor date pe parcursul semestrului. 				

(*) CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală;



ULBS

Universitatea "Lucian Blaga" din Sibiu

Ministerul Educației Naționale
Universitatea "Lucian Blaga" din Sibiu
Facultatea de Inginerie

Departamentul de Calculatoare și Inginerie Electrică

Data completării: 10/09/2014

Data avizării în Departament:.....

	Grad didactic, titlul, prenume, numele	Semnătura
Titular disciplină	conf. dr. ing. Adrian FLOREA	
Director de departament	Prof. dr. ing. Daniel VOLOVICI	