

## PREFAȚĂ

Scopul principal al acestei cărți este acela de a familiariza cititorul cu anumite aspecte, considerate mai importante și, de asemenea, mai puțin perisabile, referitoare la arhitectura microprocesoarelor de azi și, eventual, chiar de mâine. Așadar, cartea de față are un preponderent caracter formativ, focalizându-se pe prezentarea unor cunoștințe generice, fundamentale, oarecum independente de implementări particulare, comerciale. Totuși, bazat pe propria experiență de predare [Vin15b], dar și de cercetare științifică a autorului în domeniul arhitecturilor de calcul, credem că prezentarea este una originală, reflectând o meditație și un efort de înțelegere relativ îndelungate asupra subiectelor tratate, situate la nivelul interfeței *hardware-software*. Pentru fiecare metodă sau tehnică prezentată s-a încercat revelarea metaforei cognitive care stă la baza acesteia, în speranța că, astfel, cunoștințele se vor consolida. Uneori, anumite contribuții originale ale autorului în domeniul sistemelor de calcul, sau ale unor cercetători consacrați, publicate în reviste și în conferințe internaționale de un bun nivel științific, sunt integrate în prezentare, “topite” într-o formă mai accesibilă (spre exemplu, prezentarea fundamentelor predicției dinamice neuronale a instrucțiunilor de salt condiționat, idee introdusă în premieră de autor, la nivel mondial, prin publicarea unui articol la conferința *International Joint Conference on Neural Networks, Washington DC, USA, 10-16 July, 1999*). Considerăm că integrarea unor contribuții științifice ale autorului, sau familiare acestuia, constituie un demers binevenit într-un tratat universitar, contribuind la o viziune specifică utilă, la puncte de vedere care pot stârni interesul, curiozitatea și chiar dezacordul (!) cititorului. Credem că este nevoie de asemenea cărți, care nu intră – pentru că nici nu ar putea-o face – în concurență cu marile *text-book*-uri ale domeniului (unele citate în bibliografie), ci, dimpotrivă, prezintă domeniul într-un mod mult mai succint și insistând pe tematicile mai familiare autorului, printr-o experiență de cercetare științifică proprie.

Acest tratat universitar reutilizează și integrează într-un tot unitar, sub o formă îmbunătățită și extinsă într-un mod semnificativ, părți ale unor lucrări anterioare, scrise și publicate de autor sub forma unor monografii sau articole tehnico-științifice. Autorul a rescris și a actualizat anumite părți ale acestor lucrări, care abordau chestiuni esențiale referitoare la arhitectura sistemelor de calcul, revizuiindu-le, sintetizându-le și îmbogățindu-le în cartea de față, în speranța că sub această formă nouă, poate mai didactică decât versiunile

anterioare, informația prezentată va fi mai ușor asimilabilă de către studenții și specialiștii interesați. S-au utilizat în acest scop, în principal, lucrările VINȚAN N. LUCIAN – *Arhitecturi de procesoare cu paralelism la nivelul instrucțiunilor*, Editura Academiei Române, București, 2000; VINȚAN N. LUCIAN – *Prediction Techniques in Advanced Computing Architectures* (în limba engleză), Matrix Rom Publishing House, Bucharest, 2007; VINȚAN N. LUCIAN, FLOREA ADRIAN – *Microarhitecturi de procesare a informației*, Editura Tehnică, București, 2000; FLOREA ADRIAN, VINȚAN N. LUCIAN – *Simularea și optimizarea arhitecturilor de calcul în aplicații practice*, Editura Matrix Rom, București, 2003, dar și altele, importante, ale autorului sau ale altora, precum anumite *text-book*-uri de notorietate și apreciere mondiale în domeniul sistemelor de calcul (v. bibliografia cărții). Așadar, textele preluate din propriile lucrări ale autorului au fost rescrise, cu modificări și adăugiri semnificative și structurate aici sub forma unei cărți de sine stătătoare, care “curge” natural, de la simplu la complex și care, speră autorul, dă seama asupra unor aspecte tehnico-științifice fundamentale ale microprocesoarelor de uz general, ale modului în care acestea procesează programele. În ciuda caracterului preponderent formativ, la nivel universitar, considerăm că o parte importantă a acestei lucrări conține informații și cunoștințe izvorâte dintr-o experiență vie a autorului în acest domeniu, atât pe plan didactic, cât și științific, care ar putea trezi curiozitatea specialiștilor. Spre exemplu, în acest sens, arhitectura sistemelor de calcul este prezentată, în premieră într-un curs universitar credem noi, într-o strânsă legătură cu anumite metode de învățare automată (*machine learning*, de tip rețele neuronale, algoritmi genetici etc.) dar și cu anumite metode matematice, deosebit de utile în analiza unor modele de procesare (algebre logice de tip *fuzzy*, metode euristice de optimizare multi-obiectiv, teoria informației etc.) Aceasta exprimă, de fapt, viziunea interdisciplinară a autorului, dezvoltată și prin activitatea sa științifică de mai bine de 25 de ani, prin care încearcă să contribuie, după modestele sale puteri, la maturizarea empiricei discipline ingineresti numite “arhitectura sistemelor de calcul”, prin utilizarea și adaptarea unor metode teoretice mai riguroase. Printre lucrările de pionierat în acest sens au fost și cele semnate de autor, fertile prin multiple citări independente, anume STEVEN G., VINȚAN L. - *Modelling Superscalar Pipelines with Finite State Machines, "Proceedings of the 22<sup>nd</sup> Euromicro'96 Conference. Beyond 2000: Hardware/Software Design Strategies", September 1996, Prague, Czech Republic, pp. 20-25, IEEE Computer Society Press, Los Alamitos, California, USA, ISBN 0-8186-7703-1,*

*Library of Congress Number 96-79894* respectiv **VINȚAN L.** - *Towards a High Performance Neural Branch Predictor, Proceedings of The International Joint Conference on Neural Networks - IJCNN '99 (CD-ROM, ISBN 0-7803-5532-6), pp. 868 – 873, vol. 2, Washington DC, USA, 10-16 July, 1999* (această a 2-a lucrare a introdus, în premieră, conceptul de predictor dinamic neuronal în arhitectura calculatoarelor, având 57 de citări independente până în anul 2015.)

Într-adevăr, arhitectura sistemelor de calcul este încă o știință inginerască preponderent empirică, insuficient de matură din punct de vedere teoretic, bazată în principal pe metode de *benchmarking*. Dezvoltarea sa a fost una predominant conjuncturală, generată deseori de limitări tehnologice particulare, lipsindu-i un cadru de dezvoltare riguros, matematizat, încă de la începuturi. Astfel, ideile novatoare ale domeniului au apărut, deseori, fără a se conștientiza suportul lor teoretic mai adânc, deseori comun. Spre exemplu, ideea de memorie virtuală apare implementată prin anul 1962, înaintea celei de memorie cache (1965). Înțelegerea faptului că, în esență, sunt idei cu o bază comună din punct de vedere teoretic (al teoriei statistice a probabilităților), a apărut ulterior. Un alt exemplu: predictoarele dinamice aferente instrucțiunilor de salt condiționat s-au dezvoltat fără ca inventatorii lor să înțeleagă că acestea sunt, de fapt, predictoare stohastice de tip *Markov*. Abia în anul 1996, dr. *Trevor Mudge* înțelege acest fapt și publică un articol lămuritor, dar, din păcate, cu efecte limitate asupra domeniului. În fine, un alt exemplu: procesarea vectorială nu este prezentată în literatura de specialitate în contextul conceptului natural, mai general, de spațiu vectorial. Conștientizarea faptului că procesarea vectorială are ca bază teoretică noțiunea fertilă de spațiu vectorial euclidian normat, ar putea avea nu doar un beneficiu cognitiv, ci și unul utilitar, concret. În baza acestei conștientizări s-ar putea dezvolta, spre exemplu, procesoare cu facilități hardware de evaluare a similarității a doi vectori, pe baza unor norme matematice, care ar putea accelera semnificativ aplicații de tip clasificare/clustering. Și astăzi remarcăm o dezvoltare oarecum dezordonată, conjuncturală, a domeniului (a se vedea, spre exemplu, dezvoltarea *ad-hoc* a sistemelor *multicore* din ultimii ani, precum și cea a limbajelor de programare concurente, prin intermediul cărora acestea să fie exploatate corespunzător.)

O mare problemă teoretică a domeniului arhitecturilor de calcul constă în faptul că cercetările, în marea lor majoritate bazate pe *benchmarking*, după cum am mai menționat, nu sunt reproductibile sau sunt extrem de dificil (laborios) reproductibile. Complexitatea domeniului, legată în principal de procesarea unor programe obiect de mari dimensiuni și care și-au pierdut semantica în urma compilării, este departe de a fi stăpânită în

mod corespunzător. Metodologiile de cercetare și de evaluare a performanțelor sunt încă relativ empirice. De aceea, credem că orice efort de a „matematiza” această știință predominant empirică, merită subliniat.

**În ciuda caracterului preponderent formativ al acestei cărți, ea încearcă să sugereze cititorului și anumite idei științifice, unele, poate, chiar novatoare.** De altfel, la nivelul unui tratat sau curs universitar, amprenta științifică sau de interpretare originală a autorului sunt nu doar binevenite ci, credem noi, chiar necesare. (În caz contrar, am preda cu toții, 100% după *text-book*-urile clasice, într-o uniformizare păguboasă, care nu este specifică universităților autentice.) Astfel, **un mesaj esențial al acestei cărți este acela că, domeniul numit *Computer Architecture* constituie, de fapt, un set uriaș de studii de caz pentru cercetări științifice autentice, mature, pe o bază matematică rafinată. Mai mult, domeniul empirico-ingineresc al arhitecturii sistemelor de calcul poate induce și motiva cercetări științifice mai generale, cu adevărat profunde.** Pentru asta însă, este nevoie de o generalizare a problemei particulare, tratate în acest context pur ingineresc. Iată doar câteva asemenea probleme deschise, potențial fertile, cel puțin în opinia autorului, prezentate în carte în mod natural și izvorâte din chiar propria sa experiență de cercetare:

- Care este legătura dintre complexitatea programelor dinamice deterministe și comportamentul lor, uneori impredictibil, entropic, cvasi-aleator?
- Utilizarea cunoștințelor de domeniu în optimizarea multi-obiectiv a sistemelor de calcul, ridică probleme științifice serioase, legate de reprezentarea adecvată a cunoașterii de domeniu. Cum se poate reprezenta în mod eficient cunoașterea din domeniul sistemelor de calcul, în vederea optimizării eficiente a acestora? Cum poate fi integrată această cunoaștere specifică, în algoritmi generali de optimizare (în general, euristici)?
- Reprezentarea cunoașterii de domeniu, prin reguli logice de tip fuzzy, prezentată în lucrare (la finele Capitolului 4), conduce la problema deschisă a determinării gradului de contradicție intrinsec, aferent unei astfel de micro-ontologii de domeniu, implicând probleme conexe importante (spre exemplu, dacă acest grad de contradicție este prea mare, ce se poate face? Eliminarea unor reguli? Care?) Generalizări ale problemei, în contextul unor

ontologii mai generale (ca semantică, dar și ca mod de reprezentare), ar putea fi de mare interes, atât pe plan cognitiv, cât și utilitar (spre exemplu, stabilirea gradului de contradicție existent într-un text scris în limbaj natural).

- Ce ar putea deveni paradigma de optimizare multi-obiectiv de tip Pareto, într-o abordare a mulțimilor (fronturilor) Pareto în paradigma teoriei fuzzy a mulțimilor? (Adică, să se determine gradul de apartenență la frontul Pareto al fiecărui individ care aparține acestuia, pe baza gradului mutual de dominanță. În toate abordările actuale, un individ poate doar să aparțină, sau nu, frontului Pareto.)
- Abordările meta-algoritmice, de genul meta-predicțiilor, meta-clasificărilor, meta-optimizărilor etc., prezentate în această carte strict în contextul arhitecturii calculatoarelor, induc în mod natural noțiunea de sinergie. (Altfel, meta-algoritmica sau abordările hibride n-ar mai prea avea sens.) Cum s-ar putea ajunge la o teorie matematică riguroasă a noțiunii de sinergie, în context meta-algoritmice? Care este legătura între sinergia aceasta și sistemele neliniare din ingineria sistemelor? Etc.

Considerăm asemenea probleme deschise ca fiind potențial fertile. Rezolvarea lor (aspectul cognitiv deci) ar putea conduce la soluționarea multor probleme utilitare derivate. Pe de altă parte, asemenea abordări interdisciplinare dezvăluie o perspectivă mai profundă a acestei discipline, considerată preponderent empirică, **cu utilitate multă** (dată, în fond, de toate dispozitivele electronice digitale de calcul pe care le folosim), **dar implicând cunoaștere științifică relativ puțină**. Credem că situația este similară pentru multe alte discipline din domeniul mai larg al ingineriei sistemelor de calcul.

Pe scurt, lucrarea este structurată astfel. Debutul se face cu o prezentare a istoriei sistemelor electronice de calcul numeric, insistându-se nu doar pe geneza și evoluția ideilor tehnico-științifice, dar și pe oamenii care le-au dezvoltat. Apoi, se face o prezentare sintetică a structurii și funcționării unui microsistem generic de calcul, insistându-se asupra aspectelor fundamentale (microprocesor, memorii, interfețe de I/O, procesarea instrucțiunilor, moduri de lucru între microprocesor și dispozitivele periferice etc.) Capitolul 2 prezintă arhitectura sub-sistemului de memorie al unui sistem de calcul. Se arată că între viteza microprocesorului (de ordinul sutelor de picosecunde în cazul celor mai

avansate) și timpul de răspuns al memoriei principale (de ordinul zecilor de nanosecunde), respectiv al celei secundare (de ordinul câtorva milisecunde), există o “prăpastie semantică”. În consecință, se prezintă caracteristicile principale ale ierarhiei de memorii cache, dar și mecanismul de memorie virtuală. Înainte de aceste detalii însă, autorul prezintă necesitatea acestor soluții, problema propriu-zisă, care, deseori este mai importantă chiar decât unele soluții particulare. De altfel, acest mod de prezentare, care startează cu prezentarea cât mai clară a problemei puse în discuție, a importanței acesteia, reprezintă un invariant al lucrării (credem că multe cursuri universitare păcătuiesc prin prezentarea unor soluții excesiv de complicate, fără o precizare clară a problemelor aferente acestora și a importanței acestor probleme în contextul dat; plastic spus, “dețin soluție complicată, caut problema corespunzătoare!”). În capitolul următor se prezintă fundamentele microprocesoarelor RISC scalare, cu procesare *pipeline* a instrucțiunilor. Se insistă aici, în mod clasic, pe problemele hazardurilor în structurile pipeline de procesare a instrucțiunilor și, în consecință, pe schițarea principalelor soluții propuse în literatura de specialitate. De asemenea, se prezintă aspecte importante legate de problematica excepțiilor în procesoarele *pipeline*, analiza alias-urilor de memorie (*memory disambiguation*), execuția predicativ-speculativă (*sic!*) a instrucțiunilor etc. Capitolul 4 generalizează abordarea celui precedent, referindu-se, în principal, la microprocesoarele cu paralelism la nivelul instrucțiunilor, dar și la alte tipuri de sisteme de calcul mai avansate (spre exemplu, sisteme predictiv-speculative, *multithreading*, *multicore* etc.) Se analizează atât abordările hardware (algoritmul lui Tomasulo, bufferul de reordonare etc.) cât și cele software (*scheduling* static al programului obiect) care urmăresc acest scop (*Instruction Level Parallelism*), inclusiv pe baza unor studii de caz. Tot aici, se prezintă ideile principale care stau la baza microprocesoarelor cu procesări multifir, arhitecturile de calcul vectorial (SIMD), care exploatează paralelismul la nivelul datelor, dar și câteva elemente fundamentale referitoare la sistemele paralele de tip multiprocesor (*multicore*, MIMD). De asemenea, se prezintă în premieră în literatura tehnică românească, cel puțin după știința autorului, un paragraf focalizat pe câteva metode euristice de optimizare automată, de tip multi-obiectiv, aplicate sistemelor de calcul complexe. Acestea sunt augmentate prin utilizarea unor cunoștințe din domeniul arhitecturii procesoarelor, exprimate prin logici de tip fuzzy, care le fac mai eficiente, dar și mai performante. Pe baza experienței de cercetare științifică a autorului, se face aici inclusiv o introducere în problematica meta-optimizării sistemelor de calcul, constând în utilizarea

concurrentă a mai multor algoritmi de optimizare, cu beneficii sinergice. Se continuă cu probleme propuse spre rezolvare, care provoacă cititorul să aplice în mod practic-aplicativ, cunoștințele expuse în carte. Rezolvarea de aplicații practice este esențială în procesul de învățare (și) al acestui domeniu. Lucrarea se încheie cu o bibliografie selectivă și cu un glosar, în care se încearcă explicarea sintetică a principalilor termeni tehnici utilizați în carte (deseori acești termeni, preluați din limba engleză, nu mai necesită traduceri în limba română, intrând în vocabularul tehnic al specialiștilor sub forma originală.) În mod deliberat, autorul a explicat în mod redundant anumite concepte, dând deseori formulări echivalente alternative, utilizate în literatura de specialitate, în virtutea anticului principiu pedagogic care afirmă că *repetitio (est) mater studiorum*.

Pentru cititorul care urmărește strict însușirea unor aspecte pur formative ale arhitecturii microprocesoarelor de uz general, fără a fi deci preocupat momentan de probleme mai avansate, recomandăm următorul traseu de parcurgere a cărții:

- Capitolul 1- integral (O introducere în filosofia microsistemelor de calcul)
- Capitolul 2 – integral (Memorii *cache* și memoria virtuală)
- Capitolul 3 (Microprocesoare *pipeline* scalare de tip *RISC*), fără sub-paragrafele intitulate “Problema salturilor condiționate impredictibile” și „Fundamentele predicției neuronale a *branch*-urilor”
- Capitolul 4: Procesoare cu execuții multiple ale instrucțiunilor (*Instruction Level Parallelism*) – Paragrafele 4.1, 4.2, 4.3, 4.4. În plus, din acest capitol:
  - Microprocesoare *multi-thread* (Par. 4.5b)
  - Microprocesoare cu *scheduling* static: Par. 4.6 (optimizare locală – metoda *List Scheduling*), Par. 4.7 (optimizare globală – *Trace Scheduling*), Par. 4.8 (optimizarea buclelor de program – *Loop Unrolling* și *Software Pipelining*), Par. 4.9 (Microarhitecturi *TTA*) și Par. 4.11 (Arhitecturi cu paralelism la nivelul datelor – *SIMD* sau vectoriale)
  - Sisteme multiprocesor – Par. 4.12, fără sub-paragraful intitulat „Direcții de dezvoltare actuale în sistemele *multicore /manycore*”

Ne exprimăm așadar speranța că această lucrare, sub forma unui tratat universitar unitar, se va dovedi utilă studenților din domeniul științei / ingineriei calculatoarelor și tehnologiei informației sau din domenii conexe (electronică și telecomunicații, ingineria sistemelor, inginerie electrică etc.), dar și specialiștilor

care doresc să-și consolideze cunoștințele referitoare la bazele arhitecturale ale microprocesoarelor și sistemelor de calcul moderne (și nu numai).

În finalul acestei prefețe, autorul își exprimă gratitudinea față de soția sa, Maria Vințan, și față de fiul său, Radu Vințan, pentru sprijinul generos acordat pe parcursul elaborării acestei cărți și nu numai.

22 iulie 2016, Sibiu

Lucian N. Vințan




## CUPRINS

---

<b>0. O SCURTĂ ISTORIE A SISTEMELOR DE CALCUL .....</b>	<b>16</b>
<b>1. O INTRODUCERE ÎN FILOSOFIA MICROSISTEMELOR DE CALCUL ....</b>	<b>25</b>
1.1 SCHEMA BLOC A UNUI MICROSISTEM. ROLUL BLOCURILOR COMPONENTE, FUNCȚIONARE DE ANSAMBLU .....	25
1.2 MODURI DE LUCRU ÎNTRE MICROPROCESOR ȘI INTERFEȚELE I/O ..	40
1.2.1 MODUL DE LUCRU PRIN INTEROGARE (“POLLING”) .....	40
1.2.2 MODUL DE LUCRU PRIN ÎNTRERUPERI HARDWARE .....	42
1.2.3 MODUL DE LUCRU PRIN TRANSFER DMA (DIRECT MEMORY ACCESS) .....	46
<b>2. ARHITECTURA SISTEMULUI IERARHIZAT DE MEMORIE .....</b>	<b>53</b>
2.1 MEMORII CACHE.....	53
2.2 MEMORIA VIRTUALĂ .....	89
<b>3. PROCESOARE PIPELINE SCALARE CU SET OPTIMIZAT DE INSTRUCȚIUNI .....</b>	<b>99</b>
3.1. MODELUL RISC. GENEZĂ ȘI CARACTERISTICI GENERALE .....	99
3.2. SET DE INSTRUCȚIUNI. REGIȘTRI INTERNI LA MODELUL ARHITECTURAL RISC .....	101
3.2.1. DUALITATEA ARHITECTURĂ – APLICAȚIE: IMPLEMENTAREA GESTIUNII STIVELOR DE DATE ASOCIATE FUNCȚIILOR C.....	109
3.2.2. IMPLEMENTAREA RECURSIVITĂȚII .....	114
3.3. ARHITECTURA SISTEMULUI DE MEMORIE LA PROCESOARELE RISC .....	116
3.4. PROCESAREA PIPELINE ÎN CADRUL PROCESOARELOR SCALARE ..	118
3.4.1. DEFINIREA CONCEPTULUI DE ARHITECTURĂ PIPELINE SCALARĂ .....	119
3.4.2. PRINCIPIUL DE PROCESARE ÎNTR-UN PROCESOR PIPELINE .....	122
3.4.3. STRUCTURA PRINCIPALĂ A UNUI PROCESOR RISC .....	125
3.4.4. PROBLEMA HAZARDURILOR ÎN PROCESOARELE RISC.....	129
3.4.4.1. HAZARDURI STRUCTURALE (HS): PROBLEME	

IMPLICATE ȘI SOLUȚII .....	130
3.4.4.2. HAZARDURI DE DATE: DEFINIRE, CLASIFICARE, SOLUȚII DE EVITARE A EFECTELOR DEFAVORABILE.....	135
3.4.4.3. HAZARDURI DE RAMIFICAȚIE (HR): PROBLEME IMPLICATE ȘI SOLUȚII .....	143
3.4.5. PROBLEMA EXCEPȚIILOR ÎN PROCESOARELE RISC .....	197
3.4.6. AMBIGUITATEA REFERINȚELOR LA MEMORIE .....	200
3.4.7. EXECUȚIA CONDIȚIONATĂ ȘI SPECULATIVĂ.....	203
<b>4. PROCESOARE CU EXECUȚII MULTIPLE ALE INSTRUCȚIUNILOR. MULTIPROCESOARE .....</b>	<b>207</b>
4.1. CONSIDERAȚII GENERALE. PROCESOARE SUPERSCALARE ȘI VLIW (EPIC).....	207
4.2. MODELE DE PROCESARE ÎN ARHITECTURILE SUPERSCALARE .....	228
4.3. ARHITECTURA LUI R. TOMASULO .....	230
4.4. O ARHITECTURĂ REPREZENTATIVĂ DE PROCESOR SUPERSCALAR.....	239
4.5. PROBLEME SPECIFICE INSTRUCȚIUNILOR DE RAMIFICAȚIE ÎN ARHITECTURILE CU EXECUȚII MULTIPLE.....	258
4.5.b MICROPROCESOARE MULTI-MICROTHREAD .....	260
4.6. OPTIMIZAREA BASIC-BLOCK-URILOR ÎN ARHITECTURILE MEM....	275
4.6.1. PARTIȚIONAREA UNUI PROGRAM ÎN "BASIC-BLOCK"-URI.....	279
4.6.2. CONSTRUCȚIA GRAFULUI DEPENDENȚELOR DE DATE ASOCIAT.....	280
4.6.3. CONCEPTUL CĂII CRITICE.....	283
4.6.4. ALGORITMUL "LIST SCHEDULING" (LS).....	284
4.7. PROBLEMA OPTIMIZĂRII GLOBALE ÎN CADRUL PROCESOARELOR MEM .....	288
4.7.1. TEHNICA "TRACE SCHEDULING" (TS) .....	289
4.8. OPTIMIZAREA BUCLELOR DE PROGRAM.....	300
4.8.1. TEHNICA "LOOP UNROLLING" .....	301
4.8.2. TEHNICA "SOFTWARE PIPELINING" .....	304
4.9. ARHITECTURI CU TRANSPORT DECLANȘAT.....	307
4.10. EXTENSII ALE ARHITECTURILOR MEM PE BAZĂ DE	

REUTILIZARE ȘI PREDICȚIE A INSTRUCȚIUNILOR.....	311
4.11. PROCESAREA VECTORIALĂ (SIMD).....	349
4.12. SISTEME MULTIPROCESOR (MIMD).....	363
4.13. OPTIMIZAREA MULTI-OBIECTIV A SISTEMELOR DE CALCUL .....	472
<b>5. PROBLEME PROPUSE SPRE REZOLVARE.....</b>	<b>498</b>
 <b>BIBLIOGRAFIE SELECTIVĂ .....</b>	<b>530</b>
<b>GLOSAR DE TERMENI TEHNICI UTILIZAȚI .....</b>	<b>536</b>