

PLANIFICAREA LUCRĂRILOR DE LABORATOR LA DISCIPLINA “SIMULAREA ȘI OPTIMIZAREA SISTEMELOR DE CALCUL *MASTER_AAIE*”

Obiectiv: - Deprinderea abilităților de utilizare și exploatare a simulatoarelor software dedicate **arhitecturilor de calcul superscalare** și înțelegerea și aplicarea unor **tehnici de optimizare multi-obiectiv** asupra acestora.

L1. INTRODUCERE. Explicite noțiuni: simulator, metodologii de simulare. Descriere etape de simulare, comparare și determinare efectivă ale unei arhitecturi optime, pornind de la sursa HLL (High Level Languages) a programelor de test și până la implementarea hardware a arhitecturii (Fig. 1, Cap.0 SOAC). Simulatoarele determina: *performanta de procesare + model de putere (CACTI, WATCH) → estimare putere /energie consumata + configuratie termica && configuratie microarhitecturala spatia (Quilt, HotSpot) → hotspots → identificare DTM (manager termal dinamic) - (Fig. 1, Art. JSA, Fig. 3&&4 Art. thermal2), Workflow for finding a near optimal microarchitecture from thermal viewpoint.*

Prezentare generală a conținutului structurat pe cele **3 seminarii și 6 laboratoare.**

- Simularea și optimizarea arhitecturilor RISC scalare (DLX) folosind simulatorul *execution-driven WinDLX* și respectiv a unei mașini cu execuție multiplă (procesor VLIW) folosind simulatorul **VLIW-DLX**. Utilizarea simulatorului *hibrid SATSim* aferent arhitecturii PowerPC pentru vizualizarea aspectelor arhitecturale specifice procesoarelor RISC superscalare (stații de rezervare, unități funcționale de execuție, buffer de reordonare și buffer de redenumire). Varianta îmbunătățită **PSATSim** evidențiază suplimentar efectul negativ al predicției greșite a instrucțiunilor de ramificație atât asupra vitezei de execuție a procesorului cât și asupra energiei consumate. Simularea unor arhitecturi de tip multithreading simultan (SMT) și grefarea tehnicilor de reutilizare dinamică și predicție a valorilor pe arhitecturi *SMT* (simulatorul *m-sim2*).
- Explorare manuală vs. automată a spațiului de proiectare pentru optimizarea multi-obiectiv a unei arhitecturi de tip SLVP. Algoritmi genetici: schema bloc, componente, reprezentare indivizi, operatori de variație, supraviețuirea urmașilor în noua populație – generațională vs. elitistă etc. Metode de căutare / optimizare, indicatori de calitate. Metode de optimizare multi-obiectiv: non-Pareto, Pareto, bioinspirate.

L2. a) UTILIZAREA SIMULATORULUI WINDLX: configurarea WinDLX, încărcarea programelor de test, simularea benchmark-urilor, sistemul de ferestre. Investigații arhitecturale utilizând simulatorul DLX (<http://webspaces.ulbsibiu.ro/adrian.florea/html/simulatoare/dlx.zip>): benchmark - urile *Fact.s, Invers.s*, rutina *Input.s*, probleme propuse spre rezolvare. Evidențierea efectului defavorabil al **hazardurilor RAW** în cadrul arhitecturilor **pipeline scalare** asupra performanței de procesare. Modificarea programului de test *fact.s* și simularea **hazardurilor structurale** și a **conflictelor de nume (WAW)**. Vizualizarea tehnicii de **forwarding** – câștigul de performanță obținut.

b) UTILIZAREA SIMULATORULUI VLIW-DLX: ilustrarea principiilor fundamentale ale procesării VLIW (very long instruction word). Rolul software-lui 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 (număr/latențe). Optimizări software în 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 translatează un vector de numere reale simplă precizie.

L3. a) SATSIM: SIMULAREA ARHITECTURILOR SUPERSCALARE FOLOSIND ANIMAȚIE INTERACTIVĂ

(<http://webspacespace.ulbsibiu.ro/adrian.florea/html/simulatoare/SATSimG.ZIP>). Înțelegerea conceptelor legate de procesarea *pipeline* a instrucțiunilor, execuția *out of order*, impactul negativ asupra performanței al predicției greșite a branch-urilor și al miss-urilor în cache. Evidențierea, prin simulare la nivel de execuție, a aspectelor arhitecturale specifice procesoarelor RISC superscalare (stații de rezervare, unități funcționale de execuție, buffer de reordonare și buffer de redenumire).

b) PSATSIM: INSTRUMENT SOFTWARE DE EVALUARE A PERFORMANȚEI DE PROCESARE ȘI A CONSUMULUI DE PUTERE ÎN MICROARHITECTURILE SUPERSCALARE.

b.1. Modelarea consumului de putere (CACTI, WATTCH). Parametrii tehnologici, clasificarea consumului de putere în funcție de tipul resursei arhitecturale (*structuri matriceale* – memoria cache, decodificatoare de adresă, setul de regiștrii generali, predictorul de salturi; *memorii adresabile după conținut* – asociative – stațiile de rezervare, logica de reactivare a instrucțiunilor situate în buffer-ul de reordonare în așteptarea efectuării fazei de commit; *circuite logice combinaționale* – implementarea unităților funcționale de execuție (ALU), logica de verificare a dependențelor de date, selecția instrucțiunilor spre stațiile de rezervare și unitățile funcționale). **Influența caracteristicilor arhitecturale specifice procesoarelor RISC superscalare asupra consumului de putere și respectiv asupra performanței.**

b.2. Predictorul de salturi – una din principalele cauze care determină consum ridicat de putere în arhitecturi. **Evidențierea efectului negativ al predicției greșite a instrucțiunilor de ramificație** (parametrizarea acurateții de predicție) **atât asupra vitezei de execuție a procesorului cât și asupra energiei consumate.**

L4. DEPĂȘIREA UNOR BARIERE ARHITECTURALE ALE PARADIGMEI ACTUALE A PROCESOARELOR SUPERSCALARE. Dezvoltarea unei arhitecturi superscalare îmbogățită cu un mecanism complex de anticipare selectivă a valorilor instrucțiunilor cu latență ridicată de execuție, inclusiv pentru a reduce impactul negativ al salturilor nepolarizate asupra performanței de procesare. **SCHEMĂ DE REUTILIZARE** pentru instrucțiunile de înmulțire și împărțire (DIV/MUL) respectiv un **PREDICTOR DE VALORI (LVP)** pentru instrucțiunile *Load* critice. **Cuantificarea impactului mecanismului de anticipare selectivă a valorilor instrucțiunilor cu latență ridicată într-o arhitectură de tip *multithreading* simultan (SMT), care implică multiplicări ale *buffer*-elor de reutilizare și ale predictoarelor de valori, pentru fiecare fir în parte.** [Inițiere SO Linux, – simulatorul *m-sim2*.](#)

L5. [Considerații asupra performanței de procesare, energiei și disipației termice în arhitecturi superscalare.](#) Simulări folosind *hotspot* având în vedere măsurarea consumului de putere la fiecare 500 kilocicli (*trace*-uri de putere) pe șase *benchmark*-uri de test pentru fiecare configurație aleasă; Realizarea *trace*-urilor de temperatură și a hărților termice pe baza *trace*-urilor de puteri și ale configurațiilor spațiale (*floorplan*) realizate anterior. Integrarea unui predictor de *branch*-uri neural cu perceptron în mediul de simulare M-SIM2; Realizarea unor statistici privind numărul de instrucțiuni care se decodifică versus cele care fac *commit*; Implementarea unui mecanism de *fetch gating* în simulatorul [M-SIM2](#).

L6. SEMINAR DE APLICAȚII. (I) Reprezentarea informației. Sisteme de numerație. Conversii de valori între sisteme de numerație. Calculul dimensiunii spațiului de căutare a parametrilor unei arhitecturi – PSATSim, m-sim2 ([numărul de configurații posibile](#)). 50 parameters, 8 values/parameter → 8^{50} configurations. Exemplificare – plaja diferită de valori (număr de cazuri posibile diferit pentru fiecare parametru). Generalizare.

L7. Laborator dedicat [metodelor euristice de căutare](#) pentru evaluarea unui număr redus de configurații din spațiul imens existent. Aplicarea metodelor *Hill Climbing* și *Simulated Annealing* pe PSATSim (folosind 5 parametri arhitecturali – FR, RenB, ROB, UE, SR). Exemplificare grafică după rularea [FOCAP tool](#) (hc_sim_opt.csv).

L8. SEMINAR DE APLICAȚII. (II) Metafora [calculului evolutiv](#): selecție naturală și moștenire genetică. Schema bloc, componentele principale și operatorii unui [algoritm genetic](#). [Reprezentarea indivizilor \(binară\)](#) / Evaluation / Population / Parent Selection / Recombination / Mutation / Survivor Selection / Termination. Aplicație – Maximul unei funcții, problema rucsacului. Analiza cod sursă [simpleEA.c](#)

L9. SEMINAR DE APLICAȚII. (III) Reprezentarea indivizilor (int, float, permutare). Exemplificare TSP, Queens. Operatori de variație specifici algoritmilor genetici a căror reprezentare este de tip permutare: mutație (Insert, Swap, Inversion, Scramble), crossover (Order 1, Partially matched - PMX, Cycle crossover). Aplicație – Influența lungimii de reprezentare asupra convergenței soluției într-o problemă de optimizare a unei funcții cu valori reale. Reprezentarea parametrilor arhitecturali PSATSim, m-sim2 - *msim2lvpt.xml*, *PSATSimSolutionType.cs*, *PSATSimOptimizationProblem.cs*.

L10. a) Metode de optimizare multi-obiectiv. Metrici de evaluare a calității indivizilor (PF, HV, Coverage). După rularea indivizilor selectati prin FADSE se determină IPC și Energie se afisează PF și folosind sursa HyperVolume.cs – se determină HV.

b) Influența regulilor Fuzzy asupra vitezei de convergență a algoritmilor (repetarea metodologiei de la punctul a) considerând *results_msim_lvpt_fuzzy*).

L11. Metrici de evaluare a calitatii indivizilor (PF, HV, Coverage). Rulare PSATSim un număr de generații și generat HV și PF. Comparatie între metodele de optimizare: NSGA-II, SMPSO, VEGA (SPEA-2).

L12 ÷ L13. PROIECT – ANALIZĂ ȘI IMPLEMENTARE (dacă e cazul).

L14. EVALUARE PRACTICĂ PRIVIND MATERIA STUDIATĂ LA LABORATOR / PROIECT.