PLANIFICAREA LUCRĂRILOR DE LABORATOR LA DISCIPLINA "*MICROARHITECTURI*" – MASTER ICAI II

L1. INTRODUCERE. Explicitare noţiuni: microarhitectură, ISA, cache, simulator, metodologii de simulare. Avantajele utilizării simulatoarelor în studiul microarhitecturilor de procesare pentru determinarea efectivă a celei optime din punct de vedere cost / performanţă, dar şi pentru înţelegerea principiilor fundamentale specifice arhitecturilor RISC de calcul, organizării memoriei, prin prisma unor aplicaţii scrise în limbaje de asamblare. Descriere etape de simulare, comparare şi determinare efectivă ale unei arhitecturi optime, pornind de la sursa HLL (High Level Languages) a programelor de test şi 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 spatiala* (**Quilt, HotSpot**) ➔ *hotspots* ➔ *identificare DTM* (manager termal dinamic) - *(Fig. 1, Art. JSA, Fig. 3&&4 Art. thermal2)*.

Prezentare generală a conţinutului laboratorului structurat pe simularea şi optimizarea arhitecturilor RISC scalare, superscalare & VLIW, SMT. Descriere succintă a celor **14** simulatoare dintre care **7** de tip *execution driven* aferente arhitecturilor MIPS (PCspim-cache), DLX (WinDLX şi VLIW-DLX) şi Hatfield (HSA), două simulatoare *hibride* – SATSim şi PSATSim aferente arhitecturii PowerPC, şi o suită de **5** simulatoare de tip *trace driven* – de la cele mai simple la cele mai complexe: simulatoare care evidenţiază strategia de scriere în cache-uri dar şi eficienţa unui Selective Victim Cache, simulatoare care descriu funcţionarea structurilor hardware de predicţie (BTB, GAg, Markoviene, de tip Perceptron simplu şi multistrat).

L2. a) ARHITECTURA MICROPROCESOARELOR MIPS R2000/R3000: schema bloc, regiştrii procesorului MIPS R2000, modurile de adresare, sintaxa asamblor, utilizarea memoriei, formatul instrucţiunilor şi setul de instrucţiuni al procesorului MIPS R2000.

b) EVIDENŢIEREA CONCEPTELOR LEGATE DE CACHE-URI – MODUL DE ORGANIZARE, REGULILE DE MAPARE, ALGORITMII DE ÎNLOCUIRE A BLOCURILOR CONFLICTUALE, STRATEGIA DE SCRIERE – FOLOSIND SIMULATORUL PCSPIM-CACHE.

c) Se consideră următorul programul de test **g_cache.s** care realizează în paralel suma a câte 8 elemente – 1,1,1,1,2,2,2,2 stocate în memorie la adresa 0x10000480 respectiv a altor 8 elemente 3,3,3,3,4,4,4,4 de la adresa 0x10000500. Să se ruleze **pas cu pas** secvenţa de cod şi să se vizualizeze dinamic modificările de stare ale memorie cache şi realizaţi o statistică privind **impactul asupra ratei de hit în cache-uri** a următorilor factori: – dimensiunea blocului de date, gradul de asociativitate, algoritmul de înlocuire, strategia de scriere. Implementarea altor programe de test.

d) Se consideră următorul programul de test **cache_loop.s** care realizează în paralel suma elementelor unei matrici Array_A[10,20] mai intai toate elementele unei coloane. Se aplica metoda *loop interchange* (insumarea intai a tuturor elementelor de pe o linie – cod sursa **cache_loop_i.s**) si se vizualizeaza comparativ rata de hit in cache.

L3. a) UTILIZAREA SIMULATORULUI WINDLX: configurarea WinDLX, încărcarea programelor de test, simularea benchmark-urilor, sistemul de ferestre. Investigaţii arhitecturale utilizând simulatorul DLX: benchmark - urile *Fact.s*, *Invers.s,* rutina *Input.s*, probleme propuse spre rezolvare. Evidenţierea efectului defavorabil al **hazardurilor RAW** în cadrul arhitecturilor **pipeline scalare** asupra performanţei de procesare. Modificarea programului de test fact.s şi simularea **hazardurilor structurale** şi a **conflictelor de nume (WAW)**. Vizualizarea tehnicii de **forwarding** – câştigul de performanţă obţinut.

b) UTILIZAREA SIMULATORULUI VLIW-DLX: ilustrarea principiilor fundamentale ale procesării VLIW (very long instruction word). Rolul software-lui în detecţia şi eliminarea hazardurilor RAW. Relaţia dintre instrucţiunea multiplă şi instrucţiunile RISC primitive şi independente, care vor fi alocate unităţilor de execuţie în conformitate strictă cu poziţia lor în instrucţiunea multiplă (număr/latenţe). Optimizări software in procesoarele VLIW (loop unrolling, software pipelining). Avantaje / Dezavantaje faţă de procesoarele superscalare. Aplicabilitate – sisteme dedicate (procesoare de semnal, procesoare multimedia). Aplicaţie rezolvată – execuţia pe un procesor VLIW a unui program care înmulţeşte două matrici pătratice.

L4. *SATSIM*: SIMULAREA ARHITECTURILOR SUPERSCALARE FOLOSIND ANIMAŢIE INTERACTIVĂ. Înţelegerea conceptelor legate de procesarea *pipeline* a instrucţiunilor, execuţia *out of order*, impactul negativ asupra performanţei al predicţiei greşite a branch-urilor şi al miss-urilor în cache. Evidenţierea, prin simulare la nivel de execuţie, a aspectelor arhitecturale specifice procesoarelor RISC superscalare (staţii de rezervare, unităţi funcţionale de execuţie, buffer de reordonare şi buffer de redenumire).

L5. *PSATSIM*: INSTRUMENT SOFTWARE DE EVALUARE A PERFORMANŢEI DE PROCESARE ŞI A CONSUMULUI DE PUTERE ÎN MICROARHITECTURILE SUPERSCALARE.

a) Modelarea consumului de putere (CACTI, WATTCH). Parametrii tehnologici, clasificarea consumului de putere în funcţie de tipul resursei arhitecturale (*structuri matriceale* – memoria cache, decodificatoare de adresă, setul de regiştrii generali, predictorul de salturi; *memorii adresabile după conţinut* – asociative – staţiile de rezervare, logica de reactivare a instrucţiunilor situate în buffer-ul de reordonare în aşteptarea efectuării fazei de commit; *circuite logice combinaţionale* – implementarea unităţilor funcţionale de execuţie (ALU), logica de verificare a dependenţelor de date, selecţia instrucţiunilor spre staţiile de rezervare şi unităţile funcţionale). **Influenţa caracteristicilor arhitecturale specifice procesoarelor RISC superscalare asupra consumului de putere şi respectiv asupra performanţei**.

b) Predictorul de salturi – una din principalele cauze care determină consum ridicat de putere în arhitecturi. **Evidenţierea efectului negativ al predicţiei greşite a instrucţiunilor de ramificaţie** (parametrizarea acurateţii de predicţie) **atât asupra vitezei de execuţie a procesorului cât şi asupra energiei consumate**.

L6. SEMINAR – REZOLVAREA UNOR PROBLEME DE EXAMEN PROPUSE.

L7. a) SIMULAREA UNEI ARHITECTURI SUPERSCALARE PARAMETRIZABILE (HSA). Ghid de utilizare. Explicitare opţiuni (Roger Simulator). a**1)** Generaţi în urma simulării de tip *execution driven* fişierele trace (*.trc) aferente celor 8 benchmark-uri Stanford care să cuprindă doar tipurile de instrucţiuni: de salt (*Branch*) şi cele cu referire la memorie (*Load/Store*). **a2)** Exemplificaţi grafic rata de procesare obţinută pe cele 8 benchmark-uri în funcţie de tipul modelului arhitectural. **IR = f(**tip_model**)** - *model minimal* vs. *model maximal*. Vizualizaţi gradele de utilizare ale resurselor hardware pe cele 2 modele. Interpretări. Concluzii.

b) SIMULAREA INTERFEŢEI PROCESOR-CACHE PENTRU O ARHITECTURĂ RISC SUPERSCALARĂ PARAMETRIZABILĂ. Tipuri arhitecturale utilizate în proiectare: **mapare directă**, **set asociativ**, **complet asociativ**. Modalităţi de scriere: **write back**/**write through**. Avantaje şi dezavantaje. Descrierea parametrilor simulatorului. Simulare şi investigaţii cantitative. Avantaje introduse prin **DWB** (preia sarcina procesorului de a scrie datele în cache-ul de date ⇒ bypassing aplicat instrucţiunilor Load/Store din DWB). Simulare.

L8. SELECTIVE VICTIM CACHE

a) Seminar: Prezentare teoretică a conceptului de (Selective) Victim Cache. Algoritmi de predicţie. Metrici de evaluare a performanţei.

b) Lucrare practică: Detalii de implementare. Metodologia de simulare. Probleme propuse spre rezolvare.

L9. SCHEME DE PREDICŢIE IMPLEMENTATE ÎN PROCESOARELE SUPERSCALARE AVANSATE.

- **PREDICTOR DE SALTURI DE TIP BTB** care lucrează distribuit în reţea. Metodologia de simulare.
- **PREDICTOR DE SALTURI ADAPTIV, CORELAT PE DOUĂ NIVELURI (GAg)**
- **DETECTIA SI IZOLAREA SALTURILOR *DIFICIL DE PREZIS INTR-UN ANUMIT CONTEXT*. PREDICŢIA ACESTORA PRIN INTERMEDIUL UNOR SCHEME NEURALE** DE TIP ***PERCEPTRON SIMPLU*** (cu un singur strat) şi ***FAST PATH-BASED*.** Avantaje şi dezavantaje ale predicţiei cu perceptroane simple a *branch*-urilor, în raport cu schemele clasice. Descrierea simulatorului **ABPS**. Investigaţii cantitative privind acurateţea predicţiei şi costul hardware. Modelare si implementare sub platforma industrială standardizată CBP – campionatul mondial de predicţia salturilor.

L10. SEMINAR – REZOLVAREA UNOR PROBLEME DE EXAMEN PROPUSE.

L11. a) DEZVOLTAREA DE SIMULATOARE SUB MEDIUL *SIMPLESCALAR* UTILIZÂND *BENCHMARK* -URILE SPEC. Ce este "*SimpleScalar Tool Set*" ? Avantajele/dezavantajele utilizării "*SimpleScalar Tool Set*". Ce este "*GNU*" ? Utilitare GNU. Arhitectura SimpleScalar. Simulatoarele setului de instrumente SimpleScalar 3.0. Ghid de utilizare. Extensii. Evaluarea performanţelor.

b) EXTINDEREA MEDIULUI *SIMPLESCALAR* CU MODULUL DE MĂSURARE A GRADELOR DE LOCALITATE (REGIŞTRI / INSTRUCŢIUNI). PREDICŢIA VALORILOR FOLOSIND SCHEMA DE TIP *LAST-VALUE*.

L12. DEPĂŞIREA UNOR BARIERE ARHITECTURALE ALE PARADIGMEI ACTUALE A PROCESOARELOR SUPERSCALARE. Dezvoltarea unei arhitecturi superscalare îmbogăţită cu un mecanism complex de anticipare selectivă a valorilor instrucţiunilor cu latenţă ridicată de execuţie, inclusiv pentru a reduce impactul negativ al salturilor nepolarizate asupra performanţei de procesare. **SCHEMĂ DE REUTILIZARE** pentru instrucţiunile de înmulţire şi împărţire (DIV/MUL) respectiv un **PREDICTOR DE VALORI (LVP)** pentru instrucţiunile *Load* critice. **CUANTIFICAREA IMPACTULUI MECANISMULUI DE ANTICIPARE SELECTIVĂ A VALORILOR INSTRUCŢIUNILOR CU LATENŢĂ RIDICATĂ ÎNTR-O ARHITECTURĂ DE TIP *MULTITHREADING* SIMULTAN (SMT)**, care implică multiplicări ale *buffer*-elor de reutilizare şi ale predictoarelor de valori, pentru fiecare fir în parte. Simulatorul m-sim2.

L13. SIMULAREA EXECUTIEI UNOR APLICATII PARALELE PE O ARHITECTURIA MULTICORE. INSTALAREA SI CONFIGURAREA SIMULATORULUI *MULTI2SIM* PE S.O. LINUX UBUNTU / FEDORA. Familiarizarea cu comenzi / tools-uri în Linux. Suitele de benchmarkuri SPLASH-2 si PARSEC. Simulari cantitative care sa ilustreze performanta comparativa a arhitecturilor ***superscalare*** vs. ***SMT*** vs. ***multicore***.