

PLANIFICAREA LUCRĂRILOR DE LABORATOR LA DISCIPLINA “SIMULAREA ȘI OPTIMIZAREA ARHITECTURILOR DE CALCUL”

L1. INTRODUCERE. Explicare 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 spatiala (Quilt, HotSpot) → hotspots → identificare DTM (manager termal dinamic) - (Fig. 1, Art. JSA, Fig. 3&&4 Art. thermal2)*. Prezentare generală a conținutului laboratorului structurat pe cele 9 simulatoare de arhitecturi RISC superscalare (un simulator de tip *execution driven* aferent arhitecturii Hatfield – HSA și o suită de 8 simulatoare de tip *trace driven* – de la cele mai simple la cele mai complexe: simulatoare care evidențiază strategia de scriere în cache-uri dar și eficiența unui Selective Victim Cache, simulator dedicat analizei coerenței în sistemul ierarhic de memorie din arhitecturile de tip multiprocesor cu memorie partajată folosind o rețea de interconectare de tip bus (**SMPCache**), simulatoare care descriu funcționarea structurilor hardware de predicție (**BTB, GAg, Markoviene**, de tip **Perceptron** simplu, *fast path-based* și multistrat). [Repartizare teme de proiect](#) și [Conținut dosar proiect](#).

L2. SIMULAREA UNEI ARHITECTURI SUPERSCALARE PARAMETRIZABILE (HSA)

- a) **Lucrare practică**: Simulator pentru arhitectura superscalară Hatfield. Ghid de utilizare. Explicare opțiuni (Roger Simulator).
- b) **Lucrare practică**: Investigații arhitecturale utilizând simulatorul HSA:
 - b1) Generați în urma simulării de tip *execution driven* fișierele trace (*.trc) aferente celor 8 benchmark-uri Stanford care să cuprindă doar tipurile de instrucțiuni: de salt (*Branch*) și cele cu referire la memorie (*Load/Store*).
 - b2) Exemplificați grafic rata de procesare obținută pe cele 8 benchmark-uri în funcție de tipul modelului arhitectural. $IR = f(\text{tip_model})$ - *model minimal* vs. *model maximal*. Vizualizați gradele de utilizare ale resurselor hardware pe cele 2 modele. Interpretări. Concluzii.

L3. SIMULAREA INTERFEȚEI PROCESOR-CACHE PENTRU O ARHITECTURĂ RISC SUPERSCALARĂ PARAMETRIZABILĂ (sim-cache, traces)

- a) Tipuri arhitecturale utilizate în proiectare: **mapare directă, set asociativ, complet asociativ**.
- b) Modalități de scriere: **write back/write through**. Avantaje și dezavantaje. Descrierea parametrilor simulatorului. Simulare și investigații cantitative.
- c) Avantaje introduse prin **DWB** (preia sarcina procesorului de a scrie datele în cache-ul de date ⇒ bypassing aplicat instrucțiunilor Load/Store din DWB). Simulare.

L4. SELECTIVE VICTIM CACHE

- a) **Seminar**: Prezentare teoretică a conceptului de (Selective) Victim Cache. Algoritmi de predicție. Metrici de evaluare a performanței.

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

L5. SIMULAREA SISTEMULUI IERARHIC DE MEMORIE ÎN ARHITECTURI DE TIP MULTIPROCESOR CU MEMORIE PARTAJATĂ.

- Utilizarea simulatorului trace-driven **SMPCache** pentru analiza coerenței în sistemul ierarhic de memorie din arhitecturile de tip multiprocesor cu memorie partajată folosind o rețea de interconectare de tip bus. [Aprofundarea conceptelor legate de cache-uri](#) – modul de organizare, regulile de mapare, algoritmi de înlocuire a blocurilor conflictuale, strategia de scriere, protocoale de coerență, etc.

L6. IMPLEMENTAREA SCHEMELOR CLASICE DE PREDICȚIE ÎN PROCESOARELE SUPERSCALARE AVANSATE (I).

- **PREDICTOR DE SALTURI DE TIP BTB** care lucrează distribuit în rețea. Metodologia de simulare.

L7. IMPLEMENTAREA SCHEMELOR CLASICE DE PREDICȚIE ÎN PROCESOARELE SUPERSCALARE AVANSATE (II).

- **PREDICTOR DE SALTURI ADAPTIV, CORELAT PE DOUĂ NIVELURI (GAg).** Vizualizarea conținutului structurii de predicție în fiecare ciclu de execuție.

L8. OPTIMIZAREA SCHEMELOR DE PREDICȚIE PENTRU RAMIFICAȚIILE DE PROGRAM ÎN PROCESOARELE SUPERSCALARE AVANSATE (I).

- **PREDICTOR MARKOVIAN DE RAMIFICAȚII PROGRAM (Predictor contextual de tip PPM complet).** Studiul fezabilității unui predictor simplificat bazat pe context compus dintr-un predictor Markov de ordin maxim și unul de ordin 0.

L9. OPTIMIZAREA SCHEMELOR DE PREDICȚIE PENTRU RAMIFICAȚIILE DE PROGRAM ÎN PROCESOARELE SUPERSCALARE AVANSATE (II).

- **PREDICTOR NEURONAL DE RAMIFICAȚII PROGRAM (Perceptronul Multistrat).** Memento privind rețelele neuronale. Metodologia de simulare. Descriere simulator. Investigații cantitative privind acuratețea predicției.

L10. DETECTIA SI IZOLAREA SALTURILOR DIFICIL DE PREZIS INTR-UN ANUMIT CONTEXT. PREDICȚIA ACESTORA PRIN INTERMEDIUL UNOR SCHEME NEURALE DE TIP *PERCEPTRON SIMPLU* (cu un singur strat) și *FAST PATH-BASED*. Avantaje și dezavantaje ale predicției cu perceptrone simple a branchurilor, în raport cu schemele clasice. Descriere simulator. Investigații cantitative privind acuratețea predicției și costul hardware. Modelare și implementare sub platforma industrială standardizată CBP – campionatul mondial de predicția salturilor. Benchmark-uri Stanford, SPEC, Java (CBP2).

L11 ÷ L13. PROIECT – ANALIZĂ ȘI IMPLEMENTARE.

L14. EVALUARE PROIECT.