

## Rezumatul lucrării

În această lucrare de licență se urmărește proiectarea, implementarea și simularea unui microprocesor. Pentru acest lucru, vor fi aprofundate în continuare, atât proiectarea lui hardware și simularea lui, folosind un design schematic cu ajutorul aplicației Cadence OrCAD (OrCAD Capture), cât și descrierea lui hardware în cod Verilog. Testarea și simularea microprocesorului descris în limbajul Verilog se va face cu ajutorul mediului de simulare ModelSim, ce aparține companiei Mentor Graphics. Ambele aplicații, atât Cadence OrCAD, cât și ModelSim vor rula pe sistemul de operare Microsoft Windows. Microprocesorul simulat în studiul de caz se bazează pe arhitectura *Reduced Instruction Set Computing* (eng.), în română *computer cu set redus de instrucțiuni*, sau pe scurt RISC, pe 8 biți. Microprocesoarele RISC sunt concepute pentru a avea un mic set de instrucțiuni care se execută în cicluri scurte de *clock* (eng.), cu un număr mic de cicluri pe instrucțiuni. Microprocesoarele RISC sunt optimizate pentru a realiza *pipelining* (eng.) eficient a fluxurilor lor de instrucțiuni. În acest proiect se va proiecta o mașină RISC simplă. Practic, lucrarea se rezumă la implementarea aceluiași microprocesor cu două aplicații diferite: una pentru design hardware și una pentru scrierea unui cod Verilog aferent microprocesorului, făcându-se, în paralel și o comparație între cele două implementări, pentru ca mai apoi, la final, să se stabilească, care dintre cele două implementări este mai optimă. Pentru varianta Verilog, testarea microprocesorului RISC se va face printr-un algoritm de înmulțire a două numere întregi pozitive prin adunări repetate. În implementarea microprocesorului se vor folosi și câteva instrucțiuni ca în asamblare, de exemplu: mov, jump, add, xor.

Pe parcursul lucrării vor fi evidențiate anumite caracteristici specifice proiectării RISC, principii de proiectare, precum și o comparație din mai multe puncte de vedere cu cealaltă mare categorie de modele de procesoare, CISC și evoluția acestora. Pentru a înțelege de ce s-a introdus arhitectura microprocesorului RISC, aceasta va fi comparată cu arhitectura CISC și se vor evidenția lipsurile de eficiență ale procesoarelor cu set complex de instrucțiuni. De asemenea se vor exemplifica și câteva procesoare cunoscute care fac parte din aceste diverse categorii de arhitecturi.

Pentru proiectarea aplicației se va realiza o schemă hardware care va conține circuitul de prelucrare, pe care se vor putea distinge componentele hardware folosite: unitatea de memorie, unitatea de prelucrare, unitatea de control, semnalele și instrucțiunile folosite, precum și o altă schemă în care se va putea vizualiza arhitectura unității de procesare. În implementarea cu Cadence OrCAD, toate aceste componente vor fi implementate folosind bistabile *flip-flop* (eng.) și porți logice. Pentru implementarea cu ModelSim, pentru toate aceste componente vor fi scrise câteva module de descriere hardware în limbaj Verilog: modul pentru descrierea comportamentală a unității aritmetico-logice, modul pentru descrierea comportamentală a unității de control, modul pentru descrierea comportamentală a unei memorii, modul pentru descrierea comportamentală a unui multiplexor cu 3 canale, modul pentru descrierea comportamentală a unui multiplexor cu 5 canale, modul pentru descrierea ierarhică a unității de procesare, modul pentru descrierea comportamentală a contorului de program, modul pentru descrierea comportamentală a unui registru și modul pentru descrierea ierarhică a microprocesorului RISC. În aceste module se vor descrie: parametrii modulului respectiv, porturile de intrare-ieșire, declarațiile semnalelor și a variabilelor interne, corpul modulului etc.

Pentru testarea aplicației realizată în limbaj Verilog, va fi implementat un modul de verificare ce va testa funcționarea corectă a implementării RISC-ului prin algoritmul de înmulțire realizat prin adunări succesive. Algoritmul este unul simplu, iar organigrama funcțională precum și codul aferent acestuia sunt specificate în lucrare.

În final, se vor putea observa rezultatele obținute prin urmărirea formelor de undă atât pentru implementarea cu OrCAD, cât și pentru implementarea în cod Verilog, și se vor putea

trage diferite concluzii. De asemenea, pentru implementarea cu cod Verilog, pe lângă formele de undă care vor arăta semnalele interne ale microprocesorului, precum și funcționarea: memoriei, a unui registru, a contorului de program, a multiplexoarelor folosite și a unității aritmetico-logice, se va putea vizualiza și conținutul de memorie și rezultatul înmulțirii al algoritmului creat în fereastra “Memory” a mediului de simulare ModelSim, În fereastra “Memory” primul operand se va afla la adresa 255, al doilea operand se va afla la adresa 254, la adresa 253 se va găsi constanta 1 folosită pentru decrementare, iar rezultatul înmulțirii se va afla la adresa 252.