

Proiectarea unei extensii hardware pentru microprocesorul RISC-V, în vederea gestiunii dinamice a memoriei

Dragoș-Ștefan Perju

Rezumat

Proiectul de față își propune implementarea unui modul de gestiune dinamică a memoriei, integrat într-un sistem RISC-V. Arhitectura de set de instrucțiuni RISC-V fiind o noutate, se abordează problema menționată în ciuda dificultăților avute prin lipsa documentației complete și se reușește a se implementa alocatorul ca și extensie emulată la nivel software, dar și ca periferic pentru sinteza ansamblului pe platforma FPGA Digilent Arty Artix-7.

Alocatorul hardware implementat se inspiră din lucrarea *Dynamic Memory Management in Hardware* a autorilor Ingström și Daleby [1], prezentând o modalitate nouă de abordare a alocării de memorie – se caută blocuri libere în intervalele de timp în care ansamblul software ce rulează pe sistemul aferent nu lansează cereri de alocare. Astfel, se ajunge a avea un răspuns instant în multe dintre cazuri, proiectul de față studiind însă doar alocări și dealocări secvențiale – care reprezintă cel mai nefavorabil caz și care arată că viteza devine mai mică liniar cu gradul de ocupare a memoriei.

Modulul de alocare astfel creat folosește conceptul de hărți de biți, oglindind memoria cu care lucrează un proces software cu două astfel de hărți de biți: notând cu biți de 1 zonele ocupate de memorie, într-o hartă, și delimitând blocurile cu câte un bit de 1, pentru a ști fiecare bloc din acestea de unde începe, în cea de-a doua hartă. Se discută, în cadrul documentului, asupra componentelor combinaționale care li se servesc hărțile segmentat – segmentele numindu-se *vectors*, de lungime fixă de biți. Se află cum se poate lucra cu limbajul aferent implementării folosite RISC-V, denumit Chisel, încât să se genereze dinamic aceste componente combinaționale, care prelucrează la un tact câte un vector dintr-o hartă de biți.

În introducerea documentului se abordează în detaliu subiecte precum RISC vs. CISC, specificațiile RISC-V în sine și clasele mari de algoritmi dedicați pentru gestiunea dinamică de memorie. Se află că se folosesc algoritmi diferiți de la program la program, depinzând de cerințe, și că un sistem de operare complet nu are doar un singur alocator ce rulează la un moment dat – fiind alocatori dedicați pentru nucleu (kernel), pentru paginare și abia apoi pentru procesele software uzuale.

Implementarea arhitecturii RISC-V se manifestă sub forma proiectului Rocket Chip. Fiind tot o creație a arhitecților lui RISC-V, se folosește de protocolul specific RoCC pentru a „lega” modulul aproape de nucleul de procesare și a primi direct instrucțiuni de asamblare speciale pentru acele module de tip *extensie* a unui nucleu RISC-V.

Ansamblul astfel creat nu ajunge a fi sintetizabil însă, din cauza unor erori discutate. Kitul de dezvoltare pentru trecerea pe FPGA, denumit *SiFive Freedom E310*, folosește o versiune de Rocket Chip ce nu corespunde cu cea pe care s-a făcut extensia, producându-se astfel incompatibilități. Se recurge însă la ideea de a materializa modulul în FPGA sub formă de periferic mapat în memorie pentru sistemul RISC-V.

Acest lucru se realizează cu succes și se prezintă spre final explicații pe proiectarea componentelor și ansamblului modulului alocator hardware creat, a tehnologiilor inovative folosite, a rezultatelor obținute și se discută pe coduri sursă și afișaje de ecran.