

# ***Proiectare unei extensii pentru Microprocesorul RISC-V – comunicarea între task-uri***

Ștefan – Daniel Achirei

## **Rezumat**

Lucrarea și-a propus implementarea pe FPGA (Field Programmable Gate Array) a unei extensii suport pentru un procesor cu arhitectura RISC (Reduced Instruction Set Computer) care să realizeze comunicarea între task-uri.

Arhitectura folosește setul de instrucțiuni RISC-V dezvoltat în cadrul departamentului EECS (Electrical Engineering and Computer Science) al Universității Berkeley din California, Statele Unite ale Americii.

RISC-V (pronunțat „risk-five”) este o nouă arhitectură a unui set de instrucțiuni conceput inițial în scopul cercetării în domeniul arhitecturii calculatoarelor în mediul academic, acum devenind o arhitectura „deschisă” pentru implementări industriale.

Numele de „RISC-V” a fost ales să reprezinte a V-a generație majoră de ISA (Instruction Set Architecture) de tip RISC dezvoltată la Universitatea Berkeley. De asemenea utilizarea numărului roman V semnifică „variații” și „vectori” deoarece modelul RISC-V oferă suport pentru o gamă largă de arhitecturi, inclusiv numeroase acceleratoare de procesare paralelă a datelor, acesta fiind un scop explicit al ISA-ului.

În scopul creșterii performanțelor sistemului, comunicarea între task-uri se va realiza prin intermediul unei cozi de mesaje implementate hardware („hardware queue”), hardware-ul dedicat fiind mai rapid decât orice implementare software din nucleul sistemului de operare. Proiectarea s-a realizat în spiritul unui co-procesor specializat pentru un număr redus de operații/instrucțiuni specifice. Baza de procesor este proiectată încă de la început să suporte extensii, acest lucru s-a realizat prin rezervarea unui număr de 4 instrucțiuni *custom0/1/2/3* ce vor fi interpretate de procesor, însă executarea efectivă a instrucțiunii va fi realizată de către co-procesor, denumit în continuare accelerator sau extensie.

În aceasta lucrare s-a ales ca implementarea funcționalității cozii de mesaje să execute instrucțiunea rezervată „*custom0*”. Altfel spus în cazul instrucțiunilor cu codul de instrucțiune asociat *custom0* (codificarea instrucțiunii = 0x1), procesorul va transmite co-procesorului informațiile necesare realizării calculului, iar acesta va efectua operația și va întoarce un răspuns nucleului dacă acesta așteaptă un răspuns.

S-a realizat un co-procesor care să implementeze o coadă de mesaje de tipul FIFO (first-in-first-out).

În ceea ce privește metodele de testare și validare a design-ului s-au realizat simulări ale cazurilor extreme de utilizare dar s-a și implementat hardware-ul dedicat integrat în arhitectura RISC-V pe FPGA.