|  |  |
| --- | --- |
|  | Zbornik radova Fakulteta tehničkih nauka, Novi Sad |

**UDK: (Upisuje redakcija, ostaviti ovaj red)**

**DOI: (Upisuje redakcija, ostaviti ovaj red)**

**Implementacija vektorskog procesora baziranog na RISCV arhitekturi**

**Implementation of a vector processor based on a RISCV vector architecture**

Nikola Kovačević, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – EMBEDED SISTEMI I ALGORITMI**

**Kratak sadržaj –** *U ovom radu prezentovan je 32-bitni vektorski procesor zasnovan na verziji 0.8 RISCV - „V“ nacrta vektorske ekstenzije [1], implementiran na FPGA platformi.*

**Ključne reči:** *RISCV arhitektura, vektorski procesor, uvm metodologija*

**Abstract** – *This paper presents 32 bit vector processor based on the version 0.8 draft of RISCV - “V” vector extension, implemented on a FPGA platform.*

**Keywords:** *RISCV arhitecture, vector processor, UVM metodology*

**1. UVOD**

2004. godine Denardovo skaliranje (*eng. Dennard scaling*)[2] prestaje da važi i frekvencije rada procesora se od tada sve sporije povećavaju. Ta promena je mikroprocesorsku industriju primorala da pronalazi nova rešenja kako bi povećala performanse procesora i ono što se pokazalo kao prekretnica jeste stavljanje akcenta na paralelizam prilikom obrade podataka (*eng. Data level parallelism*).

Trenutno, najuspešnija i najzastupljenija arhitektura za paralelnu obradu podataka jeste GPU (*eng. Graphics processing unit*). No, Vektorski procesori, koji datiraju još od 60-tih godina prošlog veka, su jedna od arhitektura kod kojih je ova vrsta paralelizma takođe jako izražena, ali su do skora smatrani jako „skupim“. Jedan razlog je broj tranzistora, ali drugi, možda i bitniji, potreba za DRAM (*eng. Dynamic Random Accesss Memory*) memorijama koje mogu dovoljno brzo da „nahrane“ vektorski procesor podacima[2, poglavlje 4]. Napretkom tehnologije, potrebom za što većom paralelizacijom obrade podataka, sa što većom energetskom efikasnošću, pojavom RISCV instrukcijskog seta, ovi procesori su ponovo skrenuli pažnju na sebe. Iz tih razloga ovaj rad se bavi analizom i implementacijom vektorskog procesora baziranog na RISCV arhitekturi.

**1.1. Pregled rada po poglavljima**

Poglavlje 2 je uvod u vektorsko procesiranje. Tu se opisuju različite vrste paralalelizma(njihove prednosti i mane), pomoću jednostavnog primera objašnjen je način rada vektorskih procesora i opisana je RISCV arhitektura. Takođe, objašnjeno je šta su FPGA platforme i zašto je vektorski procesor u ovom radu implementiran pomoću njih. Poglavlje 3 posvećeno je implementaciji vektorskog procesora. Ono je podeljeno na tri velike celine: opis skalarnog jezgra, opis RISCV vektorske ekstenzije i opis mikroarhitekture vektorskog jezgra. U poglavlju 4 opisano je jednostavno verifikaciono okruženje zasnovano na UVM (*eng. Universal Verification Methodology*) metodologiji pomoću kojeg je potvrđen ispravan rad procesora. Poglavlje 5 je posvećeno analizi iskorišćenosti resursa i performansi prilikom implementacije procesora na *zybo* razvojnoj ploči. Poglavlje 6 sumira rad i predlaže moguća poboljšanja u budućem radu.

**2. Uvod u vektorske procesore**

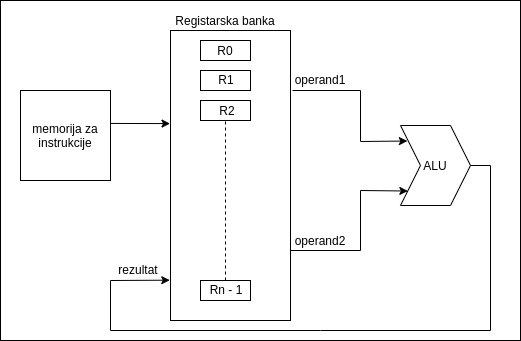
Kako bi povećali performanse, preko granica koje dopušta trenutna tehnologija izrade čipova, sistemske arhitekte pribegavaju različitim vrstama paralelizacije:

* Paralelizam na nivou instrukcija (*eng. Instruction Level Parallelism*) omogućava istovremeno izvršavanje više instrukcija iz jednog sekvencijalnog skupa. Najjednostavniji primer su procesori sa protočnom obradnom [1, poglavlje 3], kod kojih pre nego što se završi egzekucija prethodne instrukcije, kreće se sa narednom.
* Paralelizam na nivou niti (*eng. Thread Level Parallelism*) omogućava istovremeno izvršavanje instrukcija iz više odvojenih skupova. Najočigledniji primer su multiprocesorski sistemi kod kojih svaki procesor može da izvršava njemu dodeljen skup instrukcija.
* Paralelizam na nivou podataka(eng. *Data Level Parallelism*) omogućava izvršavanje jedne iste operacije istovremeno nad nizovima elemenata. Primer su vektorski procesori koji jednu vektorsku instrukciju primenjuju na više podataka istovremeno.

Naravno, arhitekte nisu ograničene na eksploatisanje samo jedne vrste paralelizacije, tako na primer u multiprocesorskim sistemima pojedinačna jezgra su u najjednostavnijoj varijanti procesori sa protočnom obradom.

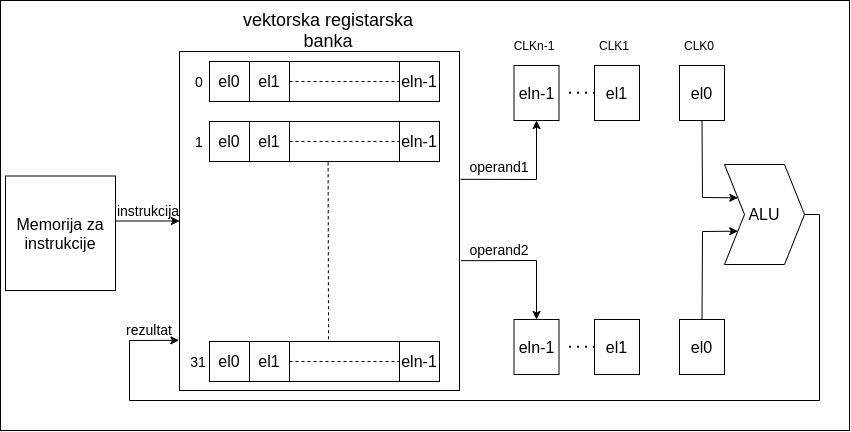
**2.1. Vektorsko procesiranje**

Princip rada skalarnih procesora predstavlja dobar uvod u vektorske procesore, te će stoga početak ove sekcije biti posvećen tome. Slika 2 ilustruje pojednostavljenu strukturu skalarnog procesora i na njoj su prikazane 3 ključne komponente: memorija za instrukcije, registarska banka i ALU (aritmetičko logička jedinica). Registarska banka unutar sebe skladišti određeni broj registara i na osnovu prihvaćene instrukcije jedan par tih registara biće pročitan i sproveden do aritmetičko logičke jedinice. Nad njima će se izvršiti određena operacija u zavisnosti od prihvaćenene instrukcije i rezultat će biti smešten u jedan od registara registarske banke.



Slika 1. *Pojednostavljena struktura skalarnog procesora*

Struktura vektorskih procesora je jako slična prethodno opisanoj, kao što se može videti na slici 2. Ključne komponente su iste, osim što je registarska banka zamenjena vektorskom registarskom bankom. Razlika između te dve komponente jeste u tome što se registarska banka sastoji iz registara određene širine (broj bita je određen arhitekturom procesora), dok se vektorska registarska banka sastoji iz vektora, pri čemu svaki vektor unutar sebe sadrži određeni broj elemenata (širina pojedinačnog elementa je takođe određena arhitekturom). Prihvatom vektorske instrukcije, kao što se kod skalarnog procesora čitaju registri registarske banke i izvršava određena operacija nad njima, tako se kod vektorskog procesora, iz vektorske registarske banke, čitaju vektori i jedna ista operacija se izvršava nad elementima unutar njih. Unutar instrukcije se nalazi informacija kojim se vektorima pristupa i svakim taktom biće pročitan jedan par elemenata, počevši od elemenata na indeksu „*i“*. Nad njma će se izvršiti određena operacija i rezultat će biti smešten u određeni vektor unutar vektorse registarske banke.



Slika 2. *Pojednostavljena struktura vektorskog procesora*

***2.2.* Prednosti vektorskog seta instrukcija**

Iz razloga što jedna kratka instrukcija može da opiše N operacija i da adresira 3N registarskih operanada, vektorski kod je kompaktan i neohpodna propusna moć prilikom prihvata instrukcija iz memorije je mnogo manja.

Vektorski set instrukcija umanjuje hardver neophodan prilikom dekodovanja vektorske instrukcije i razlog za to je što se jedna vektorska instrukcija primenjuje N puta na *N* elemenata. Takođe, *N* operacija koje se izvode su međusobno nezavisne, te iz tog razloga nije neophodan hardver za detekciju zavisnosti.

Šablon koji vektorski procesor prati prilikom izvođenja operacija nad elementima vektorskog registra je regularan. To omogućava visok stepen paralelizma, jer bi vektorski procesor mogao da se implementira pomoću više paralelnih „linija“ (*eng. Vector lanes*), pri čemu bi svaka „linija“ vršila operacije nad jednim delom vektorskih elemenata.

Vektorski set instrukcija može dodati kao ekstenzija na već postojeći skalarni set.

**2.3 RISC-V ISA (*eng. Instruction Set Architecture*)**

Arhitektura skupa instrukcija (eng. ISA, Instruction Set Architecture) opisuje na koji način određeni procesor funkcioniše i koje su njegove mogućnosti. Ona opisuje registre koje će procesor imati kao i sve mašinske instrukcije koje će podržavati [3]. Iz tog razloga prilikom projektovanja procesora neophodno je odabrati određenu arhitekturu, čiji set ili podset instrukcija će biti podržan. U ovom radu odabrana je RISCV ISA. Ova arhitektura, koja je potekla sa Berkli (*eng. Berkeley*) univerziteta, je novi set instrukcija (ISA) koji je na početku bio zamišljen da podrži naučna istraživanja i edukaciju, ali za koji sada postoji nada da će postati arhitektura koja će biti besplatna i otvorena za sve industrijske implementacije. U najjednostavnijoj varijanti, svaki procesor zasnovan na RISC-V arhitekturi bi trebalo da implementira bazni, *integer* set instrukcija (njegova skraćenica je „I“). Pored ovog baznog seta, RISC – V arhitektura podržava još mnoge i detalji o njima se mogu naći u [4].

**2.4 FPGA platforma i motivacija za njeno korišćenje**

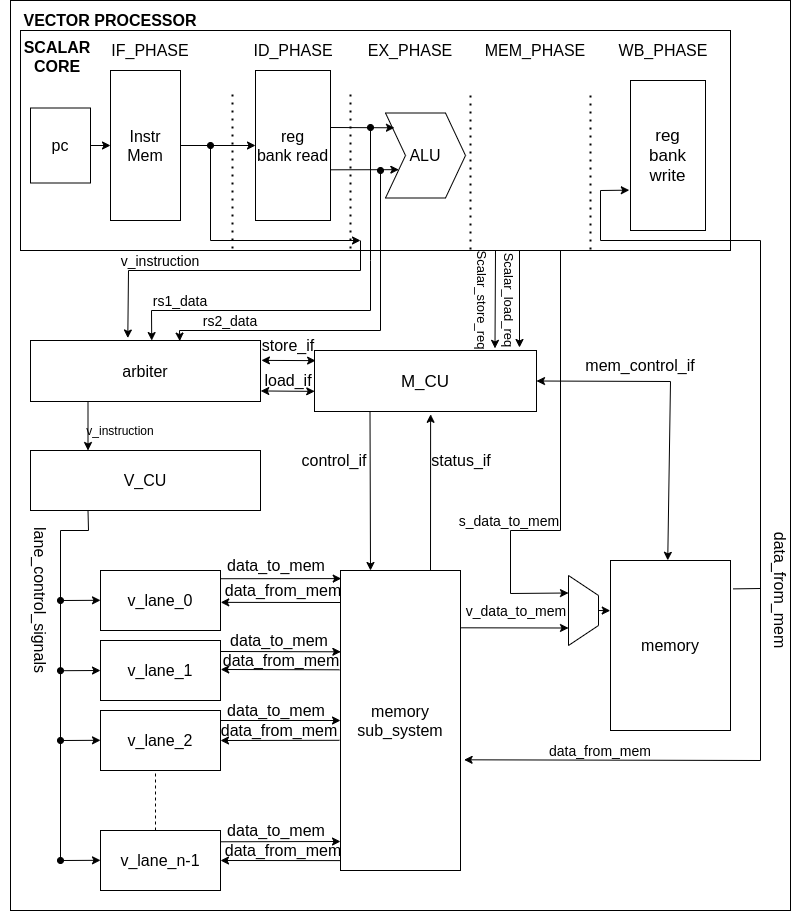
Vektorski procesor opisan u ovom radu je realizovan pomoću FPGA (*eng. Field Programmable Gate Arrays*) platforme. To su poluprovodnički uređaji zasnovani na matricama konfigurabilnih logičkih blokova povezanih pomoću programabilnih interkonekcija. Na ovim uređajima može biti implementiran bilo koji hardverski dizajn i što je najbitnije mogu biti reprogamirani kako bi se na njima implementirao neki drugi dizajn.

Sistemi koji se realizuju na FPGA platformama su najčešće akceleratori za aplikacije koje iskorišćavaju paralelizam između podataka. To su hardverski blokovi dizajnirani da obavljaju jednu vrstu zadatka sa ne toliko konfigurabilnih opcija. Oblasti u kojima je ovakav pristup zastupljen su: obrada slike i videa, mašinsko učenje (*eng. Machine learning*), emulacija hardvera, itd.

Ovaj rad istražuje alternativnu mogućnost korišćenja FPGA platforme za kreiranje vektorskog procesora kao akceleratora opšte namene. Procesor bi posedovao standardan set instrukcija, tako da bi svako, bez iskustva sa dizajnom hardvera, mogao da ga programira. Takođe, zbog mogućnosti reprogramiranja dizajna na FPGA platformama, moćiće da se menjanju neke od karakteristika procesora kako bi se povećale performanse ili kako bi se optimizovala iskorišćenost resursa. Na primer, broj vektorskih „linija“ (sekcija 2.2) bi mogao da se poveća, odnosno smanji, u zavisnosti od toga da li su od interesa performanse, ili zauzetost logičkih blokova na FPGA platformi.

**3.** [**Implementacija vektorskog procesora RV32IV**](#__RefHeading___Toc1497_4247007053)

U ovoj sekciji biće opisana RISC-V vektorska ekstenzija kao i mikroarhitektura procesora baziranog na njoj (slika 3). Osnovna ideja je da se pored skalarnog jezgra, koje podržava RISCV *integer* set instrukcija, implementira i vektorsko jezgro koje bi se ponašalo kao ekstenzija na već postojeći set instrukcija.

Slika 3. Blok dijagram vektorskog procesora.

Opis implementacije podeljen je na sledeće sekcije: Opis skalarnog jezgra, RISCV vektorska ekstenzija i karakteristike vektorskog procesora Mikroarhitektura vektorskog jezgra.

**Kratka biografija:**

|  |  |
| --- | --- |
| slika | **Petar Petrović** rođen je u Novom Sadu 1994. god. Master rad na Fakul­tetu teh­ničkih nauka iz oblasti Elektro­teh­nike i računarstva – Energetska elektro­nika i električne mašine odbranio je 2018.god.  kontakt: mail.adresa@gmail.com |