

Jedno rešenje protočnog procesora MIPS arhitekture u FPGA integrisanom kolu

Ivan Kaštelan, Pavle Savković, Zoran Krajačević i Branislav Atlagić

Sadržaj — U ovom radu je prikazano jedno rešenje računarskog sistema u FPGA integrisanom kolu (SoC). Sistem je zasnovan na protočnom procesoru MIPS arhitekture. Cilj rada je realizacija sistema sa minimalnim brojem primitiva FPGA integrisanog kola. Procesor je opisan u VHDL jeziku, dok je assembler za pomenuti procesor realizovan koristeći WinTIM meta-assembler. Realizovan računarski sistem je ispitan na primeru kalkulatora i primeru izvršavanja programa korak po korak (*debugger*). Ispitivanje je pokazalo da sistem zauzima mali broj primitiva FPGA integrisanog kola.

Ključne reči — Assembler, FPGA, MIPS, SoC, VHDL.

I. UVOD

SVRHA ovog rada je realizacija računarskog sistema u integrisanom kolu (SoC) zasnovanog na protočnom MIPS procesoru. Računarski sistem je razvijen u okviru projektnog zadatka istraživanja isplativosti i kvaliteta primene SoC u FPGA integrisanom kolu, kao procesnog elementa komercijalnih uređaja. Istraživanje je delom sprovedeno i kroz nastavni proces na Fakultetu tehničkih nauka, Univerzitetu u Novom Sadu.

Težilo se ka realizaciji sistema sa minimalnim brojem primitiva da bi se oko njega mogli smestiti dodatni namenski sklopovi kojima će MIPS upravljati. Korišćeno je FPGA integrisano kolo proizvođača Xilinx familije Spartan3 [1] na TLL5000 razvojnoj ploči [2]-[3].

Realizacija je podeljena u nekoliko koraka. Prvi korak obuhvata opis protočnog MIPS procesora sa okruženjem, u VHDL jeziku. Drugi korak je funkcionalna verifikacija projektovanog računarskog sistema simulacijom pomoću Cadence NCSim programskog alata. Treći korak je sinteza projektovanog računarskog sistema korišćenjem Xilinx WebPack ISE 9.3 programskog alata.

Ovaj rad je delimično finansiran od Ministarstva za Nauku Republike Srbije, projekat 12004, od 2008 god.

Ivan Kaštelan, Fakultet tehničkih nauka, Katedra za računarsku tehniku i računarske komunikacije, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija; (e-mail: ivankst@yahoo.com).

Pavle Savković, Fakultet tehničkih nauka, Katedra za računarsku tehniku i računarske komunikacije, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija; (e-mail: pavle.savkovic@krt.neobee.net).

Zoran Krajačević, Micronas NIT, Fruškogorska 11a, 21000 Novi Sad, Srbija; (e-mail: zoran.krajacevic@micronas.com).

Prof Dr Branislav Atlagić, Fakultet tehničkih nauka, Katedra za računarsku tehniku i računarske komunikacije, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija; (e-mail: branislav.atlagic@rt-rk.com).

Prevođenje izvornog koda programske podrške MIPS procesora je realizovano pomoću WinTIM meta-assemblera.

Ispitivanje rada sistema u stvarnom okruženju je realizovano u dva primera. Prvi primer je kalkulator u kome se koriste prekidači, tasteri, LED i LCD. Drugi primer je prikaz stanja procesora na ekranu pomoću VGA sprege na TLL5000 ploči.

II. OPIS ARHITEKTURE PROTOČNOG MIPS-A

Procesor realizovan u ovom radu je zasnovan na MIPS R2000 arhitekturi [4]-[5] sa frekvencijom rada od 24 MHz. Osnovna MIPS arhitektura je modifikovana prema potrebama primene sistema. Protočna struktura procesora se sastoji iz 5 faza: faze prihvatanja instrukcije (*Instruction Fetch*), faze dekodiranja instrukcije (*Instruction Decode*), faze izvršenja instrukcije (*Execute*), faze pristupa memoriji (*Memory Access*) i faze upisa u registre (*Write Back*). Arhitektura procesora je prikazana na sl. 1.

A. Faza prihvatanja instrukcije

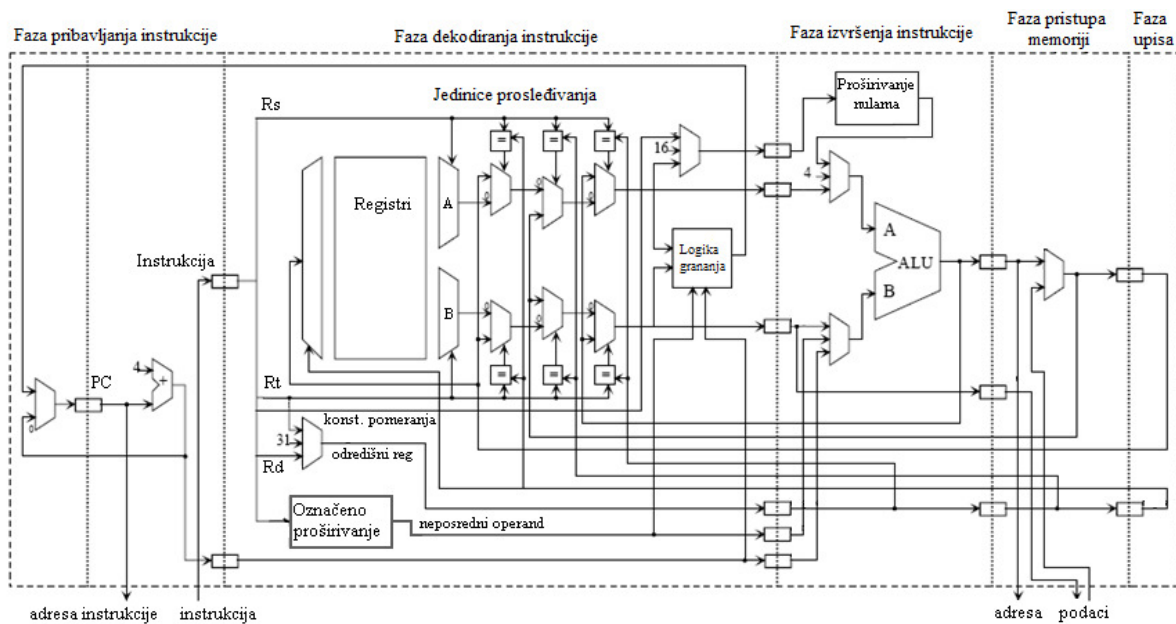
U okviru modula koji realizuje prvu fazu protočne strukture, realizovan je programski brojač koji se automatski uvećava za 4 ako trenutna instrukcija nije instrukcija skoka ili grananja. U slučaju instrukcije skoka, kao i grananja sa ispunjenim uslovom, u programski brojač se upisuje nova adresa instrukcije skoka/grananja. Upis se vrši tek kada se u narednim fazama proveriti ispunjenje uslova grananja i tip instrukcije.

Instrukcija se čita iz memorije koja je zajednička i za podatke i za instrukcije, kao što predviđa Von Neumann-ova arhitektura [6].

B. Faza dekodiranja instrukcije

Druga faza protočne strukture sadrži dekoder instrukcije, kao i registre, logiku grananja i logiku koja rešava rizike protočne strukture. Struktura i tipovi instrukcija su zadržani kao i u osnovnoj MIPS R2000 arhitekturi [4]-[5].

Rizici grananja (*control hazards*) se rešavaju odlaganjem odluke o grananju, tako što se u protočnu strukturu uvode naredne instrukcije sve dok se ne otkrije potreba za grananjem. Nakon toga se izvršava instrukcija grananja na potrebnu adresu, a nepotrebno uvedene instrukcije poništavaju i odstranjuju iz protočne strukture. Rizici podataka (*data hazards*) se rešavaju standardno, prosleđivanjem vrednosti u naredne faze protočne strukture.



Sl. 1. Arhitektura protočnog MIPS procesora

C. Faza izvršenja instrukcije

Treću fazu protočne strukture čini aritmetičko-logička jedinica. U zavisnosti od instrukcije, operandi se biraju između registara, programskog brojača, prosledenih vrednosti zbog rizika ili konstante 4, koja se koristi za uvećanje adrese prilikom instrukcije grananja ili skoka.

D. Faza pristupa memoriji

U četvrtoj fazi protočne strukture realizovana je logika kojom se pristupa memoriji, prilikom korišćenja *load* i *store* instrukcija. Pošto sistem koristi istu memoriju i za instrukcije i za podatke, procesor se prilikom pristupa memoriji u ovoj fazi zadržava 1 takt. Poseban modul, koji predstavlja automat stanja, vodi računa o zadržavanju procesora i tipu pristupa memoriji – da li je to pristup instrukciji ili podacima.

E. Faza upisa u registre

Faza upisa u registre nije realizovana u posebnom modulu, jer su registri deo modula faze dekodiranja instrukcije. Međutim, za ovu fazu je potreban jedan poseban signal takta u protočnoj strukturi.

F. Upravljačka jedinica procesora

Upravljačka jedinica procesora je realizovana kao kombinaciona mreža i u zavisnosti od trenutno prihvaćene instrukcije generiše upravljačke signale za ostatak procesora. Aktivna je u fazi dekodiranja instrukcije, a upravljački signali za kasnije faze se prenose pomoću registara između stepena protočne strukture.

G. Operativna memorija

Memorija koju koristi procesor je dvoprístupna memorija veličine 4096 reči od 32 bita. Čita se na jednom prolazu (*port*), a upisuje na drugom prolazu. Radna ivica takta memorije je suprotna od radne ivice takta procesora, da bi se sprečilo nepotrebno zaustavljanje procesora.

III. OPIS ASEMLERA ZA MIPS PROCESOR

Kao platforma za assembler je korišćen WinTIM meta-assembler [7]. WinTIM omogućava da se u definicionoj datoteci definiše mašinski format za svaku instrukciju procesora. Pomoću pomenute datoteke se prevodi assemblerski program i formira datoteka sa memorijskim sadržajem koja je potrebna za sintezu.

Realizovani procesor podržava potpuni skup instrukcija predviđen MIPS R2000 arhitekturom [4]-[5]. U arhitekturi procesora direktno su podržane aritmetičke, logičke, instrukcije skoka, uslovnog grananja, pomeranja, pristupa memoriji, poziva potprograma, povratka iz potprograma, kao i instrukcija za poređenje dva registra sa upisom rezultata poređenja u treći registar. Assembler dodatno podržava i pojedine pseudo-instrukcije koje koriste ranije navedene grupe instrukcija u svojoj realizaciji.

Procesor ne podržava magacinsku memoriju (*stack*), tako da je omogućen samo jedan nivo poziva potprograma. Povratna adresa se smešta u jedan od registara. Ako programer želi više nivoa poziva potprograma, mora sam da vodi računa da povratnu adresu prethodnog nivoa zapamti u pojedinom od registara opšte namene.

IV. PRIMER 1: KALKULATOR

Prva primena u kojoj je ispitan rad realizovanog računarskog sistema je kalkulator koji izvodi 5 računskih operacija: sabiranje, oduzimanje, množenje, bit-konjunkciju i bit-disjunkciju. Zauzetost komponenata FPGA integrisanog kola nakon sinteze i procenat njihove zauzetosti su prikazani u tabeli 1.

Pošto je okruženje procesora u ovom primeru zanemarljivo u odnosu na procesor i memoriju, rezultati iz tabele 1 se mogu iskoristiti za procenu složenosti samog procesora sa memorijom. Analiza rezultata pokazuje relativno malu zauzetost FPGA integrisanog kola, svega

8% kombinacionih blokova, koji najbolje prikazuju opštu zauzetost. Ovako mala zauzetost omogućava mnoga proširenja sistema.

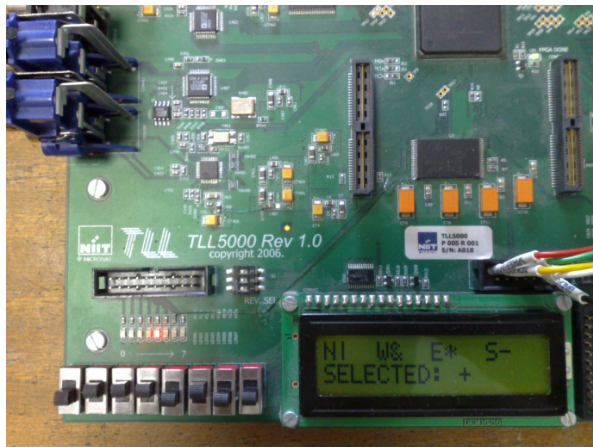
TABELA 1: ZAUZETOST FPGA INTEGRISANOG KOLA TIPA SPARTAN3 XC3S1500FG676-4 PRI REALIZACIJI KALKULATORA.

Komponenta	Zauzetost	Procenat
BUFGMUX	1 od 8	12%
Spoljni IOB	30 od 487	6%
RAMB16	8 od 32	25%
Slice (CLB)	1074 od 13312	8%

Operandi kalkulatora su 4-bitni brojevi koji se biraju pomoću prekidača na TLL5000 ploči. Operacija se bira pomoću tastera. Rezultat se ispisuje kao 8-bitna vrednost na LE diodama, dok se istovremeno poruka o trenutno izabranoj operaciji ispisuje na LCD ekranu.

Veza procesora sa okruženjem je realizovana pomoću zasebnog registarskog polja. Registarsko polje je zauzelo poslednjih 8 adresa adresnog prostora memorije. MIPS procesor komunicira sa okruženjem preko priključaka FPGA integrisanog kola koji su mapirani na pojedine bite registara u registarskom polju.

Na početku asemblerskog programa se inicijalizuje LCD, zatim se na njemu ispisuju podržane operacije, kao što je prikazano na sl. 2, a potom sledi glavna petlja u kojoj se realizuje funkcija kalkulatora. Podrazumevana operacija, u slučaju da nije pritisnut nijedan taster ili je pritisnuto više od jednog tastera, daje rezultat sabiranja. U suprotnom svaka operacija je pridružena odgovarajućem tasteru. Sve operacije su direktne, sem množenja koje je realizovano metodom uzastopnog sabiranja.



Sl. 2. Izgled TLL5000 ploče tokom rada sa kalkulatorom.

V. PRIMER 2: IZVRŠAVANJE PROGRAMA KORAK PO KORAK

Drugi primer u kojem je ispitan rad projektovanog računarskog sistema omogućava izvršavanje instrukcija MIPS procesora korak po korak i ispis sadržaja izabranih registara na VGA ekranu. Ovaj primer zahteva i realizaciju sprege na VGA ekran pomoću sprege koja postoji na TLL5000 ploči.

Zauzetost komponenata FPGA integrisanog kola nakon sinteze i procenat njihove zauzetosti su prikazani u tabeli

2. Iz tabele se vidi veća zauzetost u odnosu na primer kalkulatora zbog mnogo većeg okruženja koji predstavlja sprege na VGA ekran kao i signali kojima su prenete vrednosti iz bitnih procesorskih registara ka VGA sprezi.

TABELA 2: ZAUZETOST FPGA INTEGRISANOG KOLA TIPA SPARTAN3 XC3S1500FG676-4 PRI REALIZACIJI KORAČNOG IZVRŠAVANJA PROGRAMA.

Komponenta	Zauzetost	Procenat
BUFGMUX	2 od 8	25%
Spoljni IOB	60 od 487	12%
RAMB16	9 od 32	28%
Slice (CLB)	2660 od 13312	19%

Sprega na VGA ekran je realizovana unutar posebnog modula koji generiše izlazne signale za VGA ekran. Znaci su definisani unutar posebne ROM memorije. Na ekranu se ispisuju sadržaji prvih 8 registara opšte namene, kao i trenutne vrednosti programskog brojača, instrukcionog registra i registra na izlazu iz aritmetičko-logičke jedinice.

U ovom primeru nije korišćen sistemski takt od 24 MHz, već je korisniku omogućeno da kontroliše izvršavanje programa korak po korak, generišući takt pritiskom i otpuštanjem odgovarajućeg tastera.

Asemblerski program za ovaj primer je isti kao i program za kalkulator u prvom primeru.

Zbog prirode protočnog procesora, mora se voditi računa u kojoj fazi je koja instrukcija. Pošto se prikazani registri nalaze u različitim fazama protočne strukture, oni se odnose na različite instrukcije. Ako je na primer, trenutno prihvaćena instrukcija n , njen kod i adresa će biti prikazani kao sadržaji instrukcionog registra i programskog brojača, ali izlaz iz aritmetičko-logičke jedinice će se odnositi na instrukciju $n-2$ jer se ona u tom trenutku nalazi u fazi izvršenja.

VI. ZAKLJUČAK

Računarski sistem u integrisanom kolu, zasnovan na protočnom MIPS procesoru je prvenstveno korišćen u edukativne svrhe. Slobodan prostor u FPGA integrisanom kolu omogućava dodatno sprežavanje sa različitim fizičkim podsistemima TLL5000 ploče ili neke druge platforme sa Xilinx Spartan3 FPGA integrisanim kolom.

Realizovani procesor pruža mogućnost izvršenja širokog spektra asemblerskih programa. Prikaz stanja procesora na VGA ekranu omogućava praćenje rada programa i otkrivanje grešaka u programu korak po korak. Ovim je dobijen sistem koji se može koristiti u razne algoritamske svrhe, proširivati novim IP jezgrima zbog male zauzetosti FPGA integrisanog kola i koristiti kao osnova za praktičnu edukaciju o arhitekturi računarskih sistema zbog male cene i dostupnosti FPGA integrisanih kola.

S obzirom na pad cena FPGA integrisanih kola i tendenciju njihove primene u uređajima masovne potrošnje, ovakav računarski sistem se može primeniti i u komercijalne svrhe. Zbog male zauzetosti komponenata FPGA integrisanog kola, MIPS procesor je moguće

iskoristiti kao jezgro budućeg komercijalnog sistema.

Plan za dalje istraživanje je proširivanje skupa instrukcija koje podržava procesor, npr. uvođenje instrukcija množenja i deljenja, kao i realizacija magacinske memorije za veće mogućnosti prilikom poziva potprograma. Za komercijalne primene planira se i detaljna analiza propusne moći procesora.

LITERATURA

- [1] Spartan-3 FPGA Family: Complete Data Sheet – Product Specification, Xilinx, November 30, 2007
- [2] LL5000 User Manual: LL5000 Base Board Design, October 23, 2006, Texas Learning Labs and Micronas NIT
- [3] Nemanja Popov, Zoltan Pele, Dušan Majstorović, Zoran Krajačević, Mihajlo Katona: “Ugrađeni sistem za razvoj kontrolisan ARM 7 mikrokontrolerom”, ETRAN, 2007
- [4] Vladimir Kovačević, Branislav Atlagić: “Logičko projektovanje računarskih sistema 2 – Projektovanje računarskih sistema”, FTN izdavaštvo, Novi Sad, 2007
- [5] Zoran Krajačević: “Praktikum laboratorijskih vežbi iz predmeta Logičko projektovanje računarskih sistema 2”, Fakultet tehničkih nauka, Novi Sad, 2007
- [6] David A. Patterson, John L. Hennessy: “Computer Organization and Design – The Hardware/Software Interface”, Morgan Kaufmann, 2005

- [7] J. Hamblen: “Introduction to WinTIM”, School of ECE, Georgia Tech, USA, 2000

ABSTRACT

This paper presents one implementation of SoC based on the pipelined MIPS processor inside field programmable gate array. The goal is to design a system using the minimal number of FPGA primitives. The processor is described in VHDL and the assembly for the processor is implemented using the WinTIM meta-assembly language. The implemented SoC is tested in two examples: calculator and step by step debugger with VGA output, on TLL5000 board. The SoC occupied the small number of primitives inside FPGA, which allows further extensions of the system.

ONE IMPLEMENTATION OF THE PIPELINED MIPS ON FPGA

Ivan Kaštelan, Pavle Savković, Zoran Krajačević and Branislav Atlagić