

REALIZACIJA EDUKACIONOG MIKORORAČUNARA NA ZYNQ KONFIGURABILNOM SISTEMU NA ČIPU**EDUCATIONAL MICROCOMPUTER IMPLEMENTATION ON ZYNQ RECONFIGURABLE SYSTEM ON CHIP**Balaž Petrović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je opisan proces realizacije ZEdulent školskog mikrorračunara na ZYNQ sistemu na čipu. Prikazan je proces projektovanja mikrorračunara i grafičkog korisničkog interfejsa. Pored toga opisan je način verifikacije i testiranja.

Ključne reči: ZYNQ, ZYBO, FPGA, Edulent

Abstract – In this paper a process of ZEdulent educational microcomputer implementation on Zybo board is described. Microcomputer and graphical user interface design process is depicted. Besides, verification and testing methodology is given.

Keywords: : ZYNQ, ZYBO, FPGA, Edulent

1. UVOD

Jednostavnost opisivanja logičkih kola preko jezika za opis hardvera, kao što je VHDL, i dostupnost FPGA čipova su bili razlozi za projektovanje i realizaciju edukacionog 8-bitnog mikroprocesora (Edulent) [1] na kojem studenti mogu da izvršavaju sopstvene programe u assembleru. Pored procesora, takođe u okviru projekta dizajniran je grafički interfejs na PC računaru u Visual C++ jeziku koje omogućava studentima da prate rad mikroprocesora [1].

Takav sistem omogućava studentima da upišu program u memoriju mikroprocesora, zatim da izvršavaju program na mikroprocesoru baziranom na FPGA i da grafički prikazuju unutrašnje stanje procesora preko grafičkog interfejsa.

Koristeći Edulent, studenti su u mogućnosti da upoznaju osnovnu arhitekturu mikroprocesora i assembleri jezik programiranja. Zbog toga Edulent simulator je korišćen godinama na laboratorijskim vežbama. Na osnovu pozitivnih povratnih informacija, ovaj simulator je olakšao korake studentima, kao što su: razumevanje funkcionalnosti mikroprocesora, lakše razumevanje dizajna mikroprocesora, programiranje u assembler jeziku, izvršenje programa i izvršavanje programa korak po korak.

Komunikacija između FPGA čipa i PC grafičkog interfejsa se odvijalo preko LPT paralelnog porta. Pošto na novijim računarima LPT port je zamenjen kompaktnijim USB portovima, postoji potreba za realizacijom Edulent mikrorračunara na novijim FPGA čipovima, koji se koristi za savremeniji vid komunikacije.

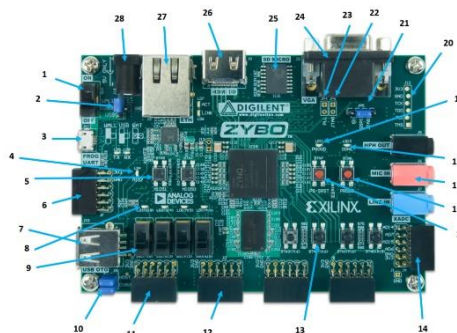
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ivan Mezei, vanr. prof.

Savremenija verzija Edulenta je realizovana na ZYNQ konfigurabilnom sistemu na čipu, pod nazivom ZEdulent. ZEdulent je edukacioni mikroprocesor projektovan da predstavlja evolutivni razvoj Edulent mikroprocesora [1]. Zahtevi koje moraju da zadovolje ZEdulent su isti kao i za Edulent: jednostavna arhitektura, grafički korisnički interfejs, sposoban za pokretanje jednostavnih programa korak-po-korak, assembler jezik za programiranje i jednostavno povezivanje sa periferijama.

1.1 Zybo razvojna ploča

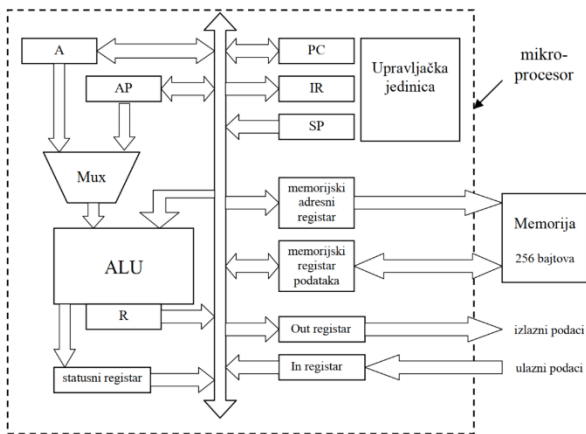
ZYBO (Zynq BOard) je fleksibilna, embedid razvojna platforma softvera i digitalnog dizajna, početnog nivoa. Razvojna platforma izgrađena oko najmanjeg člana porodice Xilinx Zynq-7000, Z-7010. Z-7010 je zasnovan na Xilinx All Programmable System-on-Chip (AP SoC) arhitekturi, koja integriše dual-core ARM Cortex-A9 procesor i programabilnu logiku Xilinx 7-series Field Programmable Gate Array (FPGA). Kada se SoC spoji sa bogatim skupom multimedijalnih i perifernih uređaja za povezivanje koji su dostupni na ZYBO-u, Zynq Z-7010 može da podržava dizajn čitavog sistema. Ugrađena memorija, video i audio ulaz/izlaz, USB, Ethernet i SD slot će omogućiti dizajn sistema bez dodatnog hardvera. Pored toga, na raspolaganju je šest Pmod portova za specijalne potrebe dizajna [2].



Slika 1. Zybo board

1.2 Model Edulent mikrorračunara

Uzimajući u obzir razmatranja iz uvodnog dela gde je naglašen značaj arhitekture za početno izučavanje iz oblasti mikroprocesora, odabrana je 8-bitna arhitektura. Postoji više različitih podela mikroprocesora prema arhitekturi. Odabrana je arhitektura koja predstavlja modifikovanu akumulatorsku arhitekturu von Neumann tipa tako da se linearan memorijski prostor koristi i za program i za podatke. U ovom poglavlju su prikazane osnovne karakteristike, načini adresiranja i skup instrukcija dati u [1].



Slika 2. Model mikroracunara

Mikroprocesor ima 11 registara i svi su 8-bitni. Dva su registra direktno programski dostupna korisniku. Dva registra služe kao pomoćni registri za realizaciju ulazno-izlazne komunikacije sa periferijskim uređajima.

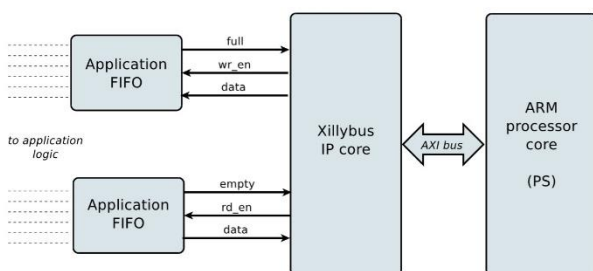
Edulent ima ukupno 40 instrukcija (uključujući različite načine adresiranja).

1.3 Xillybus

Xillybus je jednostavan, prenosiv, i efikasno rešenje za prenos podataka između FPGA i hosta na kojoj se izvšava operativni sistem (Linux ili Microsoft Windows). Dostupan je za personalne računare i za embedid sisteme koji koriste PCI Express bus za prenos podataka, takođe podržava ARM-bazirane procesore, koji koriste AMBA magistralu za prenos podataka (AXI3 / AXI4) [3].

FPGA dizajner kao i host programer komunicira sa Xillybus-om preko dobro poznatih interfejsa: FPGA programabilna logika se povezuje na IP core preko standardnih FIFO-a; korisnička aplikacija na host računaru obavlja obične I/O operacije na device fajlovima.

Tok podataka se kreće prirodno između FIFO-a i datoteke koju je otvorila host aplikacija. Ne postoji specifičan API, koji treba instalirati i koristiti.



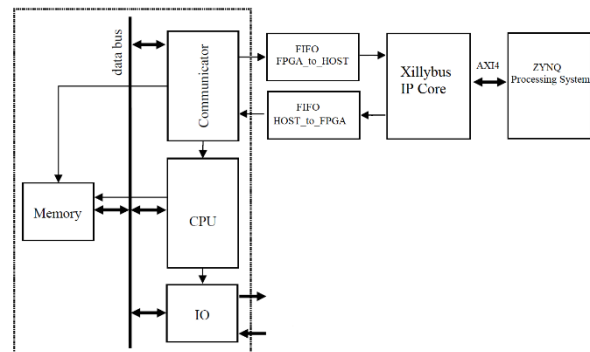
Slika 3. Xillybus ip core

Kao što je gore prikazano na Sl. 3, programabilna logika aplikacije na FPGA-u treba da bude u interakciji sa standardnim FIFO-ima.

Aplikacija na računaru čita i piše podatke u obične datoteke uređaja koje se ponašaju kao cevi. Xillybus IP jezgro i kernel drajveri prenose podatke efikasno i intuitivno između FIFO-a u FPGA i njihovih odgovarajućih device fajlova na hostu.

2. REALIZACIJA MIKRORAČUNARA NA ZYBO RAZVOJNOM SISTEMU

Blok šemu dizajna možemo videti na Sl. 4. Edulent IP jezgro je povezan preko FIFO bufera na Xillybus IP. Xillybus IP jezgro komunicira sa ARM procesorom preko AXI4 magistrale.



Slika 4. Blok šema sistema

Na blok šemi možemo videti modifikacije u odnosu na prethodne verzije Edulenta [1]. U ranijim verzijama komunikaciona jedinica, komunicirala je sa PC računaru preko LPT porta. U ovoj realizaciji komunikaciona jedinica komunicira sa host računaru preko FIFO bafera, zbog toga je potrebno projektovanje nove komunikacione jedinice.

Ostali delovi mikropcesorskog sistema kao što su CPU, memorija i ulazna-izlazna jedinica ostaju suštinski nepromenjena. Potrebne su modifikacije u opisu dvosmerne magistrale. Vivado alat zahteva eksplicitno drajvovanje za dvosmernu magistralu.

2.1 Komunikaciona jedinica

Tabela 1. Rezultati implementacije

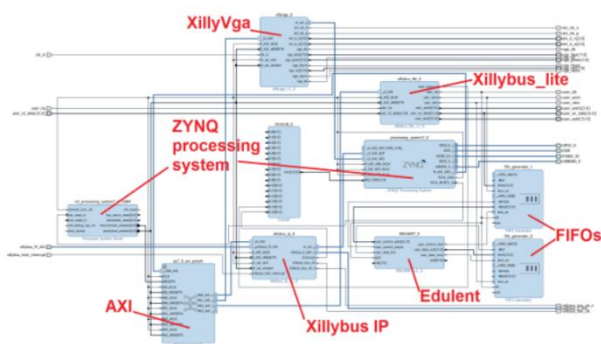
Slice blocks	5254 out of 17600 (30%)
Flip-flops	4986 out of 35200 (14%)
4 input LUTs	474 out of 4704 (10%)
I/O blocks	30 out of 146 (20%)
Tri state buffers	116 out of 2352 (4%)
RAM blocks	5 out of 60 (8%)
Clock frequency	20 MHz

Zbog edukacionog karaktera celog sistema i želje da se u svakom trenutku može imati uvid u kompletno stanje sistema, potrebno je osim mikroracunara realizovati i odgovarajući blok za komunikaciju sa personalnim računaru. Ovo je realizovano u komunikator bloku. Komunikator predstavlja vezu između mikroracunara i PC računaru. Komunikacioni blok treba da omogući izvršavanje aktivnosti kao što su pokretanje programa, izvršavanje instrukcija korak po korak itd. U suštini komunikacioni blokima funkcionalnost jednostavne upravljačke jedinice, jer treba da obezbedi odgovarajuće kontrolne signale pri čitanju stanja registara i memorije ili pri upisu u njih. Pored toga ovaj blok mora da omogući i odgovarajuće zaustavljanje takta procesora ili na neki drugi način zaustavljanje rada procesora kada je to potrebno. Način komunikacije je takav da komunikacioni blok prima instrukcije od PC hosta po određenom

protokolu. Na osnovu instrukcije komunikator šalje vrednost registara, šalje vrednosti iz memorije, upiše novu vrednost u neki registar(ako je to dozvoljeno), upiše vrednost u memoriju na određenu adresu, i da pokrene procesor na jedan od tri moguća načina.

2.2 Realizacija na FPGA čipu

Na Sl. 5. možemo videti blok šemu sistema. FPGA deo sistema se sastoji od više modula. Postoje moduli koji su potrebni u svakom dizajnu. Ovoj grupi pripada: ZYNQ processing sistem, Processing System Reset, AXI Interconnect. Moduli koji su potrebni za prikazivanje na monitoru i za komunikaciju između PL i PS sistema su: xillyvga IP jezgro, xillybus IP jezgro, xillybus_lite IP jezgro, dva FIFO IP jezgro. Modul koji implementira funkcionalnost Edulent mikroracunara je Edulent IP jezgro.



Slika 5. Šema sistema IP jezgara [4]

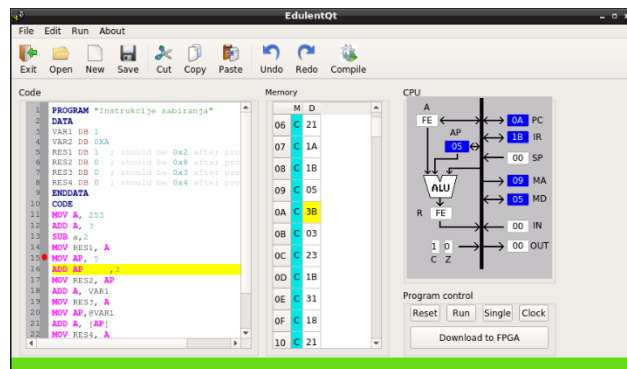
Sistem je implementiran pomoću Xilinx Vivado alata za dizajn koristeći VHDL jezik za opis hardvera za IP jezgra (Sl. 5.). Rezultati implementacije za XSC7Z010 SoC na ZYBO ploči su date u tabeli 1. Upoređivanjem rezultata implementacije sa rezultatima implementacije ranijih verzija Edulent mikroracunara [1] vidimo da broj Slice-blokova i flip-flop-ova su znatno veći. To je zbog činjenice da ZEdulent ima dodatnu logiku u odnosu na Edulent (npr. FIFO, Xillybus IP jezgro i XillyVga IP jezgro).

3. PROGRAMSKA PODRŠKA NA QT

Zbog edukacionog karaktera ZEdulenta, čiji je osnovni cilj da omogući studentima početna učenja iz oblasti mikroprocesora i mikroracunara, potrebna je vizualizacija stanja mikroracunara. U ovoj sekciji opisan je grafički korisnički interfejs.

Za projektovanje GUI korišćeno je razvojno okruženje Qt Creator. Ovaj program podržava metode potrebne za generisanje i korišćenje GUI elemenata. Qt koristi C++ jezik, s tim se program sastoji od više objekta.

Na Sl. 6. možemo videti Edulent aplikaciju u Run mod-u sa postavljenim Brake point-om. Brake point je postavljen na liniju 15 u CodeEditoru, mikroprocesor izvršio instrukciju na liniji 15 (MOV AP, 5), zatim je zaustavljen Brake point-om. Žuta linija u CodeEditor-u je postavljena na sledeću instrukciju, na liniju 16. U memorijskom prostoru žutom linijom označeno je mesto gde programski brojač pokazuje.



Slika 5 Edulent u Run modu

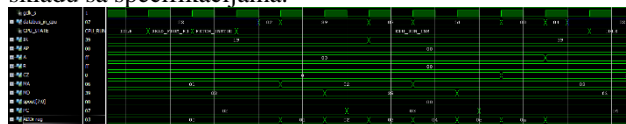
Brake point se postavlja duplim klikom na liniju teksta u CodeEditor-u, isto tako sa duplim klikom se briše. Brake point treba da bude postavljen pre kompajliranja.

4. VERIFIKACIJA

Verifikacija kao zadatak ima da potvrdi da logika dizajna radi usklađeno sa dizajniranom specifikacijom. Dakle, verifikacija pokušava da da odgovor na pitanje: "Da li ovaj predloženi dizajn radi ono što ste mislili?" Za ovaj složeni zadatak potrebno je dosta vremena i napora u većini projekata tokom projektovanja elektronskih sistema.

U okviru ovog rada je projektovana nova komunikaciona jedinica, samim tim je potrebna temeljna funkcionalna verifikacija ove jedinice. U ovom radu je izvršena funkcionalna verifikacija pomoću simulatora Xilinx Vivado alata i posmatranjem izlaznih signala.

Svaka instrukcija Edulent procesora je simulirana jedan po jedan sve dok se svi funkcionalno ne verifikuju. Svaka instrukcija komunikacione jedinice je simulirana, zatim posmatranjem izlaznih signala zaključeno je da radi u skladu sa specifikacijama.



Slika 6 Simulacija instrukcije ADD A, 0x7D

Na Sl. 7. prikazani su signali mikroprocesora iz simulacije. Simulirana je u single modu. Komunikator modul je dražovan porukom za pokretanje procesora za izvršavanje jedne instrukcije. Kada komunikator detektuje poruku u ulaznom FIFO baferu, prelazi u stanje Read_from_fifo. U ovom koraku učitava poruku u sopstvene registre. U narednom koraku je dekodovanje instrukcije Fetch_instruction. Nakon dekodovanja komunikator i/ili mikroprocesor u zavisnosti od instrukcije izvršavaju akcije vezane za instrukciju. U ovom slučaju to je izvršenje instrukcije ADD A, 0x7D. Možemo videti da procesor radi ispravno.

Poslednja metoda u postupku verifikacije je provera rada na FPGA kolu. Ovaj korak spada delimično u verifikaciju, a delimično u upotrebu. Ova metoda predstavlja pogodanu metodu za proveru ispravnosti rada celog sistema. Međutim, u slučaju pojave greške ona ne ukazuje tačno na izvor greške već samo približno. Zato je često potrebno uraditi simulaciju da bi se precizno utvrdila

greška. Kao i kod prethodne metode, ovo ukazuje na činjenicu da sve metode verifikacije daju najbolje rezultate kada se koriste u kooperaciji.

U ovoj fazi je testirana Qt programska podrška i rad celog sistema. Da bi mogao da se testira rad celog sistema prvo je potrebno verifikovati ispravnost programskog prevodioca. Ovo je urađeno tako što su redom testirane sve instrukcije sa svim tipovima adresiranja i praćene su njihove vrednosti na mašinskom jeziku koje je prevodilac generisao.

Nakon verifikacije kompajlera moglo se pristupiti funkcionalnoj verifikaciji pomoću programa za vizualizaciju rada mikror računara. Za te potrebe napisano je više programa u simboličkom mašinskom jeziku tako da se pokriju sve instrukcije i svi tipovi adresiranja. Ovi programi su izvršavani u tri režima rada. Prvi režim podrazumeva celokupno izvršavanje programa i poređenje rezultata sa očekivanim. Drugi režim rada podrazumeva izvršavanje programa po principu instrukcija po instrukcija. Ovaj režim omogućava proveru izvršavanja pojedinačnih instrukcija. U trećem režimu rada program se izvršava po principu takt po takt i on omogućava praćenje izvršavanja mikroinstrukcija. Na ovaj način detaljno je testiran i funkcionalno verifikovan ZEdulent sistem.

5. ZAKLJUČAK

U ovom radu je opisan proces realizacije novije verzije Edulent mikror računara, pod nazivom ZEdulent. Na ZYNQ ZYBO razvojnoj ploči je instalirana osnovna verzija Linux Ubuntu operativnog sistema, pod nazivom Xillinux. Zajedno sa operativnom sistemom instalirani su i drajveri potrebni za korišćenje Xillybus IP jezgra, koje služe za realizaciju komunikacije između FPGA dela ploče i Linux operativnog sistema.

U okviru rada projektovana je nova komunikaciona jedinica za komunikaciju sa Xillybus IP jezgrom. Dizajn mikrop procesora je prilagođen novoj jedinici. Funkcionalna verifikacija je urađena u simulatoru Vivado programskog paketa. Zatim je izvršena implementacija ZEdulent sistema zajedno sa potrebnim IP jezgrima za video izlaz. Programiranje FPGA dela ploče je rađeno zajedno sa podizanjem Xillinux operativnog sistema.

Razvoj programske podrške se odvijao na Windows PC računaru, korišćenjem Qt Creator programa. Razvijen je programski prevodilac, i uspostavljena je komunikacija sa drajverima Xillybus IP jezgra. Završna faza razvoja je urađena u Xillinux operativnom sistemu, na ZYBO platformi. Svrha programske podrške je aktivacija i praćenje različitih režima rada programa na mikror računaru, kao i vizuelno prikazivanje dinamike izvršavanja programa.

Testiranje i funkcionalna verifikacija celog sistema je urađena pomoću nekoliko programa pisanim na simboličkom mašinskom jeziku koje pokrivaju sve instrukcije mikrop procesora. Programi su izvršeni u sva tri načina rada mikrop procesora.

Budući pravci razvoja su proširenje procesora sa nekim dodatnim funkcionalnostima (npr. rad sa prekidima, UART itd.). Osim toga moguće je projektovanje 16-bitne i/ili 32-bitne verzije Edulent mikrop procesora.

6. LITERATURA

- [1] I. Mezei: "Formalna specifikacija i realizacija laboratorijskog mikror računara na programabilnom integrisanom kolu" Magistarska teza, FTN, Novi Sad 2005
- [2] ZYBO Reference manual: Digilent https://www.xilinx.com/support/documentation/university/XUP%20Boards/XUPZYBO/documentation/ZYBO_RM_B_V6.pdf accessed on: 29.03.2019
- [3] Xillybus IP datasheet: <http://xillybus.com/doc> accessed on: 22.03.2019
- [4] Balaz Petrovic, Ivan Mezei: „Educational microprocessor implementation on ZYNQ reconfigurable system on chip“, poslato za prikaz na međunarodnoj konferenciji

Kratka biografija:



Balaz Petrović rođen je u Senti 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Embeded sistemi i algoritmi odbranio je 2019.god. kontakt: pbalazs@stcable.net