Višestruki PWM upravljač zasnovan na FPGA tehnologiji

Korman, Zvonimir

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:688847

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

VIŠESTRUKI PWM UPRAVLJAČ ZASNOVAN NA FPGA TEHNOLOGIJI

Diplomski rad

Zvonimir Korman

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 20.09.2021.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Zvonimir Korman
Studij, smjer:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. studenta, godina upisa:	D-34ARK, 06.10.2019.
OIB studenta:	99609694324
Mentor:	Izv. prof. dr. sc. Tomislav Matić
Sumentor:	Izv. prof. dr. sc. Ivan Aleksi
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Ivan Vidović
Član Povjerenstva 1:	Izv. prof. dr. sc. Tomislav Matić
Član Povjerenstva 2:	Josip Zidar
Naslov diplomskog rada:	Višestruki PWM upravljač zasnovan na FPGA tehnologiji
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak diplomskog rada:	Potrebno je dizajnirati vlastiti PWM modul u VHDL-u. Implementirati najmanje 10 PWM modula te ih istovremeno upravljati s MicroBlaze mikroupravljačem. Tema rezervirana za: Zvonimir Korman
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	20.09.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 13.10.2021.

Ime i prezime studenta:	Zvonimir Korman
Studij:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. studenta, godina upisa:	D-34ARK, 06.10.2019.
Turnitin podudaranje [%]:	5

Ovom izjavom izjavljujem da je rad pod nazivom**: Višestruki PWM upravljač zasnovan na FPGA tehnologiji**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Tomislav Matić

i sumentora Izv. prof. dr. sc. Ivan Aleksi

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD *1*

2. PREGLED PODRUČJA TEME *2*

3. VHDL PWM UPRAVLJAČ *3*

3.1. Uvod u PWM **3**

3.2. PWM signali **4**

3.2.1. Klasični PWM 4

3.2.2. Fazno ispravni PWM 4

3.2.3. Fazno ispravni PWM s faznim pomakom 5

3.2.4. Fazno ispravni PWM upravljač s faznim pomakom (PCPS PWM upravljač) 6

4. FPGA SKLOP ZA PCPS PWM *8*

4.1. VHDL PCPS PWM kôd **9**

4.2. MicroBlaze i Xilinx XPS **10**

4.3. PCPS PWM upravljač s MicroBlaze **12**

4.4. PWM s XPS Timer/Counter jezgrom **15**

5. TESTIRANJE I ANALIZA REZULTATA *17*

5.1. Testiranje VHDL modula **17**

5.2. Testiranje PCPS PWM s MicroBlaze računalnim sustavom **18**

5.3. Usporedba **21**

5.3.1. Prednosti i nedostatci sustava 21

5.3.2. Zahtjevnost korištenja 22

5.3.3. Usporedba upotrebe resursa FPGA integriranog sklopa 22

6. ZAKLJUČAK *27*

SAŽETAK *30*

ABSTRACT *31*

ŽIVOTOPIS *32*

PRILOZI *33*

1. UVOD

Tema ovog diplomskog rada je razvoj višestrukog PWM (engl. *Pulse Width Modulation*) upravljača zasnovanog na FPGA (engl. *Field-programmable gate array*) tehnologiji. PWM, odnosno pulsno-širinska modulacija predstavlja tehniku manipuliranja analognim signalom kako bi se dobio promjenjiv napon. Promjenjivi napon se realizira na način da se manipulira sa pravokutnim signalom, odnosno količinom vremena kada je on u logičkoj jedinici ili nuli. Izlazni napon je određen omjerom „high“ signala (vrijednost 1) u odnosu na „low“ signal (vrijednost 0). FPGA predstavlja poluvodiče, odnosno uređaje bazirane na matrici konfigurirajućih logičkih blokova, koji su međusobno programibilni. Zbog te funkcionalnosti takvi uređaji mogu se reprogramirati ovisno o traženim funkcionalnostima za razliku od integriranih krugova specifičnih za pojedine aplikacije(engl. *ASICs – Application Specific Integrated Circuits*).

Cilj ovog rada je bio realizirati optimizirani sustav pomoću kojega bi krajnji korisnik mogao dinamički mijenjati broj PWM izlaza kao i traženu preciznost cjelokupnog sustava, a da se pri tome postižu zadovoljavajuće performanse. Realizirani sustav je zamišljen da bude jednostavan za korištenje te da za njega nisu potrebni dodatni sklopovi. Ostatak rada organiziran je kako slijedi. U drugom poglavlju prolazi se kroz područja tehnologije obuhvaćena temom diplomskog rada. U trećem poglavlju navedene su osnove PWM signala, podjela PWM signala te način rada realiziranoga PWM modula u VHDL-u. U četvrtom poglavlju objašnjava se računalni sustav temeljen na fazno ispravnom PWM-u s faznim pomakom i navode se njegove osnovne funkcionalnosti. Također se objašnjava i računalni sustav temeljen na postojećem rješenju kreirano od strane Xilinx tehnološke kompanije. U petom poglavlju se testiraju i analiziraju rezultati. Šesto poglavlje zaključuje rad.

2. PREGLED PODRUČJA TEME

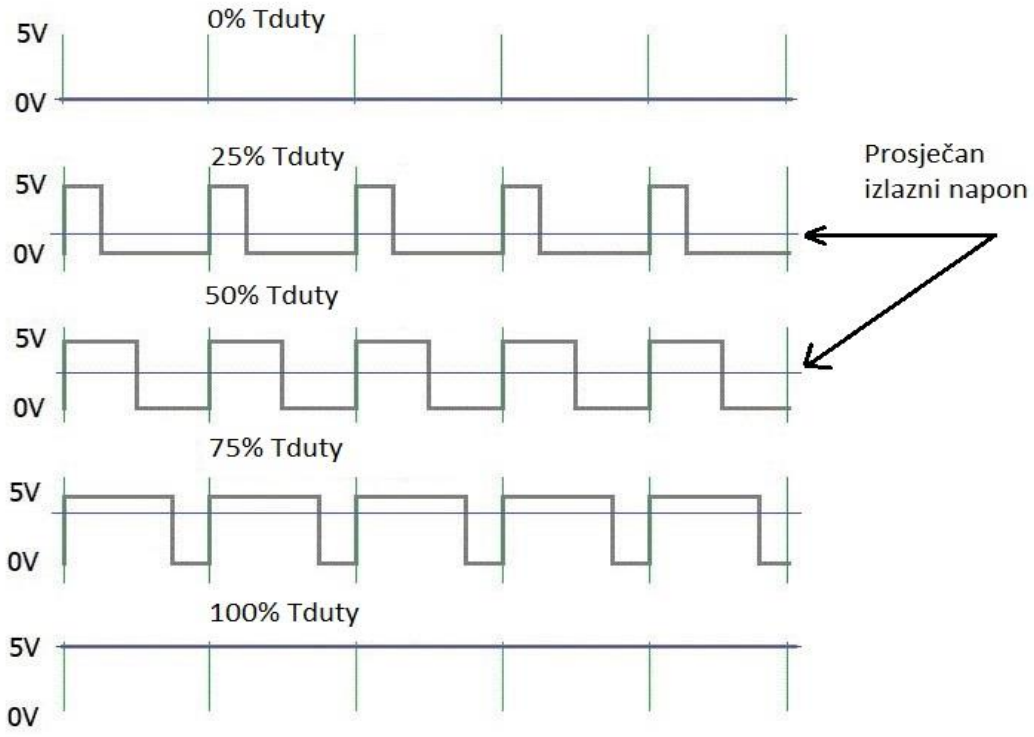
S obzirom na svoju široku upotrebu, PWM signal se može realizirati na više načina. Neki od njih su u obliku mikrokontrolera, poput AT89SP2 ili PIC18F4550 mikrokontrolera [1], [2]. Primarna uloga takvih uređaja je generiranje jednog izlaznog PWM signala i njegova kontrola. Arduino razvojne ploče također sadrže mikrokontrolere, odnosno ATmega168P i ATmega328P koji sadrže 3 PWM brojača, čime se može ostvariti do 6 PWM izlaza [3]. PWM signali se mogu ostvariti i koristeći DSP (engl. *Digital signal processor*) vrstu procesiranja za generiranje višestrukih PWM izlaza [4]. Takav pristup se koristi ukoliko je potreban isključivo veći broj PWM izlaza. Postojeća rješenja dijele isto ograničenje, a to je da je broj PWM izlaza ograničen [5]. Kombinacijom funkcionalnosti i dizajna PWM kontrole sa korisničkim sučeljem koristeći programski jezik C postiže se dinamičko mijenjanje izlaza i potrebne preciznosti izvedbe uz zadovoljavajuće performanse.

Tehničko rješenje ovog rada napravljeno je koristeći Xilinx Platform Studio (skraćeno XPS). Xilinx, tehnološka kompanija zadužena za razvoj ranije spomenutog FPGA dizajna uređaja, kroz vlastitu plaformu XPS nudi alate namijenjene za razvoj hardvera. XPS nudi mogućnost integracije prilagođenih IP jezgri (engl. *Intellectual property*) koji predstavljaju logički ili podatkovni blok koji se koristi u izradi FPGA uređaja. Među njima se nalazi i IP jezgra koja se u posebnim uvjetima može koristiti za izvedbu pulsno-širinske modulacije naziva „XPS Timer/Counter“. Takva modulacija predstavlja osnovnu mogućnost stvaranja pulsno-širinske modulacije, međutim koristeći jedan modul izvedba je ograničena na jedan izlazni signal [6]. Kao dodatna mogućnost alata XPS nude se IP jezgre s prilagođenim VHDL ili Verilog dizajnom, koji se implementira unutar modula ili od treće strane. Zbog mogućnosti korištenja navedenih programskih jezika omogućeno je kreirati vlastitu logiku koja ima funkcionalnost pulsno-širinske modulacije [7], [8].

3. VHDL PWM UPRAVLJAČ

3.1. Uvod u PWM

Digitalni signali imaju dva osnovna stanja, a to su: uključeno ili isključeno. Ta stanja se na principu binarnih brojeva zapisuju kao 1 za uključeno a 0 za isključeno. Međutim, analogni signali mogu biti različitih, odnosno od neograničeno mnogo veličina koje se nalaze između 0 i 1. Korištenjem sklopovlja i programske podrške javlja se potreba da analogni i digitalni signali rade zajedno, ali te dvije vrste signala se elektronički različito zapisuju i obrađuju. PWM je u suštini način kontroliranja analognih signala koristeći digitalni izlaz. PWM ima vrlo širok raspon upotrebe, od jednostavnijih primjera poput pokretanja jednostavnih naprava kao što su svjetlo s mogućnošću prigušivanja, aktuatori i zvučnici. Također postoje i složeniji primjeri poput kontrole električnog motora unutar vozila ili kontroliranja pojedinog automobilskog kotača u slučaju električnog auta s motorom za svaki pojedini kotač [9]. Osnovni prikaz PWM signala prikazan je na slici 3.1.



Sl. 3.1. Osnovni prikaz PWM signala, Tduty je širina trajanja logičke jedinice u postotcima.

3.2. PWM signali

Kako bi se bolje objasnila uloga različitih principa PWM signala, potrebno je dodatno pojasniti svaki od njih. Načini rada pulsno-širinske modulacije su:

- klasični PWM
- fazno ispravni PWM
- fazno ispravni PWM s faznim pomakom.

Svaki od navedenih načina ima svoje prednosti i nedostatke. Rezultat izvođenja bilo kojeg od navedenih načina rada rezultiraju generiranjem odgovarajućih izlaznih signala, no koriste različit princip generiranja [10], [11].

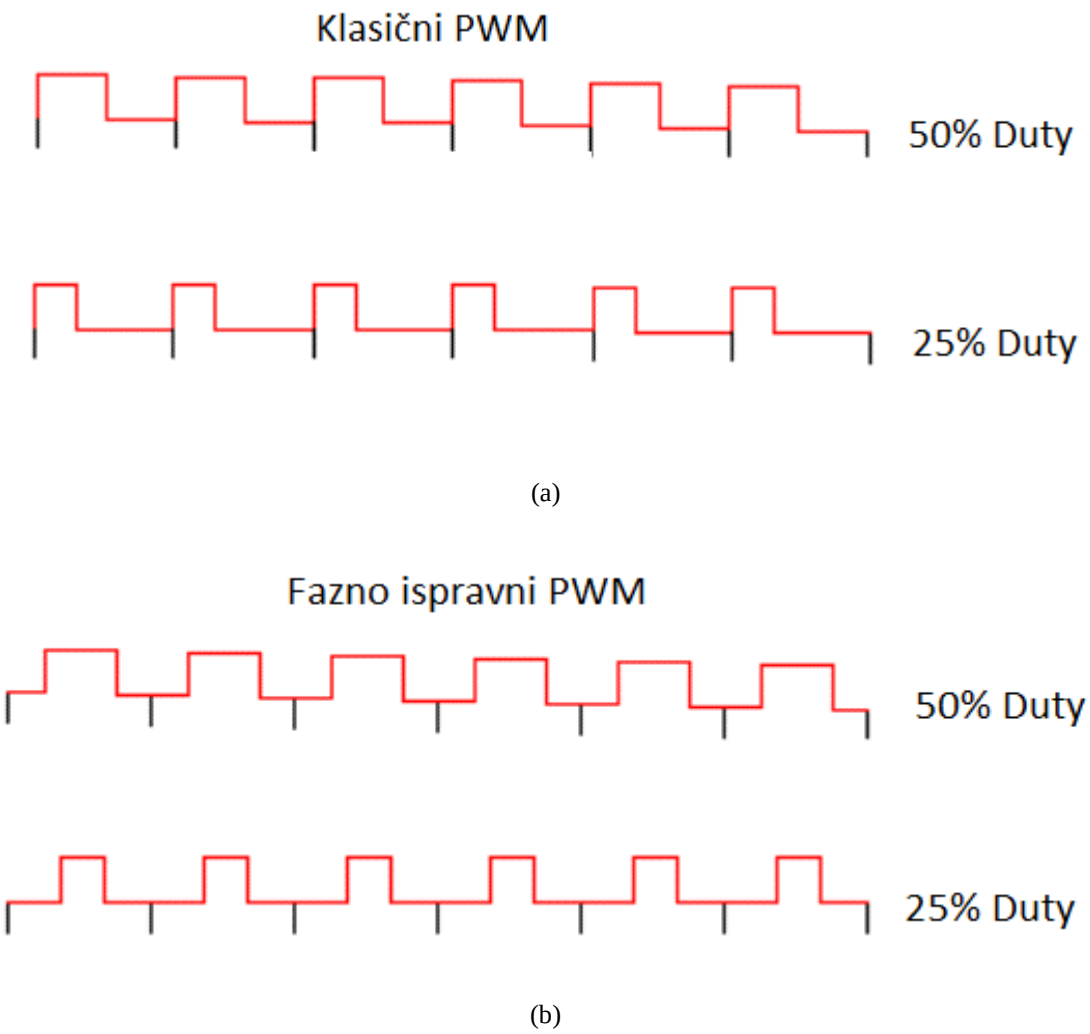
3.2.1. Klasični PWM

Klasični PWM način rada osigurava generiranje valnih oblika visoke frekvencije. Klasični PWM razlikuje se od ostalih opcija PWM-a svojim radom s jednim nagibom (engl. *single-slope*). Brojač računa od BOTTOM to TOP, a zatim se ponovno pokreće s BOTTOM. TOP je definiran heksadekadskom vrijednošću 0xFF. Zbog rada s jednim nagibom, radna frekvencija brzog PWM moda može biti dvostruko veća od fazno ispravnog PWM načina rada zato što fazno ispravni PWM način koristi rad s dvostrukim nagibom. Ova visoka frekvencija čini klasični PWM način rada vrlo prikladnim za aplikacije za regulaciju i ispravljanje. Visoka frekvencija omogućuje korištenje fizički malih vanjskih komponenata (zavojnice, kondenzatori), i samim time smanjuje ukupne troškove sustava. U brzom PWM načinu rada brojač se povećava sve dok vrijednost brojača ne odgovara gornjoj vrijednosti. Brojač se tada briše na sljedećem ciklusu radnog takta. Prikaz klasičnog PWM-a kao i njegova usporedba sa fazno ispravnim načinom rada nalazi se na slici 3.2a.

3.2.2. Fazno ispravni PWM

Fazno ispravni PWM omogućuje generiranje visoko rezolucijskog PWM valnog oblika s ispravnom fazom. Način rada PWM s ispravnom fazom temelji se na radu s dvostrukim nagibom(engl. *Dual-slope*). Brojač se neprestano broji od BOTTOM do TOP, a zatim od TOP do BOTTOM. TOP je definiran heksadekadskom vrijednošću 0Xff kao i u primjeru brzog PWM-a. Iako rad s dvostrukim nagibom ima nižu maksimalnu radnu frekvenciju od rada s jednim nagibom, zbog simetričnosti PWM načina rada s dvostrukim nagibom, ti su načini poželjniji za upravljanje motora. U fazno ispravnom PWM načinu kada brojač dosegne vrijednost TOP, mijenja smjer brojanja. Na slici 3.2b je prikazan fazno ispravni PWM te se usporedbom a) i b)

slike vidi temeljna razlika između standardnog i fazno ispravnog načina rada za PWM. Razlika je u početku perioda, gdje se u fazno ispravnom PWM-u signali poravnavaju u odnosu na fazu, a ne početak rastućeg brida kao u slučaju klasičnog.

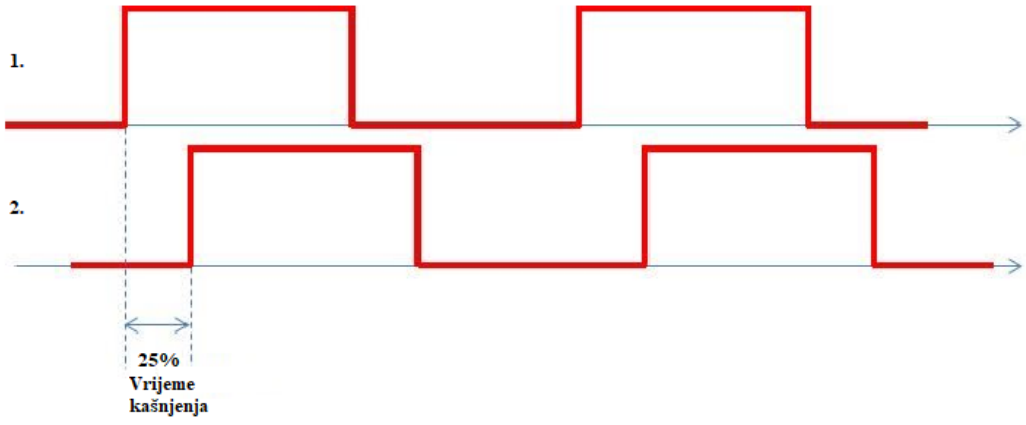


Sl. 3.2. Klasični PWM (a), fazno ispravni PWM signal (b), crne crte označavaju početak peerioda.

3.2.3. Fazno ispravni PWM s faznim pomakom

Fazno ispravni PWM način rada s faznim pomakom pruža mogućnost generiranja PWM valnog oblika visoke razlučivosti za fazu i frekvenciju. Kao i kod fazno ispravnog PWM načina, ovaj način rada je također zasnovan na radu s dvostrukim nagibom. Glavna razlika između fazno ispravnog i fazno-frekvencijskog načina rada PWM -a je vrijeme izvođenja i ažuriranja koja se izvode u međuspremniku, odnosno u fazno ispravnom PWM-u s ispravnom frekvencijom kod više izlaza se može generirati određeno kašnjenje jednog signala u odnosu na drugi a da su pri

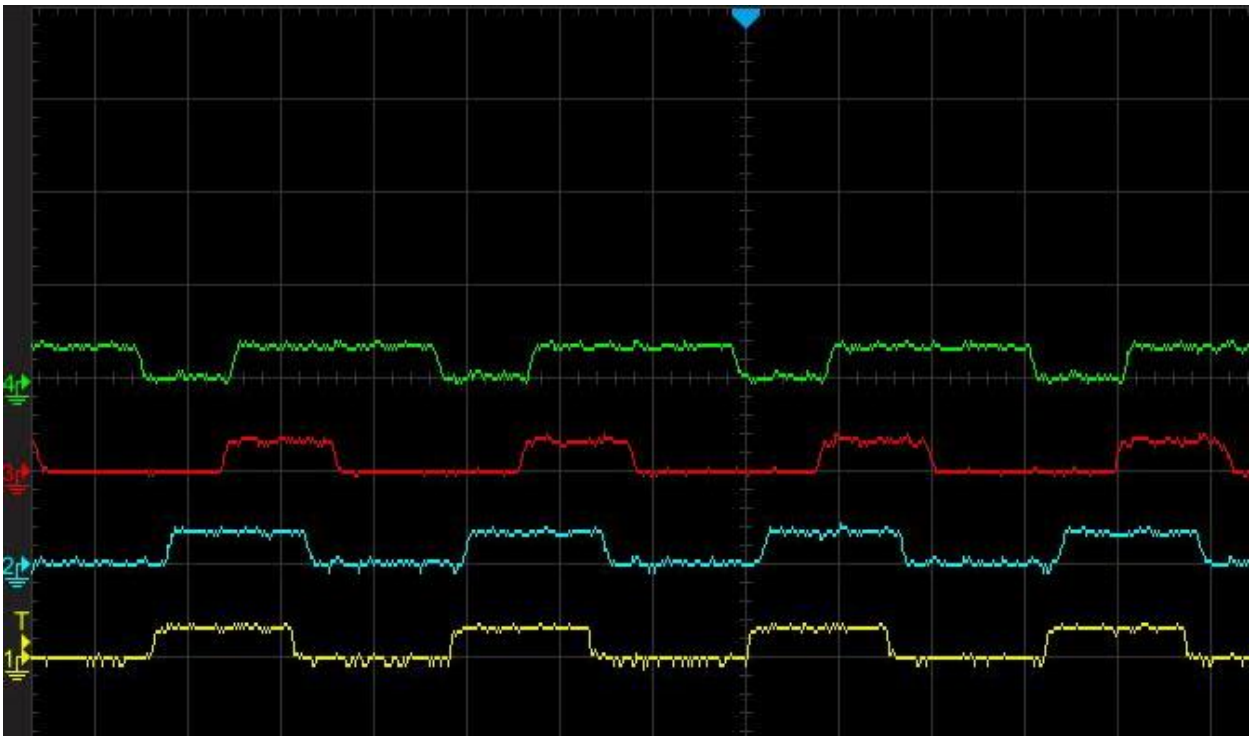
tome frekvencije jednake. Na slici 3.3 je prikazan fazno ispravni PWM s faznim pomakom, gdje je prikazano kašnjenje jednog signala u odnosu na drugi za iznos od 25% iznosa perioda.



Sl. 3.3. Prikaz fazno ispravnog PWM-a s faznim pomakom i kašnjenjem od 25%.

3.2.4. Fazno ispravni PWM upravljač s faznim pomakom (PCPS PWM upravljač)

PWM upravljač odašilje PCPS, odnosno fazno ispravne, fazno pomaknute (engl. *Phase Correct, Phase Shift*) PWM signale na temelju trajanja logičke jedinice, odnosno radnog ciklusa (Tduty) i kašnjenja u odnosu na referentni signal (Tdelay). Širina impulsa varira oko središta, a središte svakog impulsa javlja se na PWM frekvenciji. Radni ciklus određuje trajanje logičke jedinice u razdoblju u kojem se pojavljuju rastući i padajući rubovi PWM signala, dok signal za kašnjenje generira kašnjenje u odnosu na zadane vrijednosti. S obzirom na mogućnost korištenja više faza signala za PWM, komponenta generira PWM signale za svaku fazu, razmaknute i različitog trajanja. PCPS PWM upravljač može kontrolirati radni ciklus i kašnjenje davanjem novih vrijednosti određenim signalima te ponovnim generiranjem programske datoteke sa promijenjenim parametrima. Na slici 3.4 se nalazi prikaz 4 PCPS PWM signala.



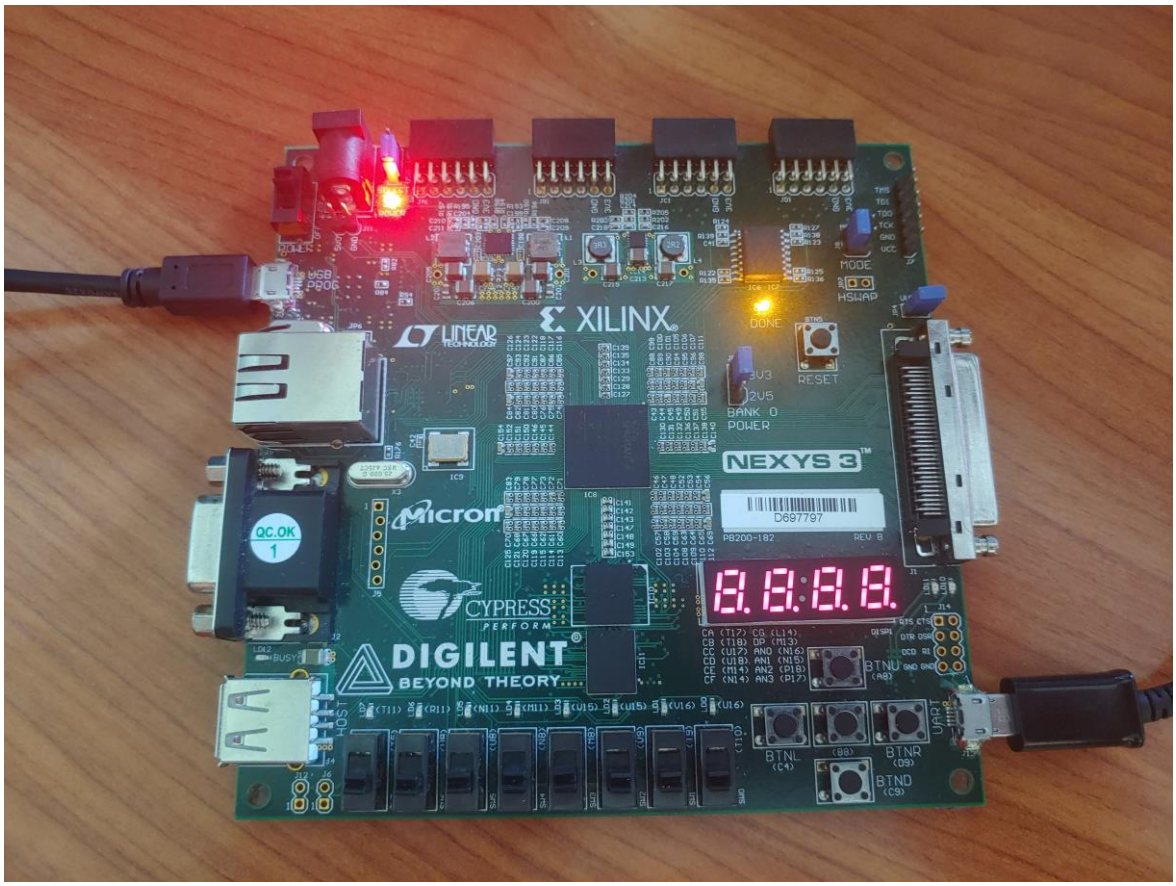
Sl. 3.4. Prikaz 4 PCPS PWM signala.

4. FPGA SKLOP ZA PCPS PWM

Budući da trenutno ne postoji PCPS PWM upravljač u ovom radu korišten je FPGA razvojni sustav i VHDL za implementaciju istog. FPGA uređaj se koristi kako bi se na njemu izvodila generirana računalna logika te zbog svojih programibilnih svojstava. VHDL je korišten kao programski jezik zbog mogućnosti komunikacije sa sklopovljem i modeliranjem ponašanja i strukture realiziranog modula [12].

Za realizaciju ovog računalnog sustava korišteno je:

- Windows 10 računalo sa Xilinx Platform Studio i ISE Design Suite 14.7
- Nexys3 razvojni sustav baziran na Spartan 6 FPGA
- računalna tipkovnica za komunikaciju putem UART-a
- Voltcraft osciloskop
- GW Instek MSO 2104E osciloskop
- GW Instek DS2-16LA logički analizator.



Sl. 4.1. Nexys3 razvojni sustav.

Nexys 3 razvojni sustav omogućava dizajniranje virtualnih sklopova izradom logičkih blokova koristeći programski jezik VHDL. Virtualni sklopovi se mogu dizajnirati i sintetizirati, što omogućava njihovu implementaciju. Implementacijom se logički dizajn pretvara u datoteku koja je prikladna za spremanje u Xilinx-ov FPGA integrirani sklop. Izgled razvojnog sustava prikazan je na slici 4.1. Jedan Micro USB kabel omogućava komunikaciju pomoću UART-a dok drugi napaja razvojni sustav i omogućuje programiranje sustava.

Sl. 4.2. GW Instek MSO 2104E i GW Instek DS2-16LA logički analizator.

Za provjeru rada i testiranje izlaza PWM signala korišten je osciloskop. S obzirom da je za potrebe testiranja bilo nužno koristiti više od 4 izlaza korišten je GW Instek MSO 2104E osciloskop koji ima mogućnost prikazivanja više od 4 signala istovremeno. Korištenjem logičkog analizatora omogućena je tražena funkcionalnost prikaza više od 8 izlaza. Koristeći ovo rješenje moguće je prikazati do 16 PWM izlaznih signala. Spomenuti osciloskop te logički analizator prikazani su na slici 4.2.

4.1. VHDL PCPS PWM kôd

PCPS VHDL modul realiziran je kako bi korisnik unutar razvojnog okruženja mogao mijenjati broj PWM izlaza i rezolucijsku varijablu. Vrijednosti radnog takta i komponente za kašnjenje svakog pojedinog PWM signala upisuju se kao vektori elemenata u obliku „std_logic_vector“ tipa podataka. U dijelu koda 4.3 vidimo dvije generičke varijable koje predstavljaju temeljne varijable potrebne za rad. Promjenom tih varijabli unutar VHDL koda i generiranjem programske datoteke sustav radi sa novim parametrima.

Linija ***Kod***

```
84:     entity user_logic is
85:     generic
86:     {
87:     nPWM : natural :=8;
88:     nRes : natural :=8;
89:     };
```

Sl. 4.3. Generičke varijable sustava.

Kako bi se uspješno prikazao fazno ispravni PWM s faznim pomakom, logika petlje kroz koju se zadaje vrijednost se nalazi u kodu 4.4. Kroz petlju se prolazi onoliko puta koliko je zadana generička varijabla nPWM. Postoje 4 različita uvjeta o kojima ovisi koja vrijednost će se predati signalu. Izlazna vrijednost se predaje na Pmod portove pri svakom rastućem bridu radnog takta sustava.

Linija ***Kod***

```
254:    for i in 0 to nPWM-1 loop
255:       if(Tduty(i) = (Tduty(i)'range => '0')) then
256:         PWMs(i) <= '0';
257:       elsif(Tduty(i) = (Tduty(i)'range => '1')) then
258:         PWMs(i) <= '1';
259:       elsif (count = Tduty_half(i) + Tdelay(i)) then
260:         PWMs(i) <= '0';
261:       elsif(count = period + Tdelay(i) - Tduty_half(i) + 1 ) then
262:         PWMs(i) <= '1';
263:       end if;
264:    end loop;
```

Sl. 4.4.Unos vrijednosti na izlazni signal PWM.

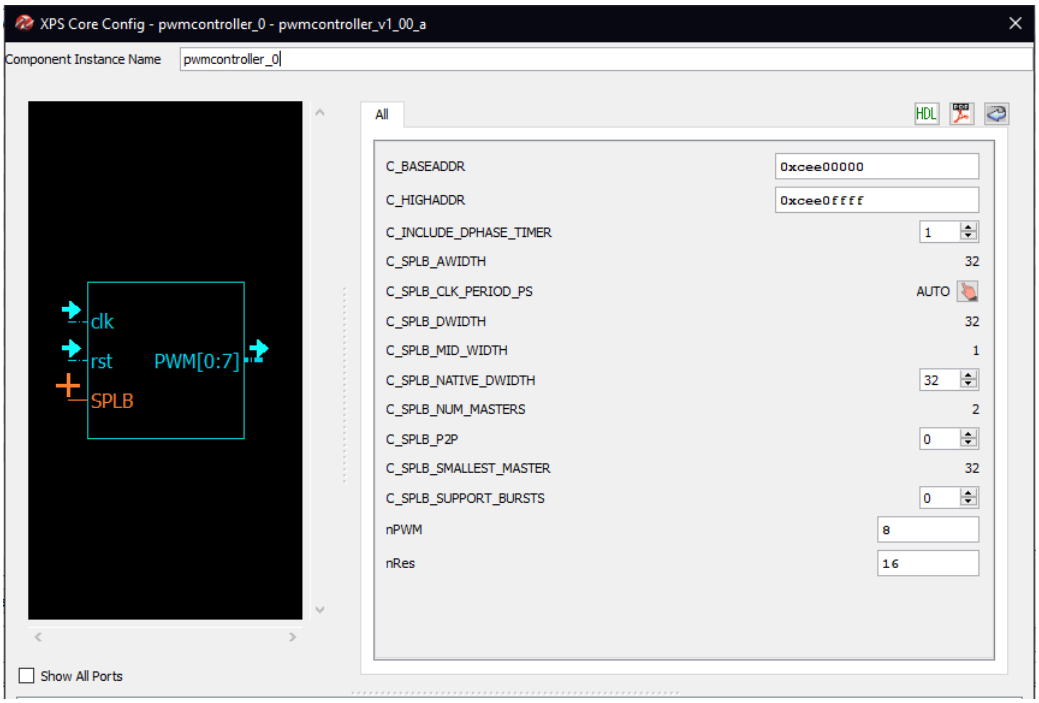
4.2. MicroBlaze i Xilinx XPS

MicroBlaze je procesorska jezgra prilagođena za korištenje ugradbenih aplikacija Xilinx alata. Može se koristiti u različitim kombinacijama dodatnih periferija i memorije. Postoje 3 načina rada MicroBlaze procesora:

- mikrokontrolerski način rada
- u stvarnom vremenu (engl. *Real-Time*)
- aplikacijski način rada.

Xilinx Platform Studio predstavlja razvojni alat koji omogućuje razvoj ugradbenih računalnih sustava varijabilne kompleksnosti. Sadrži grafičko sučelje za konfiguriranje i međusobno povezivanje sustava baziranih na ugradbenim procesorima. BSB (engl. *Base System Builder*) je alat koji se nalazi unutar XPS-a i nudi korisniku opciju implementacije MicroBlaze procesora. Koristeći BSB korisnik je u mogućnosti prilagoditi parametre razvijenog sustava. Uz prilagodbu parametara , također je omogućeno korištenje IP jezgri koje posjeduju različite funkcionalnosti. IP jezgre omogućuju korištenje različitih sklopova unutar razvojnog sustava, poput tipkala, pokaznika, i ostalih periferija [7].

Kako bi se zadovoljili potrebni uvjeti te kako bi ovakav pristup korisnički generiranih PWM signala bio moguć, potrebno je realizirati prilagođenu (engl. Custom) IP jezgru. Koristeći XPS paket alata dizajnirana je prilagođena IP jezgra pomoću EDK koja je potom integrirana unutar MicroBlaze ugradbenog sustava. Prilagođena IP jezgra kreirana je koristeći „Create and Import Peripheral“ čarobnjak unutar koje su određeni parametri za jezgru. Osnovu jezgre predstavlja ranije spomenuti VHDL modul, čijim parametrima se pristupa preko takozvanih slave sabirnica. Za izbor tipa sučelja prilagođene IP jezgre odabrana je PLB sabirnica koja predstavlja standard u povezivanju MicroBlaze procesora sa IP jezgrom [13].



Sl. 4.3. Konfiguracija IP jezgre.

PLB sabirnica temelji se na Master/Slave konfiguraciji. Sabirnica može imati više master-a i više slave-ova. U ovoj iteraciji dizajna MicroBlaze predstavlja master PLB, a ostali periferni uređaji su slave uređaji. Master kontrolira protok podataka i inicijalizaciju transakcija, a ostale periferije čekaju razmjenu podataka od strane master-a. Master sabirnica šalje zahtjeve na određeni adresni prostor, a u slučaju da je adresa prepoznata, slave periferna jedinica dekodira tu adresu i šalje odgovor na primljeni zahtjev. Adresa svakog perifernog uređaja je jedinstvena i sve transakcije jesu memorirano mapiranje. U procesu kreiranja prilagođene IP jezgre, može se dodati VHDL logički blok za slave, master te ukoliko je potrebno i za oboje. Na slici 4.3 je prikazana konfiguracija IP jezgre unutar razvojnog okruženja XPS. Za potrebe ovog diplomskog rada dizajnirana je IP jezgra sa pristupom od 32 slave sabirnice, dok čarobnjak nudi mogućnost kreiranja do 4096 slave sabirnica [14].

4.3. PCPS PWM upravljač s MicroBlaze

Kako bi se PCPS VHDL modul implementirao unutar računalnog sustava, promijenjena je automatski generirana logika vezana sa slave registre. Sustav je automatski generirao signal za svaki pojedini slave registar što nije poželjno iz perspektive jednostavnosti korištenja. Umjesto toga, slave registri su prilagođeni kao array tip podataka, veličine identične onoj automatski zadanoj (32 registra veličine 32 bita). Na taj način je omogućeno čitanje/pisanje podataka u

obliku petlje. Svaki bit signala je konfiguriran tako da odgovara jednom softverski dostupnom registru, kao što je prikazano u kodu 4.4 [15].

Linija ***Kod***

```
347:    case slv_reg_write_sel is
348:      when "10000000000000000000000000000000" =>
349:        for byte_index in 0 to (C_SLV_DWIDTH/8)-1 loop
350:          if ( Bus2IP_BE(byte_index) = '1' ) then
351:            slv_reg(0)(byte_index*8 to byte_index*8+7)
352:            <=Bus2IP_Data(byte_index*8 to byte_index*8+7);
352:          end if;
353:        end loop;
```

Sl. 4.4. Primjer upisivanja podataka u registre.

Korisničko sučelje realizirano je na način da korisnik pokretanjem C datoteke naredbom „Program FPGA“ pokreće sustav. Pokretanjem sustava korisniku se nudi mogućnost upisivanja vrijednosti za Tduty i Tdelay ovisno o parametrima za rezoluciju i broj izlaznih PWM kanala. Vrijednosti koje se upisuju su namijenjene da se upisuju kao dekadske vrijednosti koje program upisivanjem u registre konvertira u binarne vrijednosti. Upisivanje vrijednosti je ograničeno generičkim vrijednostima kako ne bi došlo do neispravnog upisivanja u registre, odnosno ukoliko korisnik upiše vrijednost koja je pre velika s obzirom na rezolucijsku varijablu, sustav će umjesto toga unijeti maksimalan broj sukladno rezoluciji. Podatci se unose pomoću računalne tipkovnice koristeći UART (engl. *Universal asynchronous receiver-transmitter*) za komunikaciju između računala i razvojnog sustava. Mogućnost upisivanja vrijednosti u registre ostvarena je koristeći UART periferni sklop. S obzirom da se očitavanje vrijednosti putem UART-a očitava kao primanje jednog byte-a, što rezultira jednoznamenkastim brojem, potrebno je koristiti

Linija ***Kod***

```
40:       for (i = 0; i < nPWM; i++) {
41          digitCounter = 0;
42:         xil_printf("\r\nUnesi %d. vrijednost Tduty: \r\n", i + 1);
43:         temp1 = 0;
44:         while(temp1!=65501)
45:     {
46:           temp1 = XUartLite_RecvByte(XPAR_UARTLITE_1_BASEADDR) - '0';
47:           recvDigits[digitCounter++] = temp1;
48:     }
49:         temp2 = digits2num(recvDigits,digitCounter);
        …
```

Sl. 4.5. Funkcionalnost upisivanja vrijednosti sa tipkovnice u registre.

funkciju koja će od više brojeva spremiti potrebnu vrijednost u registar. Kod koji prikazuje dio funkcionalnosti prikazan je na slici 4.5 [6].

Kreiranjem prilagođenog IP-a sustav automatski generira header datoteke u kojima se nalaze funkcije potrebne za upis/čitanje slave registara. Funkcija za upisivanje u registar prikazana je na slici 4.6, a parametre koje prima su osnovna adresa sklopa, pomak adrese u odnosu na osnovnu, te vrijednost koja se upisuje u registar. Za čitanje iz registara koristi se funkcija „cusotomipname_mReadReg“.

Linija Kod

```
40:          PWMCONTROLLER_mWriteReg(XPAR_PWMCONTROLLER_0_BASEADDR,
41             PWMCONTROLLER_USER_SLV_SPACE_OFFSET+ i*4, LimitDuty);
42:              } else {
43:                PWMCONTROLLER_mWriteReg(XPAR_PWMCONTROLLER_0_BASEADDR,
44:                PWMCONTROLLER_USER_SLV_SPACE_OFFSET+ i*4,temp2);
        …
```

Sl. 4.6. Funkcije za upis u registre.

Kreiranje višeznamenkastog broja realizira se koristeći funkciju na slici 4.7. Upisivanjem broja funkcija ga računa kao broj najveće dimenzije te zbraja do sada ukupan broj s njime. Kako bi ova funkcija radila kako treba bilo je potrebno provjeriti vrijednosti koje sustav šalje preko UART-a korisniku. S obzirom da se šalju ASCII vrijednosti bilo je potrebno provjeriti vrijednost koja se šalje pritiskom tipke enter na tipkovnici. Na taj način se funkcija izvršava dok korisnik ne pritisne enter, i ukupni broj koji je generiran do pritiska tipke enter se sprema u određeni registar [16].

Linija Kod

```
96:     uint32_t digits2num(uint16_t *digits, int digitCounter)
97:     {
98:        int i=0;
99:        uint32_t res = 0;
100:       for(i=digitCounter-2;i>=0;i--)
101:       {
102:         res = res + pow(10, digitCounter-2-i)*digits[i];
103:       }
104:       return res;
105:    }
```

Sl. 4.7. Funkcija za rad s brojevima.

Periferni sklopovi korišteni za ovaj rad su :

- Push_Buttons_4Bits
- RS232_Uart_1
- dlmb_cntlr
- ilmb_cntlr.

Tipkala su implementirana primarno zbog funkcije resetiranja sustava, ukoliko korisnik krivo unese neki podatak ili želi unijeti neke promjene u samim parametrima. Periferni sklop RS232_Uart_1 se koristi za komunikaciju između računala i same Nexys3 platforme. Prije pokretanja FPGA korisnik treba ostvariti konekciju sa odgovarajućim terminalom kako bi mogao komunicirati sa platformom. Sklopovi dlmb_cntlr i ilmb_cntlr koriste se u sklopu razmjene podataka (engl. *Data*) i instrukcija (engl. *Instruction*) pomoću LMB (engl. *Local Memory Bus*) sabirnice.

4.4. PWM s XPS Timer/Counter jezgrom

Kako bi se provjerila efikasnost realiziranog sustava i usporedila sa postojećim rješenjima kreiran je zaseban sustav koji realizaciju PWM signala temelji na korištenju „XPS Timer/Counter“ IP jezgre pri kreiranju projekta. Ovaj periferni sklop koristi se kao „go-to“ rješenje za kreiranje izlaznog PWM signala unutar XPS razvojnog okruženja. Ovaj periferni sklop ima 3 moguća načina rada:

- generate mode
- capture mode
- Pulse Width Modulation (PWM) mode.

Za potrebe ovog rada korišten je PWM način rada. U PWM načinu rada, dva različita brojača koriste se kao par kako bi se postigao izlazni PWM signal s određenom frekvencijom i faznim pomakom. Timer0 postavlja period dok Timer1 postavlja vrijeme u kojem je vrijednost „1“ za izlaz PWM0. Kako bi se ovaj periferni sklop koristio u PWM načinu rada potrebno je postaviti različite parametre unutar registara koji sadrže kontrolne i statusne bitove za Timer0 i Timer1. To su TCSR0 I TCRS1 registri (engl. *Timer/Counter Control/Status Register*). Modul neće raditi u PWM načinu rada ukoliko registri nisu korektno postavljeni. Period PWM -a kao i fazni pomak određuje se pomoću vrijednosti dobivenih iz TLR (engl.*Timer Load Register*) registara za oba timer-a. Period je određen vrijednošću generate u TLR0 registru, vrijeme trajanja vrijednosti logičke jedinice PWM-a je određena vrijednošću generate u registru opterećenja Timer1 (TLR1).

PWM period i radni ciklus određeni su formulama 4.1 i 4.2 kada su brojači konfigurirani da broje prema gore, a 4.3 i 4.4 kada broje prema dolje, gdje je PLB_CLOCK_PERIOD vrijednost radnog takta PLB sabirnice, a MAX_COUNT je maksimalna vrijednost brojača.

$$PWM_PERIOD = (MAX_COUNT - TLR0 + 2)\, x\, PLB_CLOCK_PERIOD \quad (4.1.)$$

$$PWM_HIGH_TIME = (MAX_COUNT - TLR1 + 2)\, x\, PLB_CLOCK_PERIOD \quad (4.2.)$$

$$PWM_PERIOD = (TLR0 + 2)\, x\, PLB_CLOCK_PERIOD \quad (4.3.)$$

$$PWM_HIGH_TIME = (TLR1 + 2)\, x\, PLB_CLOCK_PERIOD \quad (4.4.)$$

Na slici 4.8 prikazan je dio koda unutar kojeg se postavljaju vrijednosti kako bi se postigao PWM način rada. Korištenjem ovog načina rada, varijable TmrCtr0 i TmrCtr1 se koriste za podešavanje parametara PWM izlaznog signala. Vrijednosti za ove varijable su izračunate koristeći formule koje su ranije spomenute i koje su u skladu sa službenim Xilinx priručnikom za ovaj periferni sklop. Unošenje vrijednosti za Timer0 i Timer1 postiže se koristeći funkciju „XTmrCtr_SetLoadReg“, unutar koje se u trećem parametru unose vrijednosti u obliku heksadekadskog oblika. Rezultat je jedan izlazni PWM signal [17].

Linija ***Kod***

```
254:      XTmrCtr_WriteReg(XPAR_XPS_TIMER_0_BASEADDR, TmrCtr0,
        XTC_TCSR_OFFSET,
        XTC_CSR_ENABLE_PWM_MASK|XTC_CSR_EXT_GENERATE_MASK);
255:      XTmrCtr_WriteReg(XPAR_XPS_TIMER_0_BASEADDR, TmrCtr1,
        XTC_TCSR_OFFSET,
        XTC_CSR_ENABLE_PWM_MASK|XTC_CSR_EXT_GENERATE_MASK);
```

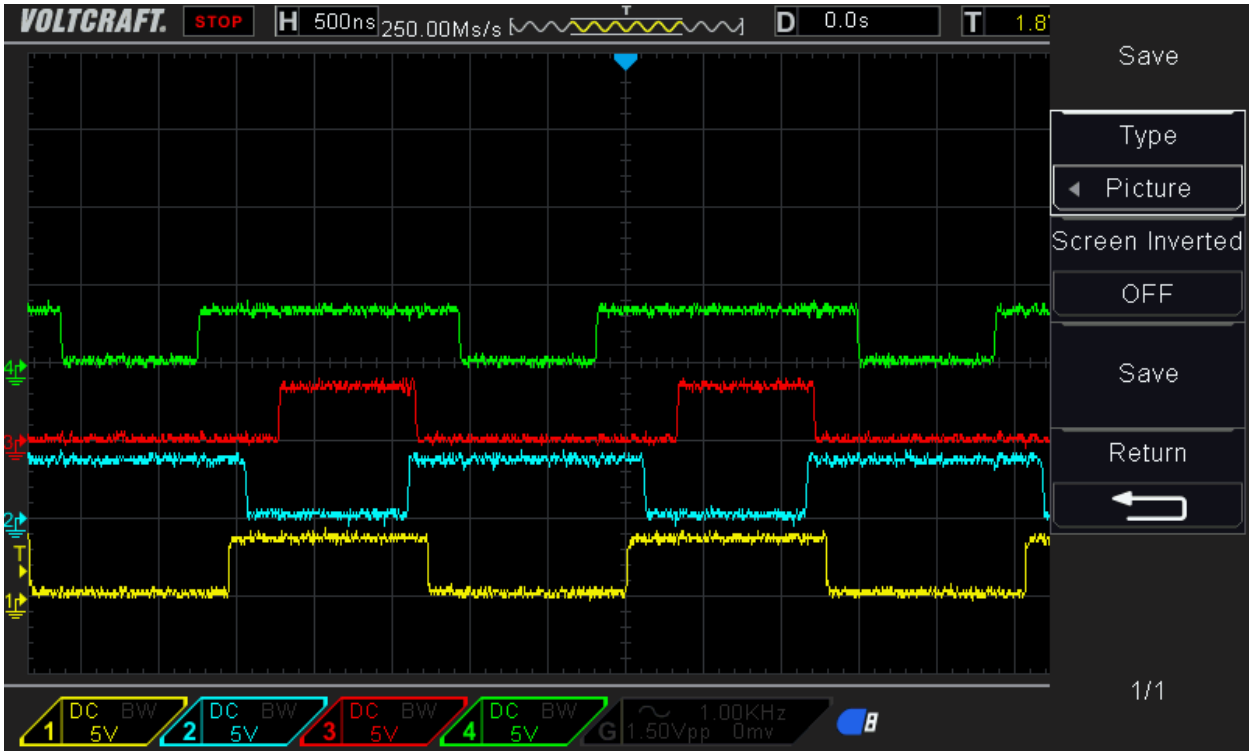
Sl. 4.8. Postavljanje PWM načina rada.

5. TESTIRANJE I ANALIZA REZULTATA

Kako bi se dobio bolji uvid u ostvarene rezultate i njihovu efikasnost, dizajnirani sustav je temeljito testiran i naposlijetku direktno uspoređen sa već poznatim rješenjem. Tijekom faze testiranja najveća pozornost usmjerena je na točnost podataka, performanse te razlike u načinu korištenja.

5.1. Testiranje VHDL modula

Početno testiranje je bilo napravljeno za samo VHDL PCPS PWM rješenje. Testiran je broj PWM izlaza, točnost očekivanih vrijednosti, provjeru kašnjenja ostalih PWM kanala u odnosu na referentni, kao i proračun frekvencije kako bi se sa sigurnošću moglo potvrditi da modul radi zadovoljavajuće. Modul je testiran sa različitim parametrima i usporedbama za 1, 4 te 8 PWM izlaza. Na slici 5.1 prikazan je zaslon osciloskopa u testiranju sa 4 PWM izlaza. Vrijednosti pomoću kojih se dobila navedena slika se nalaze u tablici 5.1, pri čemu je posljednja (4.) vrijednost referentna vrijednost. Referentna vrijednost je na slici prikazana zelenom bojom.

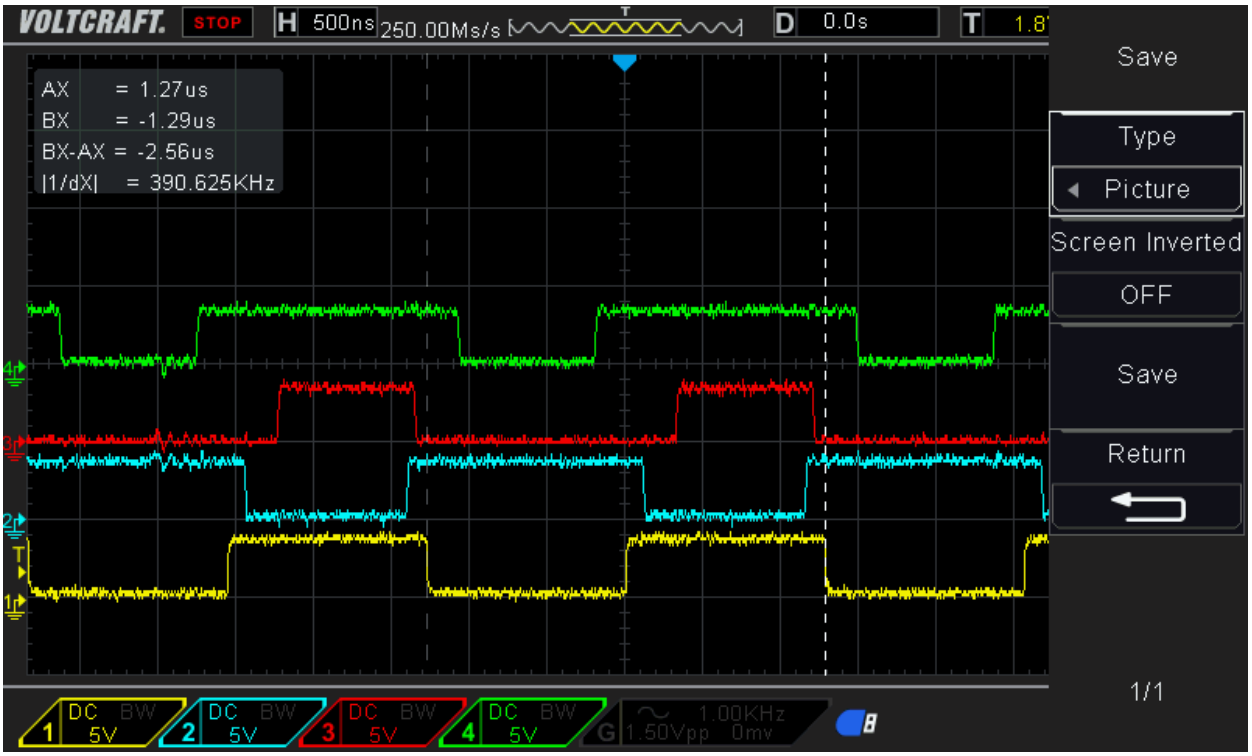


Sl. 5.1. Prikaz vrijednosti za 4 PWM izlaza za PCPS PWM VHDL modul.

Tablica 5.1. Vrijednosti testiranja VHDL modula.

PWM izlaz	*1*	*2*	*3*	*4*
Vrijednost Tduty	*10000000*	*10011000*	*01011000*	*10101000*
Vrijednost Tdelay	*00000000*	*01111111*	*00001100*	*00000000*

Kako bi se dodatno provjerila točnost modula, korištena je funkcija ugrađena u sam osciloskop pomoću koje se određuje frekvencija na temelju 2 točke, odnosno vremena u prikazanom okviru. Dobivene vrijednosti zadovoljavaju traženu točnost. Korištenje funkcionalnosti prikazano je na slici 5.2.



Sl. 5.2. Provjera točnosti za ranije spomenute parametre.

5.2. Testiranje PCPS PWM s MicroBlaze računalnim sustavom

Pokretanjem programa kroz serijsku komunikaciju ostvarenu pomoću UART kabla korisniku se pojavljuje poruka „Unesi 1. vrijednost Tduty:“, kao što je prikazano na slici 5.3. . Broj Tduty vrijednosti ovisi o generičkom parametru nPWM, odnosno brojem PWM izlaza koje korisnik očekuje. Od korisnika koji prilagođava ovo rješenje vlastitim potrebama se očekuje da poznaje proračune potrebne za dobivanje rezultata. Ukoliko korisnik upiše vrijednost koja je iznad gornje granice koju sustav s parametrima može postaviti u registar, unosi se maksimalna vrijednost

(LimitDuty ili LimitDelay). Sličan princip unosa podataka realiziran je i za komponentu kašnjenja TDelay. Razlika je u gornjoj granici, odnosno kašnjenje je prilagođeno da maksimalna vrijednost bude za jedan eksponent manji u usporedbi sa granicom TDuty. Unošenje svake vrijednosti pomoću računalne tipkovnice automatski se prikazuje na osciloskopu. Korištenjem tipke za resetiranje sustava na Nexys3 ploči sve se vrijednosti postavljaju na početnu vrijednost 0 te upisivanje podataka započinje od 1. vrijednosti za Tduty.

```
Unesi 1. vrijednost Tduty:
121
Unesi 2. vrijednost Tduty:
226
Unesi 3. vrijednost Tduty:
12
Unesi 4. vrijednost Tduty:
58
Unesi 5. vrijednost Tduty:
59
Unesi 6. vrijednost Tduty:
60
Unesi 7. vrijednost Tduty:
12
Unesi 8. vrijednost Tduty:
222
------------------------

Unesi 1. vrijednost Tdelay:

Limit za rezolucijsku varijablu 8 je 127, spremam LimitDuty
Unesi 2. vrijednost Tdelay:
127
Unesi 3. vrijednost Tdelay:
126
Unesi 4. vrijednost Tdelay:
15
Unesi 5. vrijednost Tdelay:
20
Unesi 6. vrijednost Tdelay:
25
Unesi 7. vrijednost Tdelay:
30
Unesi 8. vrijednost Tdelay:
35
```

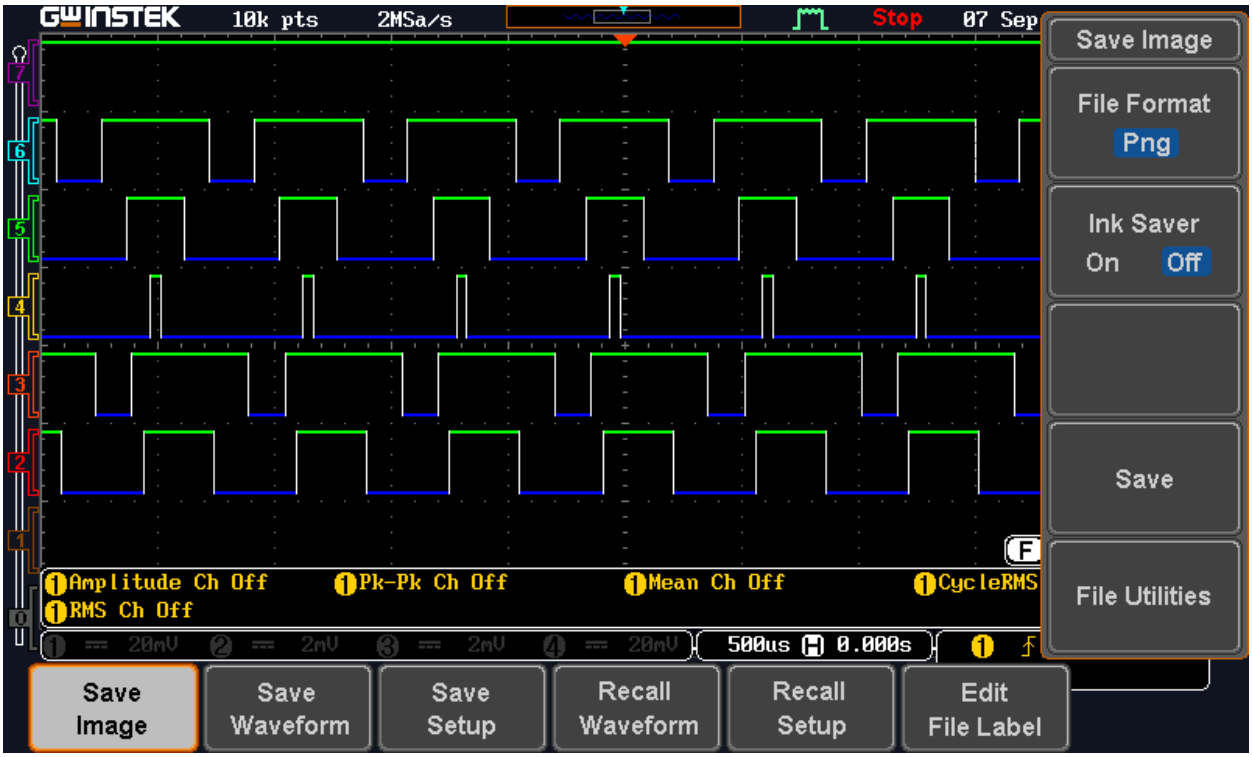
Sl. 5.3. Prikaz korisničkog sučelja prilikom unosa podataka.

U svrhu testiranja realiziranog sustava i kako bi se provjerila točnost upisanih podataka, nakon upisivanja svih potrebnih podataka sustav šalje poruku „Operacija obavljena!“. Nakon te poruke slijedi popis svih slave registara zajedno sa vrijednostima za svaki pojedini registar. Prikazuju se vrijednosti kako ih je korisnik upisao, a ne njihova binarna vrijednost. Ova funkcionalnost prikazana je na slici 5.4.

```
*Operacija obavljena!*
slv_reg(0)=121
slv_reg(1)=226
slv_reg(2)=12
slv_reg(3)=58
slv_reg(4)=59
slv_reg(5)=60
slv_reg(6)=12
slv_reg(7)=222
slv_reg(8)=0
slv_reg(9)=0
slv_reg(10)=0
slv_reg(11)=0
slv_reg(12)=0
slv_reg(13)=0
slv_reg(14)=0
slv_reg(15)=127
slv_reg(16)=127
slv_reg(17)=126
slv_reg(18)=15
slv_reg(19)=20
slv_reg(20)=25
slv_reg(21)=30
slv_reg(22)=35
slv_reg(23)=0
slv_reg(24)=0
slv_reg(25)=0
slv_reg(26)=0
slv_reg(27)=0
slv_reg(28)=0
slv_reg(29)=0
slv_reg(30)=0
slv_reg(31)=0
```

Sl. 5.4. Prikaz korisničkog sučelja nakon unosa podataka.

Rezultati se na osciloskopu prikazuju čim dođe do promjene u registrima, odnosno kada korisnik unese neki od parametara. Testiranjem rada računalnog sustava detaljno je provjerena točnost za 1, 4 ili 8 izlaza. Kako bi se usporedio računalni sustav sa početnim VHDL modulom, zadani su isti parametri za modul i PCPS PWM kontroler te su se usporedili rezultati. Osciloskop u oba slučaja pokazuje identične, odnosno iste rezultate. Testiranjem 8 izlaznih PWM kanala dobivena je slika 5.5, unutar koje se vide različiti PWM izlazi gdje je 1. signal referentni i vrijednost mu je 0, a posljednji signal ima konstantnu vrijednost 1.



Sl. 5.5. Prikaz PWM vrijednosti za 8 kanala.

5.3. Usporedba

5.3.1. Prednosti i nedostatci sustava

Temeljna prednost realiziranog računalnog sustava je jednostavnost korištenja višestrukih PCPS PWM izlaza jednostavnim promjenama unutar spomenutog sustava. Tu funkcionalnost rješenje koristeći dostupan IP unutar XPS-a ne može ostvariti, jer je ograničeno klasičnom vrstom PWM signala. Također je od velike važnosti činjenica da realizirani PWM kontroler koristi manje resursa od XPS Timer/Counter modula prilikom izvođenja iste funkcionalnosti, što će biti dodatno objašnjeno u ovom poglavlju. Glavni nedostatak ovog kontrolera je što se za postizanje najboljih rezultata performansi i učinkovitosti sustava korisnik treba optimalno kreirati prilagođenu IP jezgru u skladu sa zahtjevima korisnika. Tada sustav koristi točno onoliko slave registara koliko je korisniku i potrebno. Ukoliko performanse nisu glavni prioritet, korisnik je u mogućnosti dizajnirati IP jezgru s većim brojem registara te po potrebi mijenjati generičke parametre.

5.3.2. Zahtjevnost korištenja

Korištenje realiziranog sustava zahtjeva osnovno znanje o radu pulsno-širinske modulacije te proračun i međusobni odnos PWM signala u odnosu na referentni. Upisivanje korisničkih podataka sustavu prosljeđuje dekadske vrijednosti koje se konvertiraju u binarne vrijednosti. Ovaj pristup se smatra jednostavnijim s obzirom da zaobilazi potrebu da se svaki podatak mora ugrađivati izravno u programski kod, nego se podaci dobivaju iz vanjskih izvora, odnosno upisa vrijednosti pomoću tipkovnice kroz UART. Takva funkcionalnost rezultira bržim korištenjem. Korištenje XPS Timer/Counter jezgre za generiranje višestrukih PWM signala zahtijevalo bi ponovno konfiguriranje za promjenu nekog od parametara s obzirom da se mora koristiti više IP jezgri odjednom, za razliku od realiziranog PCPS PWM upravljača gdje je potrebno samo promijeniti generičke parametre. Podatci u ovom načinu rada upisuju se unutar programske datoteke prije pokretanja samog programa. U oba slučaja korisnik treba biti upoznat sa čitanjem i upisivanjem vrijednosti unutar dostupnih registara.

5.3.3. Usporedba upotrebe resursa FPGA integriranog sklopa

Koristeći alat za prikaz sažetka dizajna (engl. *Design Summary*) pružaju se pojedinosti o postavkama koje su trenutno odabrane na kartici dizajna unutar XPS-a, uključujući vrstu dizajna, pojedinosti o odzivu/faktoru, blokove, zauzeće memorije i ostale vrijednosti. Ti parametri su potrebni kako bi se testirale performanse i samim time njihova potencijalna upotreba izvan testnog okruženja ukoliko su rezultati zadovoljavajući. Temelji FPGA su bistabili i tablice za pretragu (engl. *Lookup Table*), skraćeno LUT-s. Kombinacijom bistabila i LUT-a nastaju logički blokovi. Performanse su provjerene za početni VHDL modul, postojeće rješenje od strane XPS-a te realizirani računalni sustav u različitim postavkama rada. Prikazane tablice sastoje se od dva dijela. Prvi dio je „XPS Synthesis Summary“, gdje su prikazana procijenjene vrijednosti uporabe bistabila i LUT tablica u različitim modulima unutar cjelokupnog računalnog sustava. U drugom dijelu nalazi se „Device Utillization Summary“, gdje su prikazane stvarne vrijednosti korištenih registara i ostalih elemenata u odnosu na dostupan broj tih elemenata unutar sustava. Odnos korištenih i dostupnih elemenata provjerava iskorištenje računalnog sustava i njegovu efikasnost.

Tablica 5.4. Sažetak dizajna za modul XPS Timer/Counter za jedan PWM.

XPS sažetak sinteze (procijenjene vrijednosti)		
Izvještaj	*Iskorišteni bistabili*	*Iskorišteni LUTs*
system	*2546*	*2654*
system_xps_timer_0_wrapper	*279*	*228*
system_proc_sys_reset_0_wrapper	*69*	*55*
system_mdm_0_wrapper	*123*	*126*
system_mem_bus_mux_0_wrapper		*57*
system_push_buttons_4bits_wrapper	*98*	*52*
system_rs232_uart_1_wrapper	*149*	*153*
system_ilmb_cntlr_wrapper	*2*	*6*
system_dlmb_cntlr_wrapper	*2*	*6*
system_dlmb_wrapper	*1*	
system_ilmb_wrapper	*1*	
system_mb_plb_wrapper	*155*	*373*
system_MicroBlaze_0_wrapper	*1667*	*1598*
Sažetak iskorištenosti sustava (stvarne vrijednosti)		
Iskorištenost logičkih registara	*Korišteni*	*Dostupni*
Broj logičkih registara	*1,655*	*18,224*
Korišteni kao bistabili	*1,648*	
Korišteni za I/ILI kogiku	*7*	
Broj LUTs-a	*1,976*	*9,112*
Korišteni za logiku	*1,790*	*9,112*
Korišteni za O6 izlaz	*1,444*	
Korišteni za O5 izlaz	*42*	
Korišteni za O5 i O6 izlaz	*304*	
Korišteni za memoriju	*145*	*2,176*

Primjer sažetka dizajna za XPS Timer/Counter modul za 1 PWM izlaz prikazan je u tablici 5.4. Vrijednosti provjerene za postojeći XPS Timer/Counter modul predstavljaju referentne parametre sažetka dizajna. Cilj ovog rada je bio dizajnirati rješenje koje uz slične ili bolje performanse nudi istu/dodatnu funkcionalnost. Dizajnirani sustav je početno uspoređen koristeći istu funkcionalnost, odnosno realiziranje jednog izlaznog PWM signala. Primjer sažetka dizajna za PCPS PWM kontroler sa 1 izlaznim PWM signalom prikazan je u tablici 5.5. Uspoređujući XPS Timer/Counter modul sa PWM kontrolerom sa 1 PWM izlazom, može se primijetiti kako PWM kontroler zauzima manje bistabila i LUT-a, odnosno razvojno okruženje pretpostavlja da

Tablica 5.5. Sažetak .dizajna za modul PCPS PWM kontroler za 1 izlaz.

XPS sažetak sinteze (procijenjene vrijednosti)		
Izvještaj	*Iskorišteni bistabili*	*Iskorišteni LUTs*
system	*2364*	*2519*
system_npwm1_0_wrapper	*195*	*146*
system_proc_sys_reset_0_wrapper	*69*	*55*
system_mdm_0_wrapper	*123*	*126*
system_mem_bus_mux_0_wrapper		*57*
system_push_buttons_4bits_wrapper	*98*	*52*
system_rs232_uart_1_wrapper	*149*	*154*
system_ilmb_cntlr_wrapper	*2*	*6*
system_dlmb_cntlr_wrapper	*2*	*6*
system_dlmb_wrapper	*1*	
system_ilmb_wrapper	*1*	
system_mb_plb_wrapper	*155*	*371*
system_MicroBlaze_0_wrapper	*1667*	*1598*
Sažetak iskorištenosti sustava (stvarne vrijednosti)		
Iskorištenost logičkih registara	*Korišteni*	*Dostupni*
Broj logičkih registara	*1,605*	*18,224*
Korišteni kao bistabili	*1,598*	
Korišteni za I/ILI kogiku	*7*	
Broj LUTs-a	*1,899*	*9,112*
Korišteni za logiku	*1,713*	*9,112*
Korišteni za O6 izlaz	*1,368*	
Korišteni za O5 izlaz	*42*	
Korišteni za O5 i O6 izlaz	*303*	
Korišteni za memoriju	*141*	*2,176*

računalni sustav sa XPS Timer/Counter modulom koristi oko 7% više bistabila i 5% više LUT-a. Također je bitno naglasiti kako s obzirom da su slave registri 32-bitni, PWM kontroler može postići veću preciznost ukoliko to korisnik želi.

Povećavanjem broja PWM izlaza i rezolucijske varijable povećava se i iskorištenost resursa računalnog sustava. Sažetak dizajna za kontroler sa 4 PWM izlaza prikazan je u tablici 5.6. Realizacijom PWM kontrolera sa 4 PWM izlaza sustav u odnosu na verziju kontrolera sa 1 izlazom koristi u prosjeku 13% više bistabila i 16% više LUT-a.

Tablica 5.6. Sažetak .dizajna za modul PWM kontroler sa 4 izlaza.

XPS sažetak sinteze (procijenjene vrijednosti)		
Izvještaj	*Iskorišteni bistabili*	*Iskorišteni LUTs*
system	*2719*	*2998*
system_npwm4_0_wrapper	*452*	*571*
system_proc_sys_reset_0_wrapper	*69*	*55*
system_mdm_0_wrapper	*123*	*126*
system_mem_bus_mux_0_wrapper		*57*
system_push_buttons_4bits_wrapper	*98*	*52*
system_rs232_uart_1_wrapper	*149*	*154*
system_ilmb_cntlr_wrapper	*2*	*6*
system_dlmb_cntlr_wrapper	*2*	*6*
system_dlmb_wrapper	*1*	
system_ilmb_wrapper	*1*	
system_mb_plb_wrapper	*155*	*373*
system_MicroBlaze_0_wrapper	*1667*	*1598*
Sažetak iskorištenosti sustava (stvarne vrijednosti)		
Iskorištenost logičkih registara	*Korišteni*	*Dostupni*
Broj logičkih registara	*1,942*	*18,224*
Korišteni kao bistabili	*1,935*	
Korišteni za I/ILI kogiku	*7*	
Broj LUTs-a	*2,264*	*9,112*
Korišteni za logiku	*2,068*	*9,112*
Korišteni za O6 izlaz	*1,618*	
Korišteni za O5 izlaz	*42*	
Korišteni za O5 i O6 izlaz	*408*	
Korišteni za memoriju	*145*	*2,176*

Dodatnim povećavanjem broj izlaza sa 4 na 8 sustav očekivano troši najviše resursa u odnosu na prethodne izvedbe. Sustav koristi 16 32-bitna registra što je duplo više u odnosnu na sustav sa 4 izlaza. Sažetak dizajna za kontroler sa 8 PWM izlaza i 8 rezolucijskih varijabli prikazan je u tablici 5.7. U odnosu na PWM kontroler sa 1 izlazom ovaj računalni sustav koristi oko 25% više bistabila te oko 38% više LUT-a. U usporedbi sa PWM kontrolerom sa 4 izlaza koristi 13% više bistabila i 27% više LUT-a. Ukoliko korisnik želi, može se kreirati prilagođena IP jezgra sa

većim brojem registara te ukoliko posjeduje dovoljno resursa koristiti ovaj modul sa više od 8 izlaza i većom razinom rezolucijske varijable.

Tablica 5.7. Sažetak .dizajna za modul PWM kontroler sa 8 izlaza.

XPS sažetak sinteze (procijenjene vrijednosti)		
Izvještaj	*Iskorišteni bistabili*	*Iskorišteni LUTs*
system	*3135*	*4095*
system_npwm8_0_wrapper	*868*	*1668*
system_proc_sys_reset_0_wrapper	*69*	*55*
system_mdm_0_wrapper	*123*	*126*
system_mem_bus_mux_0_wrapper		*57*
system_push_buttons_4bits_wrapper	*98*	*52*
system_rs232_uart_1_wrapper	*149*	*153*
system_ilmb_cntlr_wrapper	*2*	*6*
system_dlmb_cntlr_wrapper	*2*	*6*
system_dlmb_wrapper	*1*	
system_ilmb_wrapper	*1*	
system_mb_plb_wrapper	*155*	*373*
system_MicroBlaze_0_wrapper	*1667*	*1598*
Sažetak iskorištenosti sustava (stvarne vrijednosti)		
Iskorištenost logičkih registara	*Korišteni*	*Dostupni*
Broj logičkih registara	*2,354*	*18,224*
Korišteni kao bistabili	*2,347*	
Korišteni za I/ILI kogiku	*7*	
Broj LUTs-a	*3,494*	*9,112*
Korišteni za logiku	*3,290*	*9,112*
Korišteni za O6 izlaz	*2,959*	
Korišteni za O5 izlaz	*48*	
Korišteni za O5 i O6 izlaz	*283*	
Korišteni za memoriju	*145*	*2,176*

6. ZAKLJUČAK

U ovom radu razvijen je VHDL modul i implementiran na FPGA razvojni sustav za višestruki fazno ispravni PWM upravljač s faznim pomakom. Realizacija PCPS PWM upravljača ostvarena je koristeći Xilinx Platform Studio razvojno okruženje koje nudi kreiranje prilagođene IP jezgre na temelju razvijenog VHDL modula. Pomoću korisničkog sučelja upisuju se vrijednosti u registre te se na taj način mijenjaju vrijednosti PWM izlaza i rezolucijske varijable. Broj PWM izlaza kao i rezolucijska varijabla predstavljene su generičkim varijablama u VHDL-u i globalnim varijablama u programskom jeziku C. Korisnik promjenom parametara u programsku kodu VHDL-a i C-a može dinamički mijenjati broj PWM izlaza kao i rezolucijsku varijablu koja određuje preciznost kašnjenja i faznog pomaka. Prilikom realizacije računalnog sustava obratila se posebna pozornost na performanse i točnost dobivenih rezultata u odnosu na očekivane vrijednosti. Testiranjem PCPS PWM upravljača i usporedbom tih rezultata s dostupnim rješenjem ponuđenim od strane Xilinx-a u obliku IP jezgre XPS Timer/Counter je utvrđeno kako PCPS PWM upravljač koristi manje računalnih resursa za izvođenje iste funkcionalnosti.

Budući rad temeljen na ovom diplomskom radu bi bio implementirati PCPS PWM upravljač s FPGA i povezati ga sa mikroupravljačkim sustavom koji bi se koristio za upravljanje tranzistorima u svrhu pokretanja automobilskog motora ili pogona, gdje bi mikroupravljač imao kontrolu realiziranim PCPS PWM upravljačem. Također je moguće dizajnirati PCPS PWM modul na tiskanoj PCB (engl. *Printed circuit bord*) pločici koja bi se koristila kao samostalni modul i koja bi imala mogućnost spajanja na različite uređaje.

LITERATURA

[1] K., Priyanka, A., Mariyammal, „DC Motor Speed Control Using PWM“, *Int J Innov Sci Res Technol*, str. 584–587, 2018.

[2] M., Fatma, M., Hamid, „PWM speed control of dc permanent magnet motor using a PIC18F4550 microcontroller“, u *IOP Conference Series: Materials Science and Engineering*, izd. 1, sv. 602, str. 012017, 2019.

[3] M., Youssef, C., Boubahri, F., Alouli, S., Fetni, „Simulation and design of a single phase inverter with digital PWM issued by an Arduino board“, *Int. J. Eng. Tech. Res.*, 2020.

[4] H. A., Shah, S. K., Shah, R., Patel, „DSP based PWM generation for high switching frequency voltage source inverter“, u *2015 International Conference on Communications and Signal Processing (ICCSP)*, str. 0517–0520, 2015.

[5] N., Corna, N., Lusardi, F., Garzetti, A., Geraci, M., Gustin, „Multi-Channel High-Resolution Pulse-Width Modulation IP-Core Implementation for FPGA and SoC Device“, u *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, str. 1–3, 2019.

[6] M. R., Khalil, S. B., Mahmood, „Designing of a pulse width modulation system using embedded system design techniques“, *J. Theor. Appl. Inf. Technol.*, izd. 1, sv. 49, str. 101–106, 2013.

[7] R., Patel, V. P., Singh, „Xilinx Implementation of Pulse Width Modulation Generation using FPGA“, *Int. J. Eng. Manag. Res. IJEMR*, izd. 1, sv. 7, str. 411–415, 2017.

[8] I., Grout, „Pulse-Frequency Modulation Signal Generation for Programmable Logic Using Python and VHDL“, u *2018 International Electrical Engineering Congress (iEECON)*, str. 1–4, 2018.

[9] R., Leuzzi *i ostali*, „High-speed machines: Typologies, standards, and operation under pwm supply“, u *2018 AEIT International Annual Conference*, str. 1–6, 2018.

[10] P., Patel, M. A., Mulla, „DSP-FPGA based real time implementation of carrier based PWM technique for an indirect matrix converter“, *Aust. J. Electr. Electron. Eng.*, str. 1–17, 2021.

[11] A., Al-Safi, A., Al-Khayyat, A. M., Manati, L., Alhafadhi, „Advances in FPGA based PWM generation for power electronics applications: Literature review“, u *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, str. 0252–0259, 2020.

[12] V. A., Pedroni, *Circuit Design with VHDL*. MIT Press: Cambridge, MA, USA, 2004.

[13] V., Moroz, „Application of Xilinx Series 7 on FPGA (XADC)“, PhD Thesis, NURE, MC&FPGA, 2019.

[14] M. R., Khalil, L. A., Mohammed, „Embedded processor system for controllable period-width multichannel pulse width modulation signals“, *Telkomnika*, izd. 1, sv. 19, str. 220–238, 2021.

[15] C., Zet, C., Fosalau, „Generating Programmable Analog Signals using FPGA“, u *2019 International Conference on Electromechanical and Energy Systems (SIELMEN)*, str. 1–4, 2019.

[16] N., Corna, N., Lusardi, F., Garzetti, A., Geraci, M., Gustin, „Multi-Channel High-Resolution Pulse-Width Modulation IP-Core Implementation for FPGA and SoC Device“, u *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, str. 1–3, 2019.

[17] „LogiCORE IP XPS Timer/Counter (v1.02a) Manual, Xilinx“. .

SAŽETAK

U ovom radu opisan je realizirani računalni sustav koji ovisno o korisnički postavljenim parametrima nudi višestruke PWM izlaze sa varijabilnom rezolucijskom varijablom, a da je pri tome računalno efikasniji od trenutno dostupnog rješenja. PCPS PWM upravljač razvijen je koristeći Nexys 3 razvojni sustav baziran na Spartan 6 FPGA, dok je korisničko sučelje realizirano u Xilinx razvojnom okruženju Software Development Kit koristeći programski jezik C. Prilikom testiranja detaljno je provjerena točnost dobivenih signala pomoću izračuna, a nakon toga se testirala iskorištenost računalnih resursa i dobiveni rezultati su se usporedili sa postojećim rješenjem.

Nakon testiranja VHDL modula, implementirano je korisničko sučelje pomoću kojega korisnik upisuje vrijednosti. Prije programiranja FPGA korisnik pomoću generičkih varijabli određuje broj PWM izlaza te rezolucijsku razinu kojom će se dodavati kašnjenje i fazni pomak. Kako ne bi došlo do nepravilnog upisivanja podataka pomoću računalne tipkovnice koristeći UART, sučelje je testirano se su postavljeni krajnji uvjeti. Korištenjem ovog načina rada PWM signala, može se kontrolirati složenije naprave poput električnih motora unutar automobila. Također, s obzirom na ostvarene performanse i njegovu učinkovitost, PCPS PWM kontroler može zamijeniti postojeće rješenje koje nudi Xilinx kroz vlastitu IP jezgru.

Ključne riječi: PWM, VHDL, XILINX, MicroBlaze, FPGA.

ABSTRACT

Title: Multi channel PWM controller based on FPGA technology.

In this thesis a computer system that offers multi-channel PWM signals with a variable resolution scale depending on the user-based parameters, while at the same time being more memory efficient and better than the current solution is described. Embedded system is designed using the Nexys 3 development platform based on a Spartan 6 FPGA, while the user interface is developed in Xilinx Software Development Kit using C programming language. During the testing phase accuracy of the results was checked in detail using calculations, as well as efficiency of the computer resources. The results were then compared to the existing solutions.

After VHDL module testing, user interface was implemented in which a user can input values. Before starting the programming sequence, user can change the number of PWM signals and the resolution scale for the delay and to use the phase correct functionality. By testing the user interface and setting the conditions, the chance to incorrectly input the values has been disabled. Using this PWM controller can have its benefits in controlling complex machines such as electric motors inside vehicles. Also, based on the performance and the efficiency of the system, this PCPS PWM controller can replace the existing solution in the form of a Xilinx Timer/Counter IP core.

Key words: PWM, VHDL, XILINX, MicroBlaze, FPGA.

ŽIVOTOPIS

Zvonimir Korman rođen je 15. siječnja 1997 godine u Osijeku. Završio je Osnovnu školu Višnjevac, i 1. Gimnaziju Osijek. Tijekom srednjoškolskog školovanja zainteresirao se za informatiku i tehnologiju. Trenutno je redovan student 2. godine diplomskog studija, smjera Automobilsko računarstvo i komunikacije na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tečno govori engleski jezik. Trenutno radi u tvrtki Atos Convergence Creators d.o.o preko studentskog ugovora.

PRILOZI

Prilozi na CD-u su:

- PWM kontroler projekt
- VHDL modul
- diplomski rad u docx formatu
- diplomski rad u pdf formatu.