Implementacija DCSK modulatora i demodulatora u FPGA

Vinogradac, Domagoj

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

IMPLEMENTACIJA DCSK MODULATORA I DEMODULATORA U FPGA

Završni rad

Domagoj Vinogradac

Osijek, 2019.

SADRŽAJ:

1. UVOD 1
2. TEORIJA KAOSA 2
2.1. Digitalna komunikacija utemeljena na kaosu 3
2.2. Diferencijalna diskretna promjena kaotičnog signala 5
2.3. Prednosti i nedostaci DCSK modulacije 7
3. FPGA i VHDL 8
3.1. Zybo Z-7000 9
3.2. VHDL 11
4. IZRADA DCSK MODULATORA I DEMODULATORA 13
4.1. DCSK modulator 13
4.2. DCSK demodulator 14
5. SIMULACIJA, IMPLEMENTACIJA I TESTIRANJE DCSK MODEMA 16
5.2. Vivado razvojno okruženje 17
5.3. UART 18
5.4. Implementacija DCSK modema na Zybo Z-7000 18
5.5. Testiranje DCSK modema 21
6. ZAKLJUČAK 23
LITERATURA 24
SAŽETAK 25
ABSTRACT 26
PRILOZI 28

1. UVOD

U sedamdesetim i osamdesetim godinama prošlog stoljeća u svijetu je došlo do velikog napretka glede znanja o nelinearnim sustavima. Kao najveće otkriće dobiveno tim znanjima, bilo je otkriće kaosa u deterministčkim sustavima, ujedno i jedno od većih otkrića u znanosti općenito. Otkriće kaosa omogućilo je opisivanje dotad neprimjetnih regularnosti u kompleksnim fizikalnim sustavima pod jedinstvenim nazivom: teorija kaosa.

Povećani interes za novootkrivenu teoriju kaosa doveo je do mnogih nastojanja iskorištavanja iste za korisnu primjenu u inženjerskim sustavima. Komunikacije su jedno od područja gdje je teorija kaosa viđena kao korisna zbog širokopojasne karakteristike kaotičnih signala. Korištenjem kaotičnih signala za prenošenje informacija dobivaju se prednosti poput otpornosti na prigušivanje, male vjerojatnosti presretanja i robusnosti u višestaznim okruženjima, a generiranje kaotičnih signala je prilično jednostavno pa stoga predstavlja jeftino rješenje za komunikaciju u širokom spektru frekvencija.

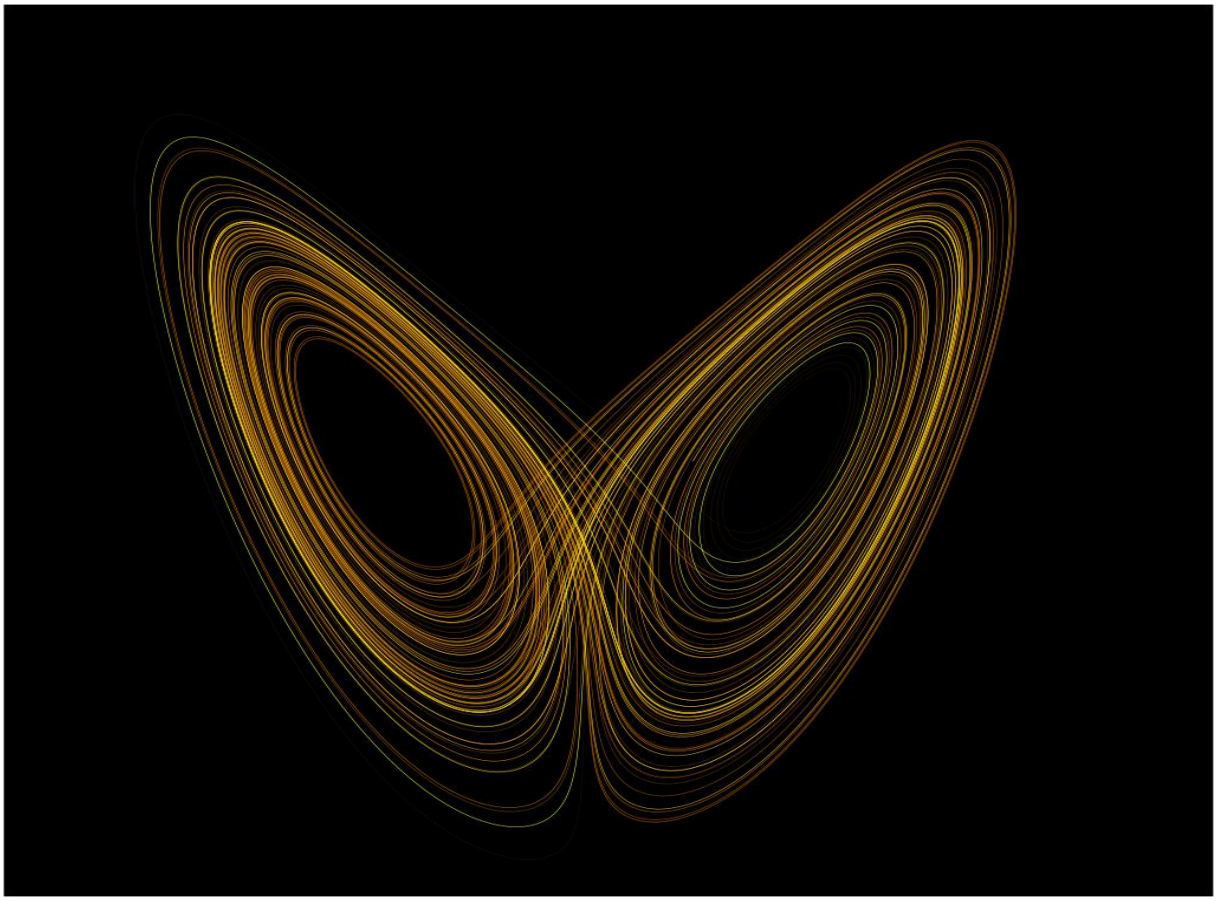
Primjenom kaotičnih signala došlo se do raznih vrsta modulacije informacija poput CSK (engl. *Chaos Shift Keying*), DCSK (engl. *Differential Chaos Shift Keying*), COOK (engl. *Chaotic On – Off Keying*), FM – DCSK (engl. *Frequency Modulated Differential Chaos Shifty Keying*) itd. Među boljima pokazala se DCSK modulacija zbog svoje otpornosti na šumove i ostale smetnje u komunikacijskom kanalu iako je sama izvedba sustava složenija [1].

Cilj ovog rada je opisati postupak DCSK modulacije i demodulacije informacija te alate korištene za opisivanje, implementaciju i testiranje iste. Za dizajn modulatora i demodulatora koristi se VHDL (engl. *VHSIC Hardware Description Language*) jezik za opisivanje sklopovlja, dok se sinteza, simulacija i implementacija izvodi na FPGA (engl. Field Programmable Gate Array) pločici Zybo Z – 7000 uz pomoć Vivado razvojnog okruženja.

U drugom poglavlju dana je teorijska podloga teorije kaosa, komunikacije utemeljene na kaosu te DCSK modulacije. U trećem poglavlju opisani su VHDL jezik za opisivanje sklopovlja i pločica Zybo Z-7000. U četvrtom poglavlju opisana je izrada DCSK modulatora i demodulatora u VHDL jeziku za opisivanje sklopovlja. Nakon opisa izrade, u šestom poglavlju opisan je proces implementacije sustava na Zybo pločicu, te testiranja i verifikacije rada. Na kraju je dan zaključak o radu.

2. TEORIJA KAOSA

Teorija kaosa je interdisciplinarna teorija koja govori o obrascima ponavljanja, fraktalima, povratnim vezama i samoorganizaciji u naizgled kaotičnom sustavu. Prvi je ovu teoriju predložio Henry Poincaré osamdesetih godina 19. stoljeća kada se bavio problemom triju tijela gdje je otkrio da orbite mogu biti neperiodičke, a da također niti zauvijek rastu niti se približavaju nekoj fiksnoj točki. Sve do šezdesetih godina 20. stoljeća i razvoja elektroničkog računala nije bilo praktično računati ponavljajuće iteracije jednostavnih matematičkih formula koje su bile opisane u matematici teorije kaosa te znanstvenici nisu mogli doći do značajnih otkrića. Nakon šezdesetih godina mnogi istraživači poput Edwarda Lorenza, Benoita Mandelbrota, Roberta Maya, Davida Ruellea i mnogih drugih došli su do raznih otkrića na poljima meteorologije, matematike, biologije, astrofizike i drugih znanosti što je dovelo do razvoja interesa u samu teoriju i daljnjih otkrića koja su pokazala prisutnost kaosa u raznim aspektima života i svemira.



Sl 2.1. Grafički prikaz Lorenzovog atraktora. [2]

Općenito govoreći teorija kaosa promatra deterministične sustave čije vladanje može predvidjeti s određenom tolerancijom. Kako bi predviđanje vladanja sustava bilo uspješno, važno je koliko nesigurnosti se može tolerirati u predviđanju, koliko precizno se njegovo sadašnje stanje može izmjeriti, te tzv. Lyapunovo vrijeme koje predstavlja vremenski raspon u kojem je sustav kaotičan. No kako se u ovim sustavima nesigurnost povećava eksponencijalno s vremenom, sve predikcije vladanja sustava preko dva do tri puta Lyapunovog vremena postaju beznačajne.

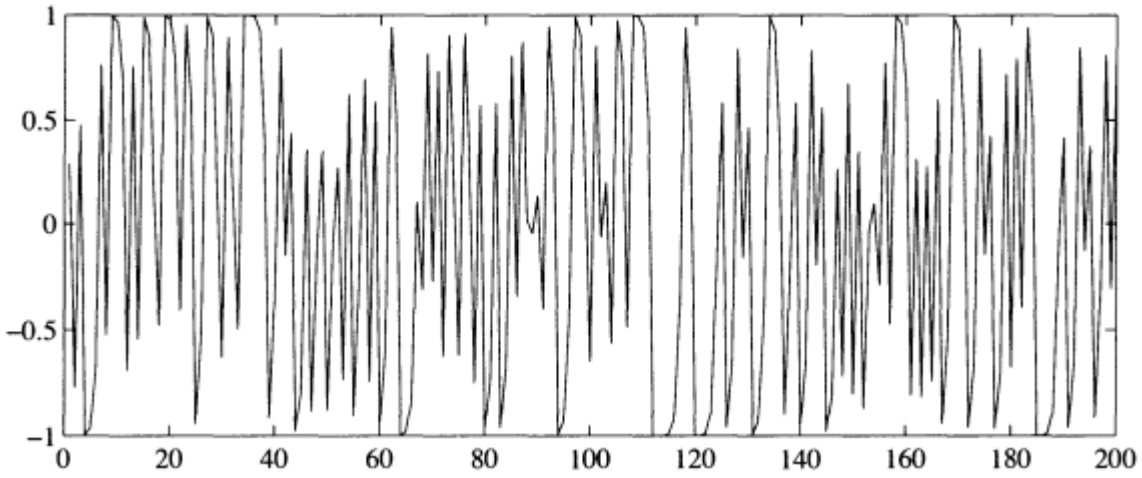
Kako ne postoji prihvaćena matematička definicija kaotičnog sustava, obično se uzima definicija Roberta L. Devaneya, profesora matematike na Bostonskom sveučilištu: „Kako bi mogli klasificirati sustav kao kaotičan, on mora imati sljedeća svojstva [3]:

1. mora biti osjetljiv na početne uvjete – mala promjena uvjeta na početku može dovesti do velikih promjena u kasnijim stanjima
2. mora biti topološki tranzitivan – označava da se bliske točke sustava nakon nekog vremena počnu udaljavati jedne od drugih
3. mora imati zbijene periodičke orbite

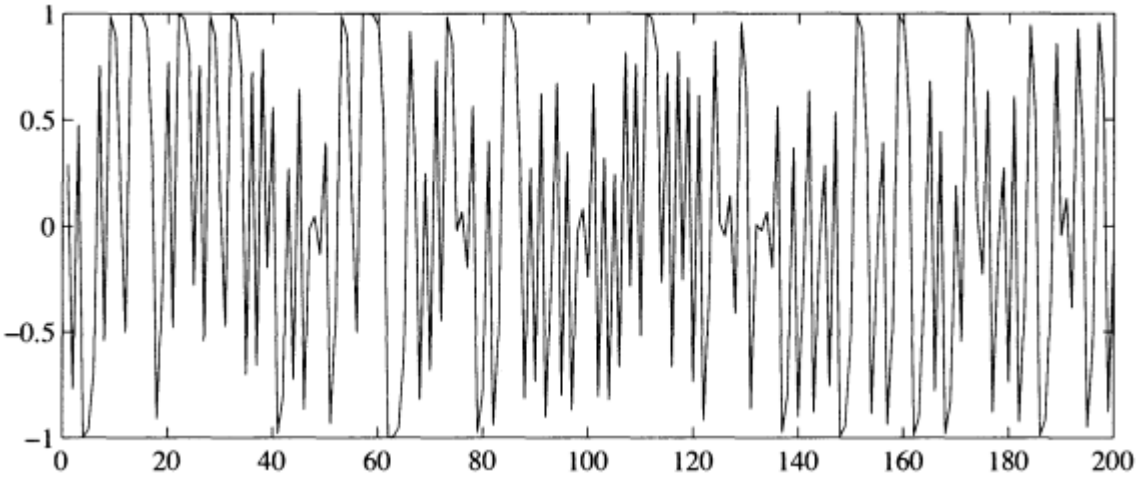
Primjena teorije kaosa do danas seže u jako puno područja poput matematike, biologije, mikrobiologije, kriptografije, računalnih znanosti, robotike, ekonomije, meteorologije i ostalih znanosti.

2.1. Digitalna komunikacija utemeljena na kaosu

Osnovu digitalne komunikacije temeljene na kaosu nose kaotični signali. Kaotični signali izvedeni su iz nelinearnih dinamičkih sustava koji imaju fiksan broj varijabli stanja. Putanje, odnosno vrijednosti koje poprimaju varijable stanja, su određene diferencijalnim jednadžbama koje sadrže te varijable stanja. Putanje varijabli stanja u kaotičnim sustavima su ograničene, ne periodičke i nalik nasumičnim, a njihovo najvažnije svojstvo je osjetljivost na početne uvjete. To svojstvo daje mogućnost generiranja bezbroj nevezanih kaotičnih signala iz jednog sustava mijenjajući samo njegove početne uvjete.



a)



b)

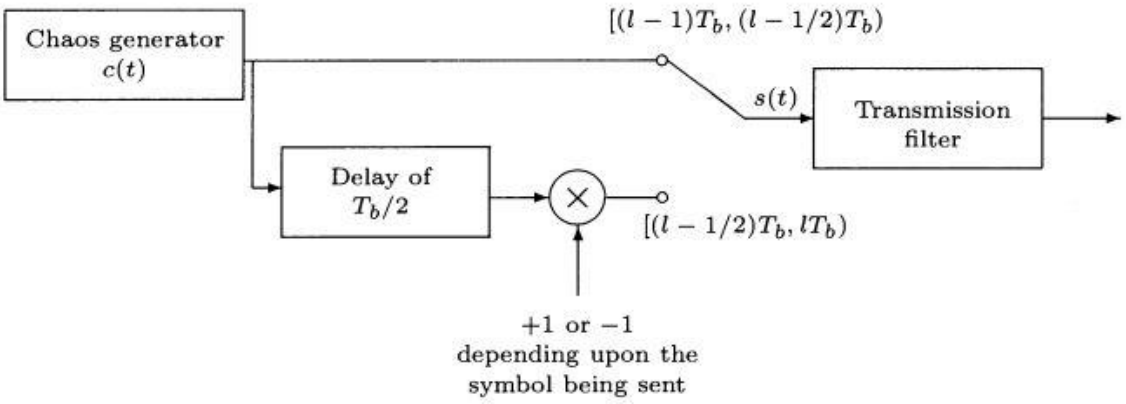
Sl 2.2. Prikaz valnih oblika kaotičnih signala vremenski diskretnog dinamičkog sustava prvog reda. Jednadžba sustava $x_n = 4x_{n-1}^3 - 3x_{n-1}$, predstavlja kubičnu mapu za koju se zna da je kaotična. Početni uvjeti sustava su a) 0.29; i b) 0.28999. [1]

Primjena kaotičnih signala pronalazi se u analognoj i digitalnoj modulaciji. U analognoj modulaciji koriste se tehnike kaotičnog maskiranja i kaotične modulacije. Kod kaotičnog maskiranja analognih signala, analogni signal se dodaje izlazu kaotičnog sustava, dok se na prijemniku pomoću sinkronizacije ponovno stvara isti kaotični signal te oduzima od primljenog signala čime se dobiva

odaslana informacija. Kod kaotičnog moduliranja, informacija se kodira u pomno odabrani parametar kaotičnog sustava koji generira kaotični signal koji tada sadrži kodiranu informaciju, a prijemna strana prati promjene u dinamici kaotičnog signala i izdvaja kodiranu informaciju. U digitalnim modulacijama postoji mnogo vrsta modulacije od kojih su bitnije COOK, CSK, DCSK i FM – DCSK [1].

2.2. Diferencijalna diskretna promjena kaotičnog signala

DCSK modulacija je ona kojom se bavi ovaj rad, a temelji se na nekoherentnom otkrivanju odnosno nije potrebna sinkronizacija odašiljača i prijemnika sustava. Odašiljačka strana generira dva uzastopna okvira kaotičnog signala, od kojih je prvi referentni, a drugi informacijski te ovisno šalje li se 1 ili -1, informacijski okvir će biti isti kao referentni odnosno invertiran. Općenito, referentni okvir se šalje u prvoj polovini perioda jednog simbola, dok se informacijski okvir šalje u drugoj polovini.

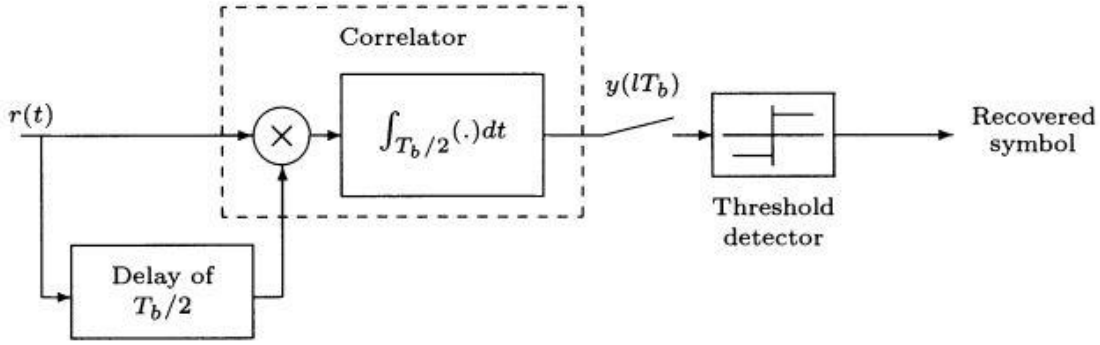


Sl 2.3. Opća blok shema DCSK modulatora. [1]

Iz ove blok sheme vidimo da se modulator sastoji od generatora kaotične sekvence c(t), koji se na jednoj strani šalje na filter za prijenos podataka preko kanala s(t), dok se na drugoj strani taj isti signal zakašnjava, za pola perioda takta slanja podatka, i množi s dolaznim bitom simbola koji se kodira. Izlaz iz odašiljača za period l-tog simbola dan je izrazom:

$$s(t) = \begin{cases} c(t) & za\ (l-1)Tb\ \leq t < \left(l - \frac{1}{2}\right)Tb \\ c\left(t - \frac{Tb}{2}\right) & za\ (l-1/2)Tb\ \leq t < lTb \end{cases} \quad (2\text{-}1)$$

gdje je s(t) izlaz iz odašiljača u trenutku t, c(t) izlaz iz generatora kaotične sekvence u trenutku t, Tb period trajanja l-tog simbola.



Sl 2.4. Opća blok shema DCSK demodulatora. [1]

Iz blok sheme demodulatora vidimo da se prvi okvir dolaznog kaotičnog signala zakašnjuje za pola perioda takta slanja podatka, te se na izlazu množi s drugim okvirom. Signal dobiven množenjem šalje se na korelator koji pak određuje vrijednost dobivenog signala dok detektor praga odlučuje koji simbol je bio poslan. Izlaz iz korelatora dan je izrazom:

$$y(lTb) = \int_{\left(l-\frac{1}{2}\right)Tb}^{lTb} r(t)r\left(t - \frac{Tb}{2}\right)\,dt \quad (2\text{-}2)$$

gdje je y(lTb) izlaz iz korelatora, Tb period trajanja l-tog simbola, r(t) vrijednost signala u trenutku t, r(t – Tb/2) vrijednost signala u trenutku t - Tb/2.

Izlaz iz korelatora uz aditivan šum dan je izrazom:

$$y(lTb) = \int_{\left(l-\frac{1}{2}\right)Tb}^{lTb} [s(t) + n'(t)]\left[s\left(t - \frac{Tb}{2}\right) + n'\left(t - \frac{Tb}{2}\right)\right]dt \quad (2\text{-}3)$$

gdje je y(lTb) izlaz iz korelatora, Tb period trajanja l-tog simbola, s(t) vrijednost izlaza iz odašiljača u trenutku t, n'(t) vrijednost šuma u trenutku t, s(t – Tb/2) vrijednost izlaza iz korelatoga u trenutku t – Tb/2, n'(t – Tb/2) vrijednost šuma u trenutku t – Tb/2.

Pri detekciji, prag detekcije se može staviti na 0 jer dobiveni simbol će se interpretirati kao 1 ili -1, neovisno razini šuma, što znači da sve integralne vrijednosti šuma imaju srednju vrijednost 0.

2.3. Prednosti i nedostaci DCSK modulacije

Prednosti DCSK sustava nad svi ostalim sustavima u kaotičnoj digitalnoj modulaciji su nekoherentno otkrivanje odnosno manjak sinkronizacije odašiljača i prijemnika, dobar odnos signala i šuma, te konstantna vrijednost optimalnog praga, koji ostaje na 0 iako su moguće greške u razlikovanju simbola u „bučnim“ komunikacijskim kanalima. Glavni nedostatak ove modulacije je prijenos informacije u samo pola perioda slanja podataka, te iako postoji način povećanja brzine s višerazinskom demodulacijom, kompliciraniji sustav i degradacija BER (engl. Bit Error Ratio) zbog prigušenja komunikacijskog kanala brzo prerastu dobivenu korist.

3. FPGA i VHDL

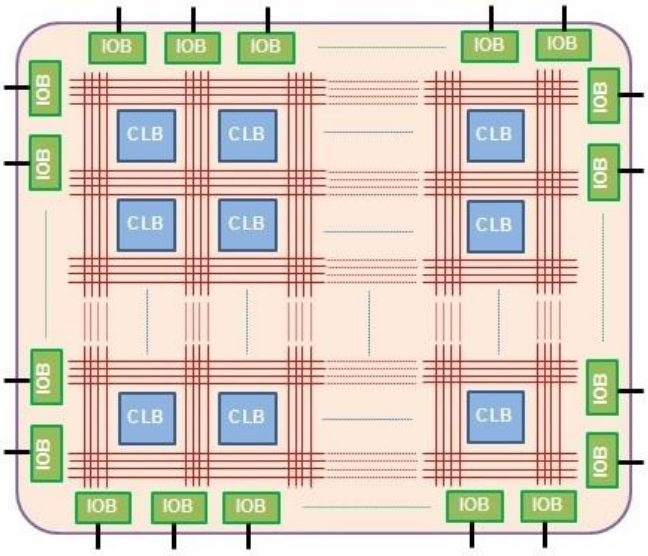
FPGA (engl. *Field-Programmable Gate Array*) je uređaj temeljen na matrici podesivih logičkih blokova povezanih pomoću programibilnih veza. FPGA se mogu konfigurirati pomoću jezika za opisivanje sklopovlja, HDL (engl. *Hardware Description Language*). Svaki FPGA sastoji se od:

- CLB-ova (engl. *Configurable Logic Block*) koji omogućavaju konfiguriranje sekvencijalne ili kombinacijske logike, sastoji se od preglednih tablica, elemenata za pohranu podataka i multipleksora

- Switch blokova koji zajedno sa sabirnicama povezuju logičke blokove u digitalni sklop koji je korisnik opisao putem HDL-a

- I / O blokova koji predstavljaju ulaze i izlaze koji povezuju digitalni sklop s vanjskim svijetom, od kojih su neki *pull-up* i *pull-down* otpornici, međuspremnici i pretvarači

Kako bi CLB-ovi komunicirali jedan s drugim i vanjskim sklopovljem, povezani su programibilnim vezama i ulazno izlaznim blokovima. Programi koje korisnik unese u FPGA se spremaju u SRAM (engl. *Static Random Access Memory*) memoriju. Na slici 3.1. vidimo način na koji su postavljeni svi navedeni elementi unutar same FPGA pločice.

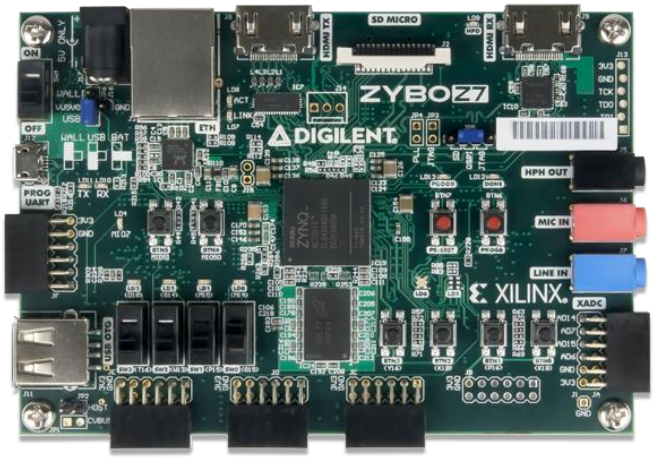


Sl 3.1. Prikaz FPGA. [4]

3.1. Zybo Z-7000

ZYBO (engl. ***ZY****nq* ***BO****ard*) je razvojna ploča razvila je kompanija Diligent Corporation. Ona se sastoji od Xilinxovog potpuno programibilnog sustava-na-čipu (*All Programmable System-on-Chip*, AP SoC) uz koji je integriran dvojezgreni ARM Cortex-A9 procesor sa Xilinxovom FPGA logikom serije 7. Za povezivanje sa vanjskim svijetom, na razvojnoj pločici nalazi se multimedijskih i konekcijskih periferija poput video i audio ulaza i izlaza, USB ulaz s dvostrukom ulogom, Ethernet ulaz, ugrađene memorije te utor za SD memorijsku karticu. Neke od karakteristika sustava-na-čipu su:

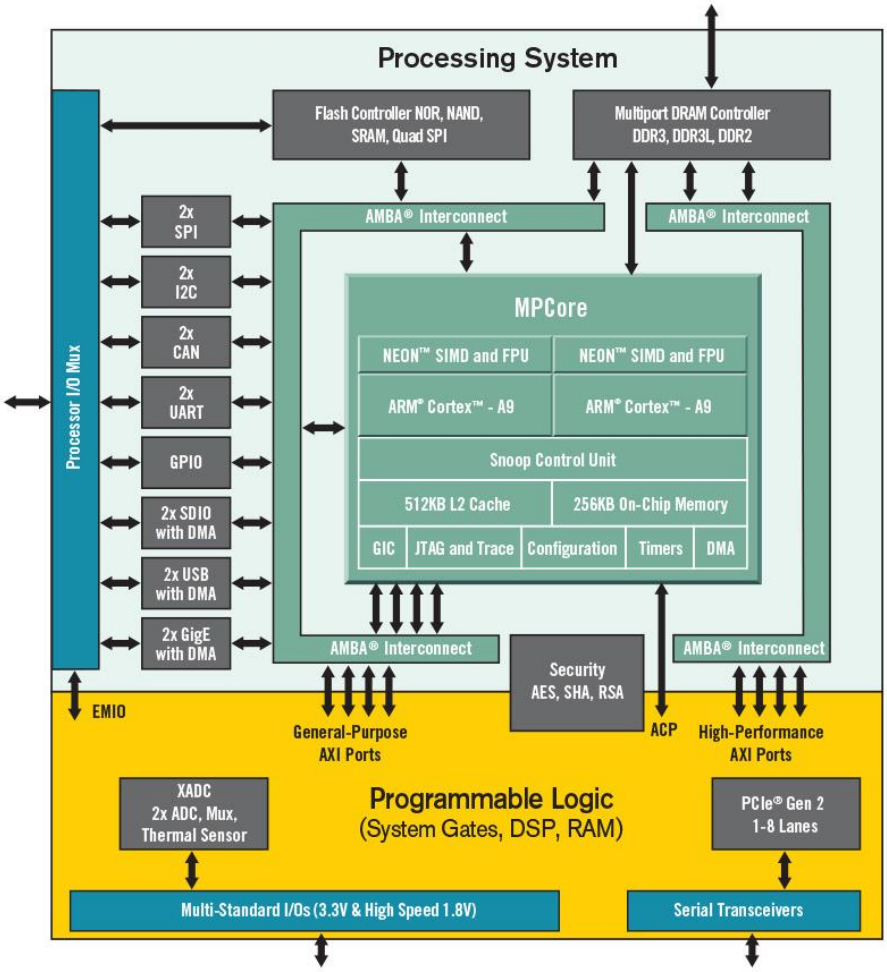
- 650 MHz dvojezgreni Cortex – A9 procesor
- DDR3 memorijski kontroler sa 8 DMA kanala
- kontrolere: 1G Ethernet, USB 2.0, SDIO
- kontrolere: SPI, UART, CAN, I^2C
- programibilnu logiku, ekvivalentnu Artix – 7 FPGA



Sl 3.2. Zybo Z7 razvojna ploča. [5]

ARM Cortex – A9 procesori korišteni u Zybo imaju 28 nanometarsku arhitekturu. Programibilna logika temeljena je na Artix – 7 ili Kintex®-7 što ju čini vrlo učinkovitom, a u sebi sadrži preko 6.6 milijuna logičkih ćelija. Ima jako puno primjena te vrlo često služi za ugrađene aplikacije poput sustava za pomoć vozaču preko više kamera, medicinske endoskope, male telefonske stanice osnovnog frekvencijskog područja, računalni vid te ostale slične primjene [5].

Na slici 3.3. vidimo arhitekturu Zybo pločice te kako su povezani svi dijelovi programibilne logike (engl. *PL*) te sustava za obradu (engl. *PS).*



Sl 3.3. Arhitektura Zybo ploče sa Zync 7000. [6]

3.2. VHDL

U prijevodu VHDL je jezik za opisivanje i dizajn digitalnih sklopova vrlo velikih brzina. VHDL je krajem 70-ih i početkom 80-ih počelo razvijati Ministarstvo obrane SAD-a u svrhu dokumentiranja ponašanja opreme odnosno integriranih krugova specijalne namjene koji su dolazili od različitih proizvođača. Nastao je iz programskog jezika ADA što je kasnije vidljivo iz same sintakse te pojedinih funkcija. Vrlo brzo nakon što je stvoren, mnogi inženjeri su se počeli njime koristiti kako bi stvarali elektroničke sustave i proizvode. Godine 1987. udruga IEEE (engl. *Institute*

of Electrical and Electronics Engineers, Institut inženjera elektrotehnike i elektronike) je ratificirala VHDL kao službeni jezik za dokumentaciju, implementaciju dizajna i funkcionalnu verifikaciju.

Pri opisivanju sklopovlja stvara se VHDL datoteka čiji su glavni dijelovi entitet (engl. *entity*) koji sadrži sve ulaze i izlaze opisanog sklopa i njegove opće vrijednosti (engl. *generic*), arhitektura (engl. *architecture*) koji pak sadrži sve signale kojima se sklop služi te ključnu riječ begin iza koje se opisuje ponašanje sklopa. I

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
use ieee.std_logic_arith.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity mux4u1 is
Port(
        inCONTROL : in std_logic_vector (1 downto 0);
        inDATA : in std_logic_vector (15 downto 0);
        outDATA : out std_logic_vector (3 downto 0)
    );
end entity;

architecture arch4u1mux of mux4u1 is
signal sDATA : std_logic_vector (3 downto 0);

begin
sDATA <= inDATA(3 downto 0) when (inCONTROL = "00") else
        inDATA(7 downto 4) when (inCONTROL = "01") else
        inDATA(11 downto 8) when (inCONTROL = "10") else
        inDATA(15 downto 12) when (inCONTROL = "11");
outDATA <= sDATA;
end architecture;
```

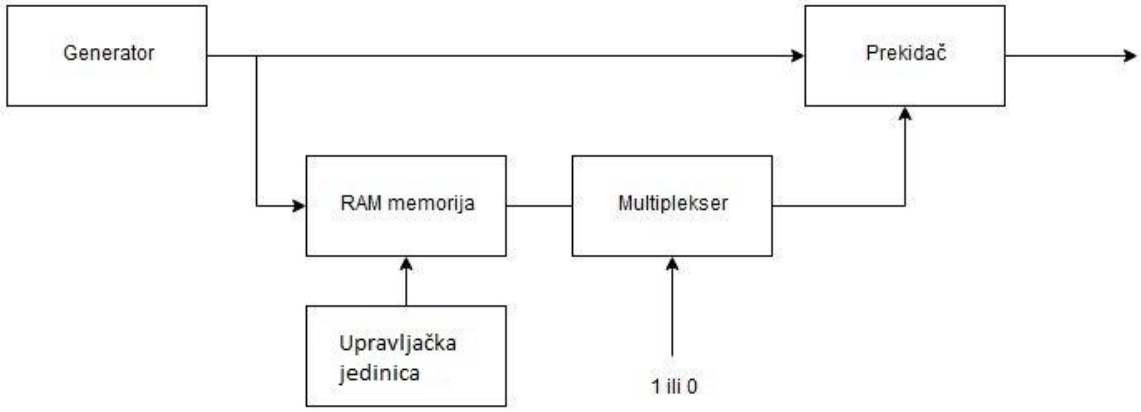
Sl 3.4. Primjer VHDL koda.

4. IZRADA DCSK MODULATORA I DEMODULATORA

Izrada odnosno opisivanje svih blokova sustava modulatora i demodulatora izvedena je u VHDL jeziku za opis sklopovlja. Svaki blok, u blok shemi, predstavlja jednu *.vhd datoteku, dok modulator i demodulator predstavljaju datoteke vrha hijerarhije u kojoj se ostale datoteke zajedno povezuju i čine sustav.

4.1. DCSK modulator

Za izradu DCSK modulatora korištena je shema na slici 4.1. koja je dobivena modifikacijom opće blok sheme DCSK modulatora sa slike 2.3.



Sl 4.1. Blok shema DCSK modulatora.

Blok „Generator“ je generator kaotičnih signala. Radi na principu LFSR (engl. *Linear Feedback Shift Registers*). Sklop je sastavljen od 12 D – bistabila kod kojih su izlazi iz prvog, četvrtog, šestog i dvanaestog spojeni na ulaz prvog preko isključivo - ILI logičkih vrata. Ovaj način spajanja omogućuje 4095 različitih iteracija sustava prije nego se vrati u početno stanje. Izlaz je paralelno izveden sa svih 12 D–bistabila što nam daje 12–bitni vektor. Veličina vektora izabrana je proizvoljno te o njoj ovisi broj iteracija kroz koje sustav može proći te rezoluciju signala koji može generirati.

Blok „RAM memorija“ (engl. *Random Access Memory*) izvedena je od 64 memorijska polja veličine 12 bitova te se u nju spremaju 64 generirana bit vektora. Ona ima mogućnost upisa i ispisa iz memorije, ovisno o stanju bita Pisanje/Čitanje. Ta 64 okvira predstavljaju informacijski okvir DCSK

modulacije. Također, o veličini bloka te memorije ovisi trajanje slanja svakog bita simbola te njegove otpornosti na šum, a time i samog simbola.

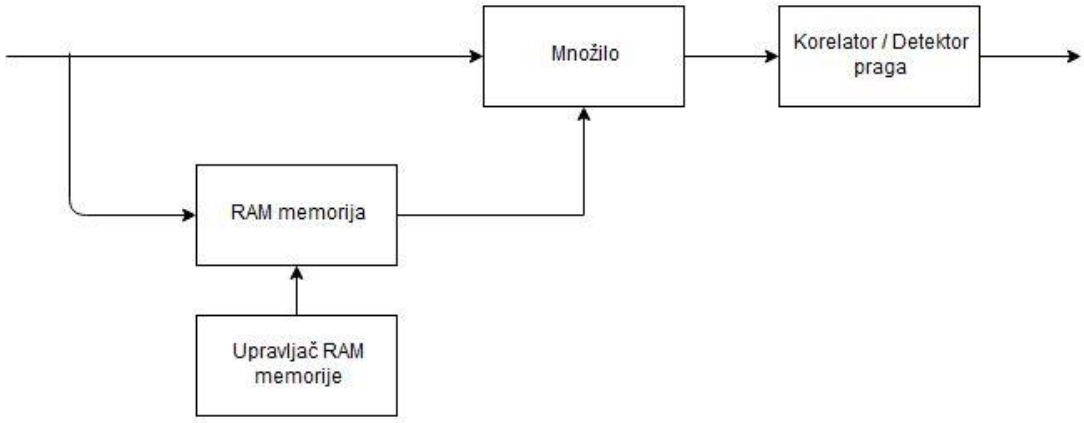
Blok „Upravljačka jedinica“ kontrolira upis i ispis RAM memorije te brine na koju adresu će se upisati pojedini podatak.

Blok „Multiplekser“ na jednom ulazu prima 12 - bitne vektore iz RAM memorije, a na drugom prima bitove simbola koji se treba modulirati. Ako je bit simbola „1“, na izlazu sklopa će biti jednak 12 - bitni vektor kao i na ulazu, dok ako je bit simbola „0“, na izlazu sklopa bit će invertirani 12 - bitni vektor onog na ulazu u sklop.

Blok „Prekidač“ predstavlja sklopku koja uklapa izlaz iz Generatora i Multipleksera. Prvo propušta 64 12-bitna vektora, odnosno referentni okvir, sa Generatora, te tada uklapa Multiplekser i propušta 64 12-bitna vektora odnosno informacijski okvir.

4.2. DCSK demodulator

Za izradu DCSK demodulatora korištena je shema na slici 4.2. koja je dobivena modifikacijom opće blok sheme DCSK demodulatora sa slike 2.4.



Sl 4.2. Blok shema DCSK demodulatora.

Blok „Množilo“ na jednom ulazu prima 12 – bitne vektore s DCSK modulatora, dok na drugom ulazu prima 12 – bitne vektore iz RAM memorije. Kada se RAM memorija napuni do zadnje

adrese i počne slati podatke, Množilo počinje množiti 12 – bitne vektore s oba ulaza uz pomoć isključivo – NILI logičkih vrata što znači da, ako su 12 – bitni vektori isti dobivamo 12 – bitni vektor sastavljen samo od „1“, dok u suprotnom dobivamo vektor sastavljen od „0“.

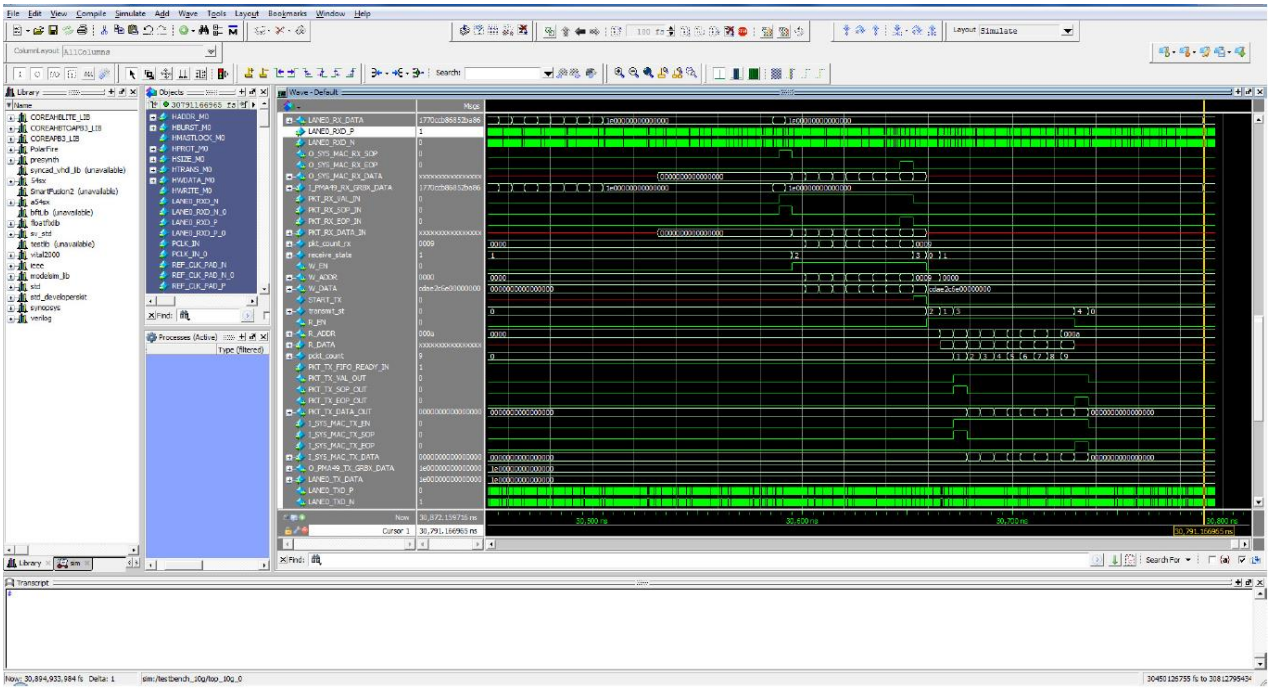
Blok „Korelator / Detektor praga“ je sklop koji prosuđuje koji bit, odnosno na kraju i simbol koji je poslan. Prag ovog detektora postavljen je na vrijednost od 128. Ako ovaj sklop primi 12 – bitni vektor od samo „1“ on inkrementira brojač koji se pri svakom resetiranju postavlja na 64, a u suprotnom, ako primi 12 – bitni vektor od samo „0“ taj brojač se dekrementira. Ako je vrijednost brojača nakon 64 primljena 12 - bitna vektora 128, možemo znati da je poslani bit „1“, no ako je vrijednost brojača 0, tada znamo da je poslan bit „0“.

5. SIMULACIJA, IMPLEMENTACIJA I TESTIRANJE DCSK MODEMA

U ovome poglavlju opisani su alati koji su korišteni za simulaciju, implementaciju i testiranje. Također je opisan i UART (engl. *Universal Asynchronous Receiver/Transmitter*) modem za serijsku komunikaciju između računala i Zybo pločice. Za simulaciju je korišten program za računalnu simulaciju ModelSim, dok je za implementaciju dizajna sustava na pločicu korišteno razvojno okruženje Vivado.

5.1. ModelSim program za računalnu simulaciju

ModelSim je program za računalnu simulaciju koji je razvila tvrtke Mentor Graphics. On služi za simulaciju jezika za opisivanje sklopovlja poput VHDL, SystemC i Verilog. Simulacija se izvodi pomoću grafičkog korisničkog sučelja ili korištenjem automatskih skripti. Također postoji više inačica ovog proizvoda od kojih je ModelSim PE verzija za studente i hobiste, dok je Questa Sim verzija sa najviše mogućnosti uz najviše performanse. Na slici 5.1. prikazano je grafičko sučelje ModelSim programa na kojem se također vidi simulacija sustava.

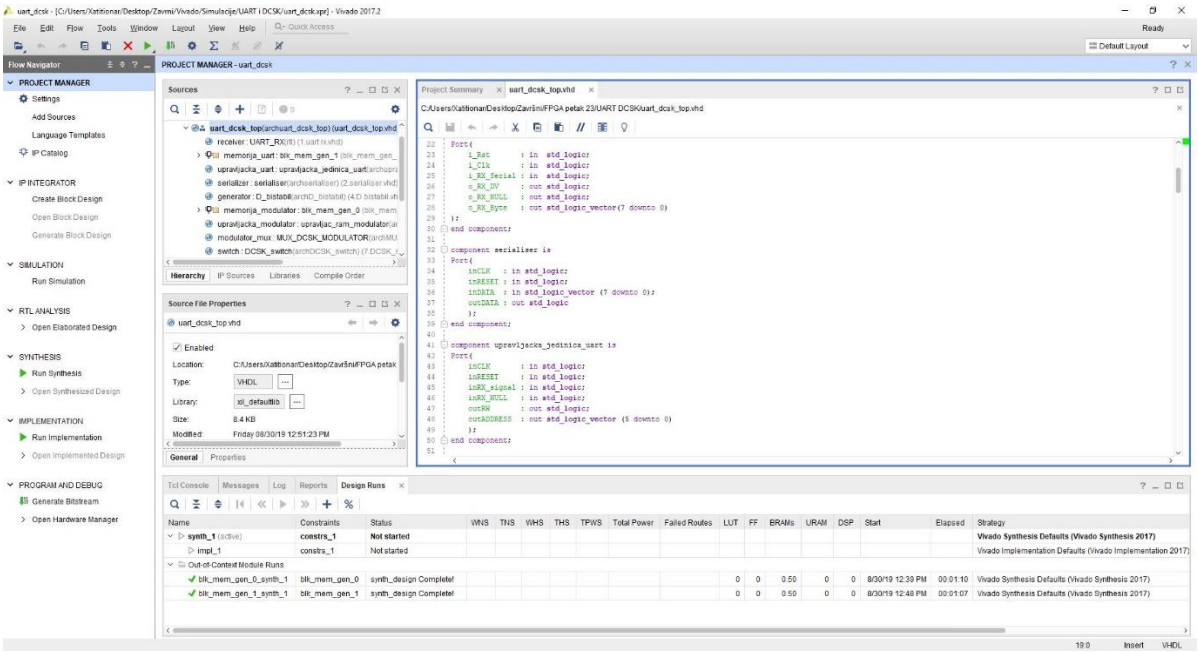


Sl 5.1. Prikaz grafičkog sučelja ModelSim programa.

5.2. Vivado razvojno okruženje

Vivado je razvojno okruženje koje je razvila tvrtka Xilinx u travnju 2012. godine. Služi programerima za analizu, sintezu, implementaciju i vremenske analize HDL (engl. *Hardware Description Language*) dizajnova, ispitivanje RTL (engl. *Register – transfer level*) dijagrama, no kao proizvod tvrtke Xilinx može raditi samo s Xilinx-ovim čipovima. Također Vivado u sebi sadrži i logički simulator ISIM, sintezu visoke razine te ima mogućnost prevođenja koda iz programskog jezika C direktno u programibilnu logiku. Za prikaz veličine i kompleksnosti ovog razvojnog okruženja – za njegov razvoj utrošeno je preko 1000 godina ljudskoga rada i preko 200 milijuna Američkih dolara [7].

U ovome radu koristit će se PLFS (engl. *Post –Layout Functional Simulation*) simulacija koja u obzir uzima broj perioda takta koji je potreban za izvođenje pojedinih operacija opisanih u dizajnu sustava, propagacijska kašnjenja kroz sklopove te kašnjenja signala kroz sabirnice kojima su povezani sklopovi u FPGA pločici. Uzimajući u obzir sve ove parametre, PLFS simulacija predstavlja najpreciznu simulaciju koju možemo izvesti u Vivado razvojnom okruženju.



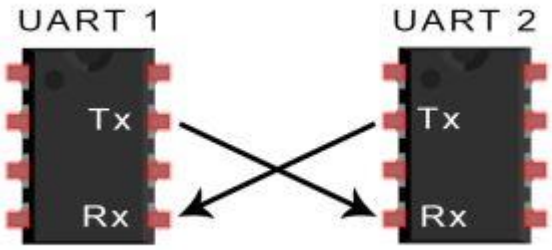
Sl 5.2. Prikaz Vivado razvojnog okruženja.

5.3. UART

UART je modem koji komunicira serijskom vezom odnosno koristi jednu žicu za prijenos podataka, od predajnika do prijemnika. Podaci se mogu slati i primati asinkrono, a svaki podatak koji je poslan između predajnika i prijemnika sadrži start i stop bit. Brzina kojom UART prima i šalje podatke je određena bitovima po sekundi, te da bi komunikacija između predajnika i prijemnika bila moguća, oboje moraju raditi na istoj brzini prijenosa koja pak može varirati od 110 do 128000 bita po sekundi. Informacijski okvir može biti između 5 i 8, odnosno 9 bita ako se ne koristi paritetni bit.

Prilikom rada UART – a, prijenosna žica se drži na visokoj naponskoj razini. Kada se želi prenijeti podatak, napon se spušta na nisku razinu u trajanju od jednog bita, što govori prijemnoj strani da može početi primati podatke. Kada se primio podatak, napon se diže na visoku razinu u trajanju od barem dva bita.

Prednosti korištenja UART – a su: korištenje samo jedne žice za prijenos, nema potrebe za izvorom takta, ima paritetni bit za provjeru pogreške, dobro dokumentirana i široko korištena metoda. Također postoje i mane: veličina podatka može biti samo 9 bita, ne podržava više nadređenih niti podređenih sustava, brzina prijenosa mora biti usklađena s tolerancijom od maksimalno 10%.

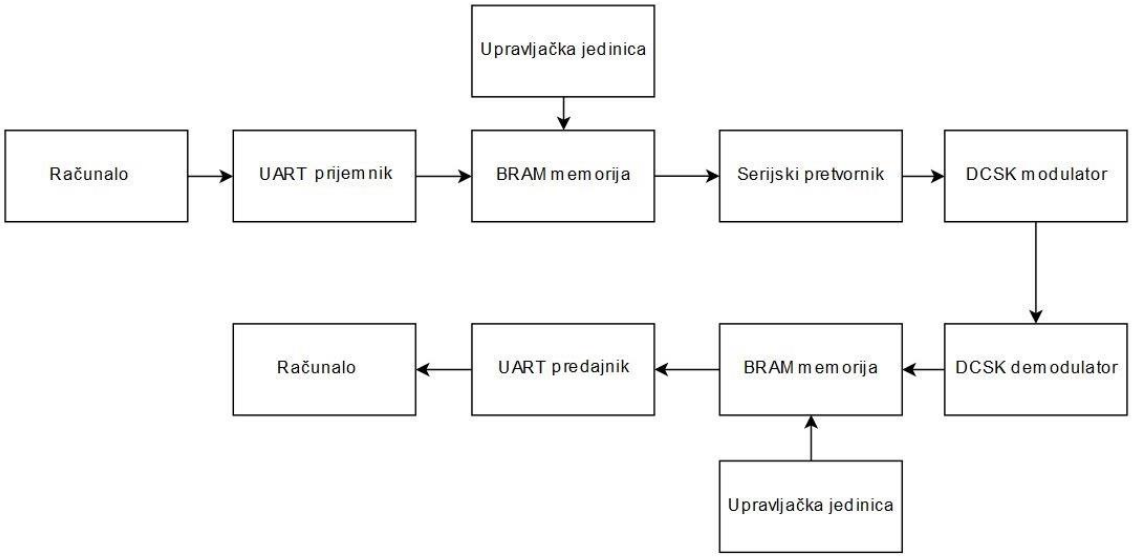


Sl 5.3. Prikaz UART modema.

5.4. Implementacija DCSK modema na Zybo Z-7000

Za implementaciju DCSK modema na Zybo Z – 7000 dodan je UART modem te je RAM memorija zamijenjena BRAM (engl. *Block Random Access Memory*) memorijom. Na slici 5.4.

prikazana je blok shema sustava koji se implementira na Zybo pločicu, a sastoji se od DCSK modulatora i demodulatora, dvije BRAM memorije te njihovih upravljača, serijskog pretvornika koji prima podatke iz memorije i pretvara ih niz bitova, UART prijemnika i predajnika koji služe sustavu za serijsku komunikaciju između računala te samog računala koje služi za slanje podataka na sustav i primanje istih iz sustava. VHDL kod svih dijelova sustava dan je u prilogu P1.



Sl 5.4. Blok shema sustava za implementaciju na Zybo Z – 7000.

Blok „UART predajnik“ služi za pretvorbu serijski primljenih podataka sa računala u 8 – bitne vektore odnosno simbole. Izvorni kod preuzet je sa stranice Nandland te modificiran po potrebi. [8]

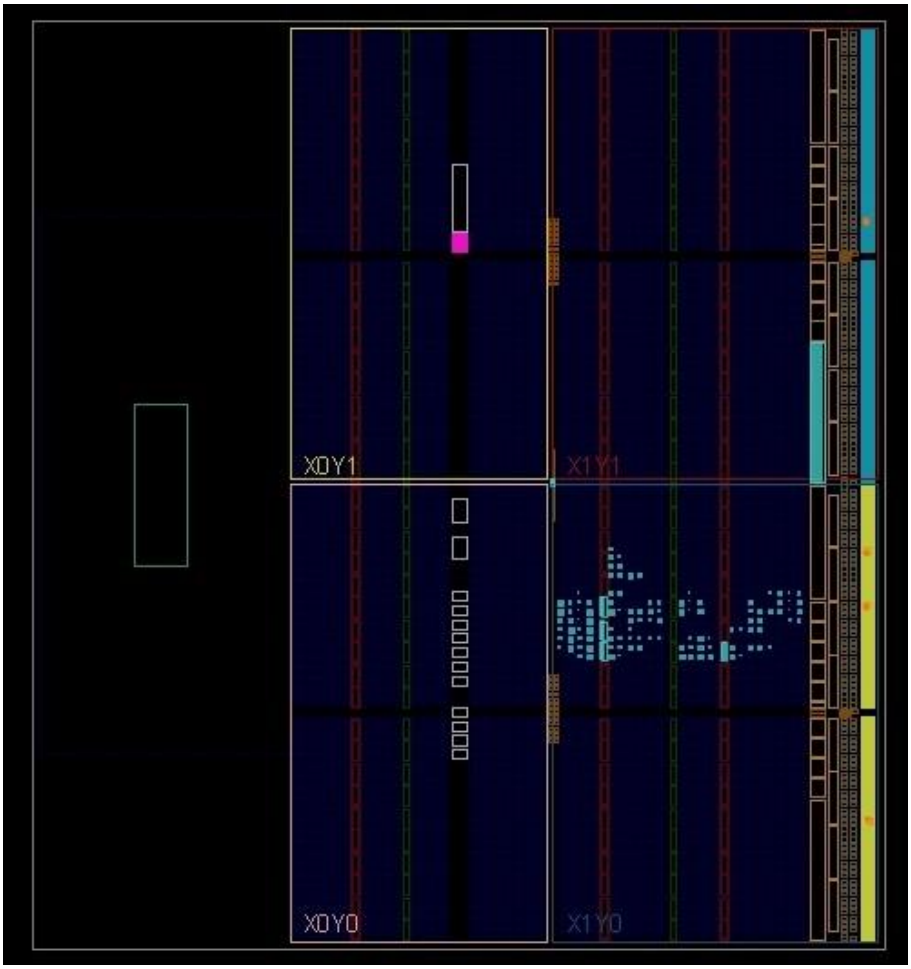
Blokovi „BRAM memorija“ je IP (engl. *Intellectual Property*) sadržan u Vivado razvojnom okruženju i predstavlja blokovsku memoriju kakva se nalazi unutar FPGA pločicama. Kao realna komponenta potrebno joj je dva perioda takta za radnje upisa i ispisa. Upotrebljena je u primanju podataka od UART prijemnika, te DCSK demodulatora.

Blokovi „Upravljačka jedinica“ služe za upravljanje upisom i ispisom iz BRAM memorija sustava te adrese na koju će se pojedini podataka upisati.

Blokovi „DCSK modulator“ i „DCSK demodulator“ opširnije su opisani u poglavlju 4.1. i 4.2.

Blok „UART predajnik“ služi za pretvorbu podataka koji pristižu iz memorije u obliku 8 – bitnih vektora nazad u serijski oblik podataka te slanje istih nazad na računalo.

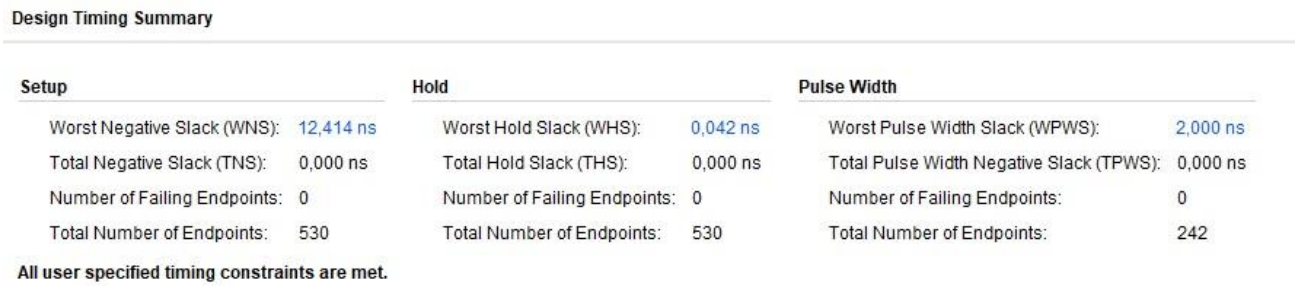
Sustav implementiran na Zybo pločicu zauzeo je 2% LUT (engl. *Lookup Table*), 1% FF (engl. *Flip -Flop*), 3% BRAM, 4% IO (engl. *Input / Output*), 6% BUFG (engl. *Global Clock Buffer*) te 50% MMCM (engl. *Mixed – Mode Clock Manager*). Iz slike 5.4. može se vidjeti kako je Vivado razvojno okruženje postavilo sve potrebne komponente na pločicu.



Sl 5.5. Prikaz zauzeća komponenti na Zybo pločici.

Nakon uspješne implementacije dobivaju se rezultati vremenske analize sustava koji stavljamo na pločicu. Ova vremenska analiza služi Vivadu kako bi odredio može li zadani sustav raditi s danom frekvencijom signala takta. Ukoliko Vivado odredi da sustav ne može raditi s danom frekvencijom

signala takta, problem se rješava smanjivanjem frekvencije signala takta ili ubacivanjem registara u kombinacijske mreže (engl. *pipelining*). Iz dobivene vremenske analize, na slici 5.5. možemo vidjeti da je parametar „Worst Negative Slack“ plave boje te iznosit 12.414 nanosekunde. Iz tog možemo zaključiti da sustav koji sada radi na 50 MHz, čiji je period 1/50 000 000 = 20 nanosekundi, može raditi na periodu od 20 – 12.414= 7.586 nanosekundi, odnosno frekvenciji od 1 / 0.000000007586 = 131.82 MHz.



Design Timing Summary

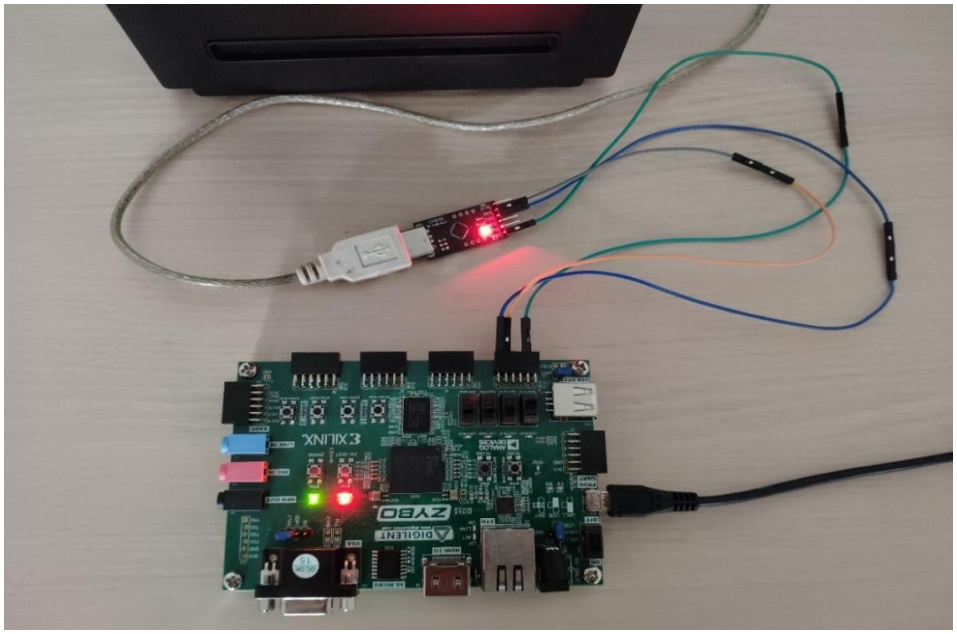
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	12,414 ns	Worst Hold Slack (WHS):	0,042 ns	Worst Pulse Width Slack (WPWS):	2,000 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	530	Total Number of Endpoints:	530	Total Number of Endpoints:	242

All user specified timing constraints are met.

Sl 5.6. Prikaz vremenske analize sustava.

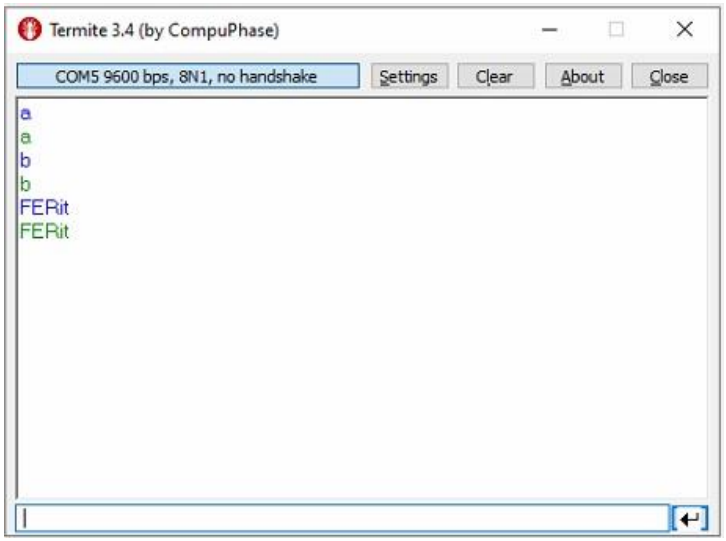
5.5. Testiranje DCSK modema

Testiranjem ovog sustava dobijamo uvid u valjanost rada istog sustava te možemo vidjeti obavlja li sustav svoj zadatak odnosno vraća li iste simbole koje primi. Za testiranje korišteno je stolno računalo s aplikacijom Termite koja predstavlja serijski terminal. Za komunikaciju računala s Zybo pločicom korišten je USB – UART adapter koji služi za serijsku komunikaciju.



Sl 5.7. Prikaz Zybo pločice spojene sa računalom preko USB – UART adaptera.

Nakon programiranja Zybo pločice binarnom datotekom sustava spojili smo USB – UART adapter u računalo i na Zybo pločicu. Sa slike 5.8. možemo vidjeti aplikaciju Termite te poslane i primljene simoble. Plava boja simbola predstavlja poslani simbol, dok zelena boja predstavlja primljeni simbol.



Sl 5.8. Prikaz Termite terminala te prikaz valjanosti rada sustava.

6. ZAKLJUČAK

Zadatak ovog rada bio je opisati DCSK modulator i demodulator, temeljen na teoriji kaosa i kaotičnim signalima, u VHDL jeziku za opisivanje sklopovlja te implementirati ih na FPGA integrirani krug Zybo Z – 7000. Također, za uspješnu implementaciju, cijelom dizajnu dodan je UART prijemnik i odašiljač, kako bi se omogućila komunikacija sklopovlja s računalom.

Pri izradi korišten je jezik za opisivanje sklopovlja VHDL. Za implementaciju korištena je Zybo Z – 7000 razvojna pločica koja pruža pogodnosti u smislu periferije, ARM procesora s dvije jezgre i dovoljno ugrađene memorije za upoznavanje sa svijetom dizajna sklopovlja.

Opisivanje sklopovlja u VHDL jeziku za opisivanje sklopovlja bio je prvi zadatak koji se trebao napraviti. Svaki blok dizajna dizajniran je kao posebna *.vhd datoteka, te su na kraju sve zajedno spojene preko entiteta više hijerarhije. Nakon dizajna, cijeli sustav je simuliran u ModelSim okruženju za računalnu simulaciju, a nakon uspješne simulacije prelazi se na implementaciju sustava na Zybo pločicu. Kako bi implementacija bila uspješna, a sačuvali se registri Zybo pločice, koriste se blokovi BRAM memorije te sklop za manipulaciju odnosno preskaliranje frekvencije signala takta, koji predstavljaju intelektualno vlasništvo softverskog paketa Vivado. Pomoću istog softverskog paketa provedena je sinteza, implementacija i generiranje binarne datoteke za programiranje Zybo pločice. Za ispitivanje valjanosti rada, preko računala je pomoću UART protokola poslan podatak koji je DCSK modem modulirao, demodulirao i poslao nazad isti.

DCSK modulator i demodulator moguće je proširiti u vidu više bitova koji se koriste za generiranje kaotičnog signala, povećanjem veličine okvira kojim se prenose pojedinačni bitovi simbola. Također, za bolje ponašanje sustava u uvjetima sa šumom i raznim prigušenjima u komunikacijskom kanalu, predlaže se dodavanje bloka koji računa konjugirano kompleksnu vrijednost referentnog okvira na strani demodulatora i zbraja ju s informacijskim okvirom. Ovako poboljšan sustav mogao bi se koristiti za prenošenje podataka i preko energetskih vodova i u raznim situacijama gdje je velika mogućnost izobličenja poslanog podatka zbog šuma u komunikacijskom kanalu.

LITERATURA

[1] F. C. M. Lau, C. K. Tse, Chaos-Based Digital Communication Systems, Springer – Verlag Berlin Heidelberg, New York, 2003.

[2] https://en.wikipedia.org/wiki/Chaos_theory [12.9.2019.]

[3] Hasselblatt, Boris, Anatole Kato, A First Course in Dynamics: With a Panorama of Recent Developments, Cambridge University Press, 2003.

[4] https://www.allaboutcircuits.com/technical-articles/what-is-an-fpga-introduction-to-programmable-logic-fpga-vs-microcontroller/ [12.9.2019.]

[5] https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo-z7-1.png [12.9.2019.]

[6] ZYBO Reference Manual, Diligent, 14.2.2014.

[7] https://www.xilinx.com/products/design-tools/vivado.html [12.9.2019.]

[8] https://www.nandland.com/vhdl/modules/module-uart-serial-port-rs232.html [12.9.2019.]

[9] D. L. Perry, VHDL Programming by Example, Fourth Edition, McGraw-Hill, London, 2002.

[10] Lenny Smith, Leonard Smith, Chaos: A Very Short Introduction, Oxford University Press, 2007.

[11] Clive Maxfield, Tutorial: Linear Feedback Shift Registers (LFSRs) - Part 1, December 20, 2006.

[12] http://www.circuitbasics.com/basics-uart-communication/ [12.9.2019]

[13] https://www.mentor.com/company/higher_ed/program_details [12.9.2019.]

SAŽETAK

U ovom radu opisan je postupak izrade DCSK modulatora i demodulatora temeljenih na teoriji kaosa, odnosno kaotičnim signalima, opisanih koristeći VHDL jezik za opisivanje sklopovlja te implementirnih na FPGA integrirani krug Zybo Z – 7000. Prvo je opisana teorija kaosa te kako se od te teorije došlo do kaotičnih signala i DCSK modulacije. Zatim su opisani VHDL jezik za opisivanje sklopovlja i zbog čega je nastao te kako se pomoću njega opisuju sklopovi, te Zybo razvojna pločica i sve njene komponente i mogućnosti. Zatim je opisan proces izrade DCSK modulatora i demodulatora te što svaki blok u dizajnu predstavlja i kako radi. Nakon toga, opisani su Vivado i ModelSim alati koji su korišteni za simulaciju i implementaciju dizajna. Završno, prikazano je zauzeće dizajna na samoj Zybo pločici te testiranje ispravnosti rada samog dizajna pomoću računala, preko USB - UART adaptera odnosno UART protokola.

Ključne riječi: DCSK, komunikacija, kaos, kaotični signali, modulacija, modem, VHDL, FPGA

ABSTRACT

Implementation of DCSK modulator and demodulator on FPGA

This paper describes the procedure for designing DCSK modulator and demodulator, based on Chaos theory and chaotic signals. VHDL hardware description language is used to describe the design, while FPGA integrated circuit, Zybo Z – 7000, is used for implementation. Firstly, the Chaos theory was described and how DCSK modulation was derived from the knowledge about chaos. Secondly, the VHDL hardware description language was introduced and an explaination on how to use it to describe electrical circuits and why it was made in the first place is given. Thirdly, Zybo development board and all it's components and functionalities are described. Then the proces of designing the DCSK modulator and demodulator is described, along with what each block of the design represents and how it works. Afterwards, Vivado, ModelSim and their use for simulation and implementation of the design is given. Finally, the design's usage of the Zybo board and the testing of the design itself using a computer and an USB – UART adapter is shown

Key words: DCSK, communication, chaos, chaotic signals, signal modulation, modem, VHDL, FPGA

ŽIVOTOPIS

Domagoj Vinogradac rođen je 2. svibnja 1997. godine u Osijeku. Školovanje započinje u osnovnoj školi Matije Petra Katančića u Valpovu. Nakon osnovne škole upisuje Srednju Školu u Valpovu, smjer Elektrotehničar, koju završava u školskoj godini 2015./2016. Odmah nakon završetka srednje škole, 2016. godine upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, na preddiplomski studij računarstva. Trenutno je redoviti student 3. godine.

PRILOZI

P1 Izvorni kod projekta u VHDL – u, nalazi se na CD – u