Univerzitet u Kragujevcu
Fakultet inženjerskih nauka

Seminarski rad iz predmeta
Osnovi računarske tehnike 2

Tema:
Sistem zaštite – lozinka

Student:
Sofija Anđelković 565/2015

Predmetni profesor:
Aleksandar Peulić

Kragujevac 2017.

SADRŽAJ

1. Uvod

U ovom projektu opisana je realizacija sistema zaštite korišćenjem razvojnog sistema FPGA-Spartan-3E-S500, kao i arhitektura samog razvojnog sistema. Spartan-3E porodica je samo jedna od mnogobrojnih porodica FPGA (engl. *Field-Programmable Gate Arrays).* Ova porodica je specijalno dizajnirana da zadovolji potrebe velikog obima, prilagođen potrošačima koji su osetljivi na cenu. Spartan-3E porodica broji pet članova, od kojih jedan od članova sadrži čak 1,6 miliona sistemskih kola.

Cela porodica je izgrađena na uspehu Spartan-3 porodice, tako što je povećan broj ulazno/izlaznih uređaja, što značajno smanjuje cenu po logičkoj ćeliji. Nove funkcije su poboljšale performanse sistema i smanjile troškove konfiguracije. Spartan-3E je prvi FPGA sa tehnologijom od 90nm. Kada je pušten u prodaju bio je funkcionalniji od svih prethodnika i postavio nove standarde u industriji programabilne logike.

Zbog svoje izuzetno niske cene, ova generacija je pogodna za široki spektar primene u elektronskim uređajima od kućnih mreža preko projektovanja slike do opreme za digitalnu televiziju.

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits(1)	Block RAM bits(1)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XC3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	232	92
XC3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XC3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

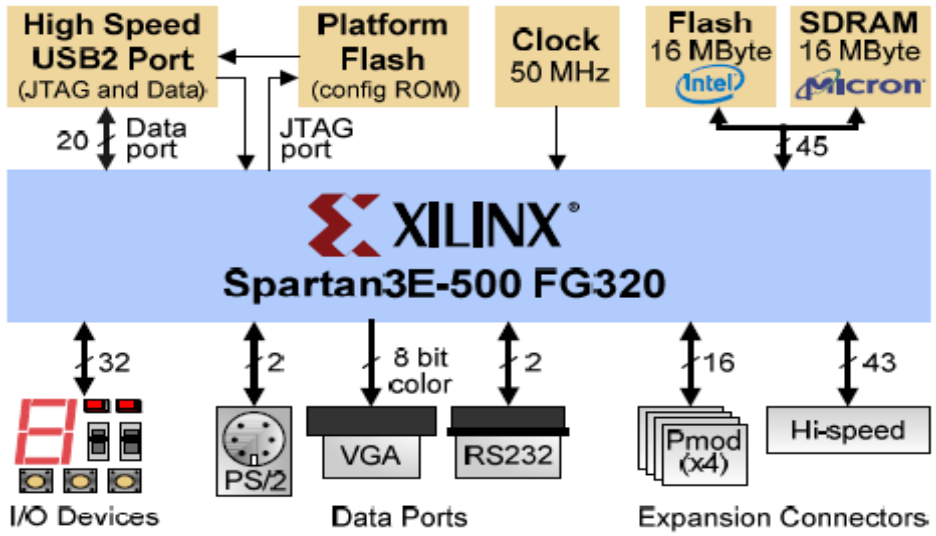
Notes:
1. By convention, one Kb is equivalent to 1,024 bits.

2. Arhitektura

FPGA je integrisano električno kolo, odnosno složeno kolo sastavljeno od mnoštva elemenata, uglavnom tranzistora smeštenih na jednoj podlozi. Oni su napravljeni tako da krajnji korisnik treba da ih konfiguriše, tj programira. Programira ih tako što pravi bit fajl. Bit fajl može da se napravi pomoću slobodnog ISE/WebPack CAD softvera od Xilinx-a korišćenjem VHDL-a, Veriloga ili šematski. Taj bit fajl se prenosi u memorijski deo FPGA da bi se definisale logičke funkcije i međusobni odnosi u kolu. Uloga FPGA je višestruka, ali osnovna im je implementacija logičkih funkcija.

FPGA je smešten na Nexys 2 platformi. Nexys 2 je moćna digitalna platforma bazirana na Xilinx Spartan-3E-FPGA. Što se arhitekture tiče, nju čine:

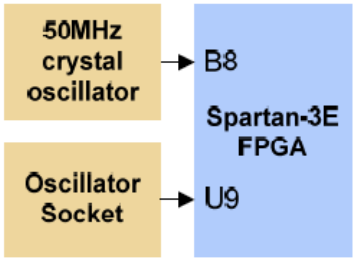
- FPGA konfiguracija koja je zasnovana na USB2 portu i brzom protoku podataka (uz korišćenje besplatnog Adept Sutie softvera)
- 500K-gate Xilinx Spartan 3E FPGA
- USB napajanje (može se koristiti baterija)
- 16MB Micron PSDRAM-a i 16MB-a Intel StrataFlash ROM-a
- Xilinx Platform Flash za razne FPGA konfiguaracije
- Efikasno napajanje preko prekidača (korisno kada je u pitanju napajanje baterijom)
- 50MHz oscillator uz dodatni ulaz za drugi oscillator
- 60 FPGA ulazno/izlaznih jedinica povezanih na proširujuće konektore (jedan brzi Hirose FX2 konektor i četiri 6-pinski kvadratna konektora)
- 8 LED, 4 sedmo-segmentna displeja, 4 dugmeta i 8 prekidačca
- Prodaje se u plastičnoj kutiji zajedno sa USB kablom



Trenutno glavni proizvođači FPGA komponenti čine kompanije Altera i Xilinx. Zajedno kontrolišu preko 80% tržišta, a sam Xilinx preko 50% čitavog tržišta

2.1. Clocks

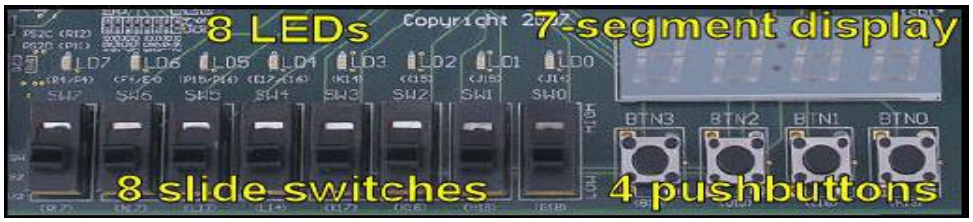
Nexys2 ploča sadrži oscilator od 50MHz, kao i priključak za drugi oscilator. Signali kloka koje šalje oscilator se konektuju na input pinove globalnog kloka FPGA čipa kako bi blokovi klokovog sintesajzera bili dostupni u FPGA čipu. Sintisajzeri kloka (ili DLL-ovi) pružaju mogućnost uvećanja ulazne frekvencije za duplo ili četiri puta više, deljenje ulazne frekvencije nekim celim brojem, kao i precizno definisanje faze i kašnjenja između različitih signala kloka.



2.2. Ulazno/Izlazne komponete

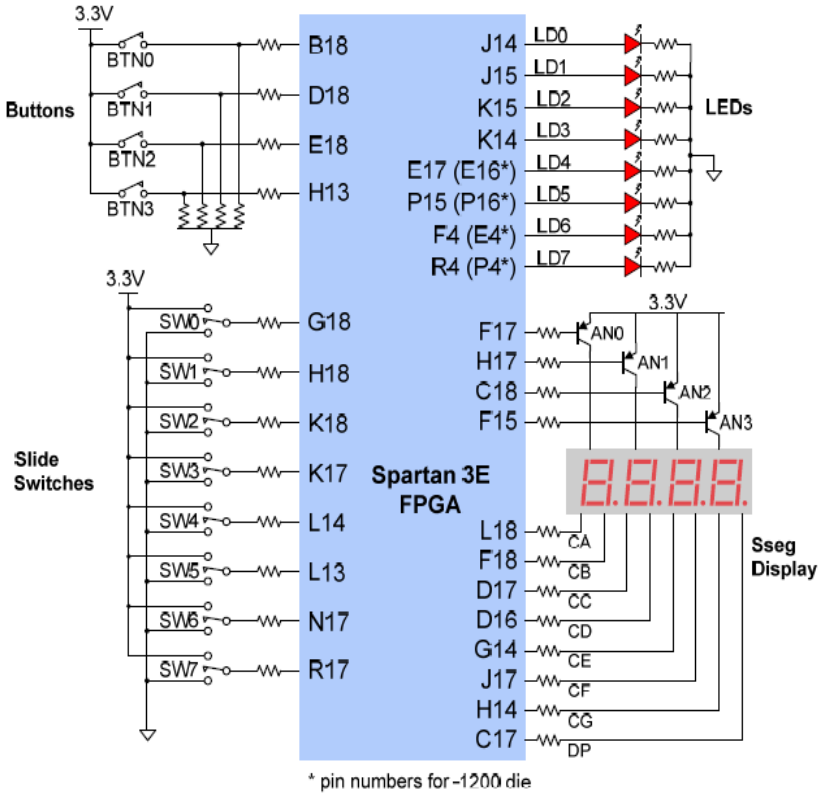
2.2.1. Ulazne komponente

Ulazne komponente Nexys2 platforme su četiri tastera i osam prekidača. Stanje je inače nekativno (logička 0). Do pobuđivanja, tj do aktivnog stanja (logička 1) dolazi kad pritisnemo taster i traje samo dok je taster pritisnut. Što se tiče prekidača, od njihovog položaja zavisi da li generišu logičku 1 ili logičku 0.

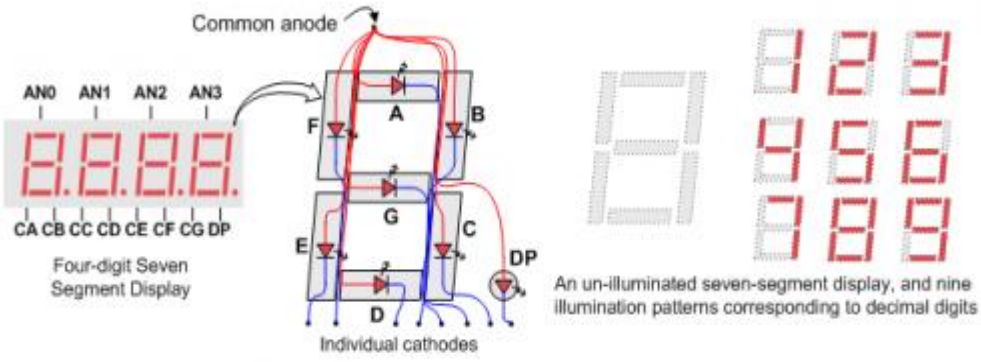


2.2.2. Izlazne komponente

Izlazne komponente čine četiri sedmo-segmentna displeja i osam LED-a (engl. ***L**ight **E**mitting **D**iode*). Do paljenja diode dolazi kada se na LED anodu pošalje logička 1, tada se proizvodi struja od 3mA, a da ne bi došlo do opterećenja samog FPGA, ispred svake anode se nalazi otpornik od 390Ω.



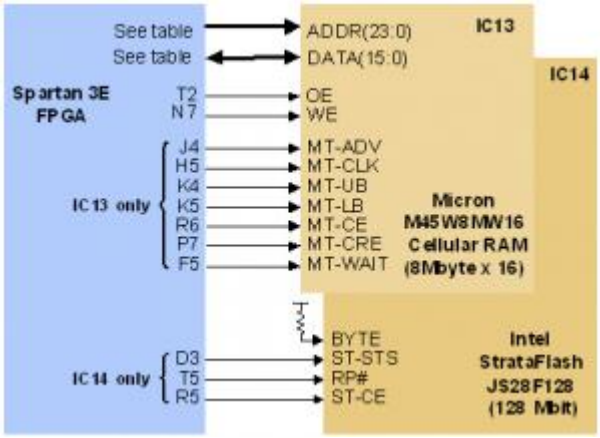
Nexzs2 ploča sadrži četiri spojena 7-segmentna displeja. Anode sedmosegmentnih LED-ova koje formiraju svaku cifru su povezane u "zajednicko anodsko" kolo, ali katode LED dioda su odvojene. Signali zajedničke anode su dostupni preko signala input dozvole od četiri cifre za četvorocifreni displej (AN0, AN1, AN2, AN3). Nekativno stanje anode je logička 1, tako da slanjem logičke 0 biramo koji će displej da svetli. Što se tiče katoda, oni prave isti obrazac za sva četiri displeja, a od anodne dozvole zavisi koji će da svetli. Slanjem logičke 0 na određeni segment (a, b, c, d, e, f, g), gasimo taj segment i na taj način pravimo obrazac koji želimo.



2.3. Memorija

Nexys 2 ploča ima eksterne RAM i ROM uređaje. Ekstermi RAM je 128megabitni Micron M45W8MW16 Cellular RAM pseudo-statičan DRAM uređaj organizovan sa 8megabajta x 16bita. Može da radi kao tipičan asinhroni SRAM sa read i write ciklusima za vreme od 70 nanosekundi, ili kao sinhrona memorija sa 80MHz. Kada radi kao asinhroni SRAM, Cellular RAM automatski obnavlja interne nizove DRAMa, što je omogućeno od strane jednostavno dizajniranog kontrolera u FPGA. Kada radi kao sinhroni, prenosi idu i do 80MHz.

Eksterni ROM je 128megabitni Intel TE28F128J3D75-110 StrataFlash uređaj organizovan kao 8megabajta x 16bita. Sadrži 128 blokova koji pojedinačno mogu biti izbrisani i podržava read cikluse za vreme od 110 nanosekundi. Ima interni 32bitni write bafer koji može biti ispisan za 70ns ciklusa i 32bajtni bafer može biti prebačen Fleš niz za 218 us. Oba uređaja dele zajedničku 16-bitnu magistralu podataka i 25-bitnu adresnu magistralu.



2.4. Xilinx Spartan-3E (XC3S500E)

Jedan od pet članova Spartan-3E je XC3S500E i njega odlikuju sledeće:

- 500k sistemskih logičkih kola
- 10,476 ekvivalentnih logičkih ćelija
- 73K distribuiranih RAM bitova
- 360k blok RAM bitova
- 20 množača
- 4 DCM-ova
- 158 korisničkih ulazno/izlaznih portova
- 65 diferencijalnih ulazno/izlaznih parova

3. Projektni zadatak

Projektni zadatak čini sistem zaštite, što je 7-bitna lozinka. Može se koristi za zaštitu bilo kakve vrste, ali direktno primenu može naći kod sefova i uopšte ulaznih vrata, kako vrata na ulazu u kuću/stan ili bilo koju prosotoriju gde je dozvoljen pristup samo određenim ljudima. Lozinka se sastoji od 7 bitova i samim tim ima 128 različitih kombinacija (2^7). Za osnovne potrebe to je sasvim visoka zaštita. Ukoliko se ukuca tačna binarna kombinacija i pritisne 'ok' prekidač dolazi do ispisivanja 'g00d' na displeju i blinkanja svih 8 dioda. One blinkaju naizmenično da bi se stvorio bolji efekat. Ukoliko se ne ukuca tačna lozinka, na displeju se ispisuje 'FAIL'. Lozinka je ukucana u sam kod, ako treba da se promeni, dovoljno je pristupiti izvornom kodu, promeniti lozinku i ponovo generisati.

4. Realizacija projektnog zadatka

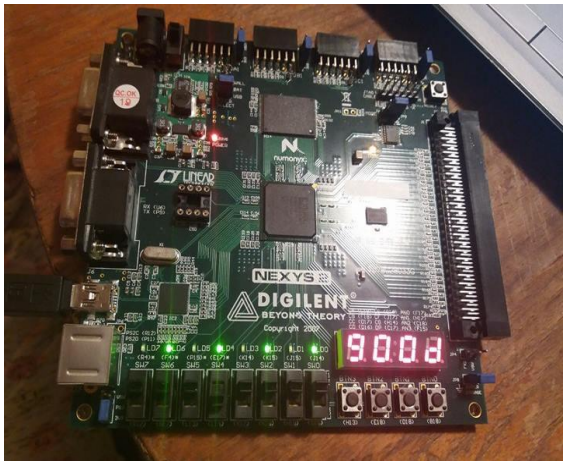
Za pisanje koda korišćen je Verilog. Ceo kod je napisan u jednom modulu. Na početku su definisani svi ulazi i izlazi. Ulaze čine 7-bitna lozinka, koja je uneta u sam kod i spojena u implementacionom fajlu sa sedam prekidača, počev od G18, zaključno sa N17, kao i dozvola 'en' koja je spojena sa prekidačem R17 i 'reset' koji je spojen sa tasterom H13. Izlaze čine svi LED-ovi, kao i 7-segmentni displej.

Prvo je definisano blinkanje dioda. Napravljen je 32-bitni brojač, koji se inkrementira na svaku pozitivnu ivicu kloka. Kada god se na 22.bitu pojavi logička jedinica, diode LD0, LD2, LD4, LD6 svetle, dok je na diodama LD1, LD3, LD5 i LD7 na 22.bitu brojača postavljen inverotor, tako da kad dođe logička 1, on je invertuje, tako da neparne diode se pale kada na 22.bit brojača dođe logička 0. Sve to vizuelno stvara naizmenično blinkanje dioda. One su AND naredbom povezane sa lozinkom, tako da blinkaju samo u slučaju ukucane tačne binarne kombinacije. Diode su takođe AND naredbom povezane i sa prekidačem dozvole, jer dolazi do blinkanja tek kada se potvrdi lozinka, a uz to je i tačna.

Za realizaciju projekta je korišćen i 7-segmentni displej za ispisivanje određenih reči. Za ovu realizaciju bilo je potrebno da ubacimo još jedan brojač, u ovom slučaju 17-bitni, koji se inkrementira na svaku pozitivnu ivicu kloka. U slučaju da nemamo brojač, ne bismo mogli da u isto vreme prikažemo različite obrasce na svakom od četiri različita displeja. Tehnički je nemoguće prikazati različite obrasce istovremeno, ali ubacivanjem brojača i brzom menjaju stvara se osećaj da svi svetle u isto vreme. Ceo proces moramo da podelimo u četiri dela, u prvom delu svetli samo prvi displej određenim obrascem, tako što na zajedničko anodno kolo šaljemo četvorobitnu dozvolu, takvu da je logička 0 (u ovom slučaju) na prvom mestu, a sve ostalo su logičke 1. Na taj način smo realizovali da svetli samo prvi displej, a ostala tri ne. Analogno tome se realizuju i ostala tri slučaja. Kada se ta četiri slučaja pomoću brojača brzo menjaju u krug, stvara se privid da istovremeno svelte sva četiri displeja.

Jedan deo koda čini definisanje svih mogućih obrazaca uz pomoć case naredbe. Isto tako se uz pomoć case naredbe i pozivaju obrasci koji pod određenim uslovima treba da se pojave na određenom displeju.

U samoj osnovi celog projekta je jedna If naredba od koje zavisi sve ostalo. Ako je ukucana binarna kombinacija jednaka binarnoj kombinaciji unetoj u sam kod dolazi do ispisivanja reči 'g00d' na displeju i blinkanju dioda naizmenično (slika1). Ako nije onda na displeju se ispisuje 'FAIL' (slika2). Uz sve to je dodata dovzvola, u smislu naredbe 'ok' kada se unese šifra, kao i 'reset' taster koji vraća brojače na početak.



Slika1

Slika2

Kada se sve ovo ukomponuje, implementacionim fajlom poveže sa komponentama na ploči, generiše se. Pomoću besplatnog Adept Sutis softvera se isprogramira tako što se bit fajl smesti u memorijsku ćeliju FPGA. Na taj način su se definisale međusobne veze u kolu, kao i logične funkcije i projekat je spreman za korišćenje.

5. Zaključak

S obzirom da živimo u eri informacionih tehnologija, na svakom koraku se susrećemo sa nekim vrstama zaštite kako podataka, tako i stvari sa visokim stepenom vrednosti. Jedna od najzastupljenijih načina zaštite jeste lozinka. Postoje razne vrste lozinka, u vidu brojeva, komibnacije karaktera, šablona itd… U ovom projektu je reč o 7-bitnoj lozinci, koja se može koristiti za zaštitu sefova ili bilo kojih prostorija u kojima je odobren pristup isključivo određenim ljudima, tj. ljudima koji znaju lozinku. Može biti postavljena na vrata, kako sefa, tako i obična. Samo u slučaju ukucane tačne 7-bitne kombinacije, dolazi do otključavanje brave. Sad, za vrlo malu cenu, svima dostpunu, može da postavi 7-bitnu šifru na neku vrstu sefa. S obzirom da ima 128 različitih kombinacija, u slučaju obične provale, sa kojom se i najčešće susrećemo, provalnik neće imati vremena da pogađa lozinku, tako da će najvrednije stvari ostati van njegovog domašaja. Isto tako, ovakva vrsta lozinke može da bude i deo alarmnog sistema, pa da u slučaju nekoliko pogrešnih ukucanih lozinki, da se aktivira alarm. Svakako nešto što treba svako da ima, radi lične bezbednosti, a i zaštite stvari i podataka sa velikom vrednošću

.

5.1. Prilog kodovi

```
module lozinka(
input reset,
input clk,
input [6:0] x,
input en,
output a, b, c, d, e, f, g,
output [3:0] an,
output led1, led2, led3, led4, led5, led6, led7, led8);
reg [32:0] brojac;

always @(posedge clk)
 begin
 if (reset)
 brojac <= 0;
 else
 brojac <= brojac + 1;
 end
 assign led1 = brojac[22] & en & x == 7'b1101111;
 assign led2 = ~brojac[22] & en & x == 7'b1101111;
 assign led3 = brojac[22] & en & x == 7'b1101111;
 assign led4 = ~brojac[22] & en & x == 7'b1101111;
 assign led5 = brojac[22] & en & x == 7'b1101111;
 assign led6 = ~brojac[22] & en & x == 7'b1101111;
 assign led7 = brojac[22] & en & x == 7'b1101111;
 assign led8 = ~brojac[22] & en & x == 7'b1101111;
reg [17:0]count;

always @ (posedge clk or posedge reset)
 begin
  if (reset)
   count <= 0;
  else
   count <= count + 1;
 end
reg [6:0]seg;
reg [3:0]an_temp;
always @(*)
begin
if (en == 0)
begin
 case(count[17:16])

   2'b00 :
    begin
     seg = 8;
     an_temp = 4'b1110;
    end

   2'b01:
    begin
     seg = 8;
     an_temp = 4'b1101;
    end

   2'b10:
    begin
     seg = 8;
     an_temp = 4'b1011;
    end

   2'b11:
    begin
     seg = 8;
     an_temp = 4'b0111;
    end
  endcase
  end
```

```
else if (x == 7'b1101111 & en == 1)
 begin
 case(count[17:16])

   2'b00 :
    begin
     seg = 3 ;              // d
     an_temp = 4'b1110;
    end

   2'b01:
    begin
     seg = 2 ;              // o
     an_temp = 4'b1101;
    end

   2'b10:
    begin
     seg = 2;               // o
     an_temp = 4'b1011;
    end

   2'b11:
    begin
     seg = 1;               // g
     an_temp = 4'b0111;
    end
  endcase
  end

else
   case(count[17:16])
   2'b00 :
    begin
     seg = 7;
     an_temp = 4'b1110;
    end

   2'b01:
    begin
     seg = 6;
     an_temp = 4'b1101;
    end

   2'b10:
    begin
     seg = 5;
     an_temp = 4'b1011;
    end

   2'b11:
    begin
     seg = 4;
     an_temp = 4'b0111;
    end
  endcase
 end
assign an = an_temp;


reg [6:0] seg7;
always @ (*)
begin
case (seg)
1: seg7 = 7'b0000100; //g
2: seg7 = 7'b0000001; //O
3: seg7 = 7'b1000010; //d
4: seg7 = 7'b0111000; //F
5: seg7 = 7'b0001000; //A
6: seg7 = 7'b1111001; //I
7: seg7 = 7'b1110001; //L
8: seg7 = 7'b1111110; //O
endcase
end

assign {a, b, c, d, e, f, g} = seg7;

endmodule
```

Implementacioni kod

```
NET "a" LOC = "L18" ;
NET "b" LOC = "F18" ;
NET "c" LOC = "D17" ;
NET "d" LOC = "D16" ;
NET "e" LOC = "G14" ;
NET "f" LOC = "J17" ;
NET "g" LOC = "H14" ;

NET "clk" LOC = "B8";

NET "an<3>" LOC = "F15";
NET "an<2>" LOC = "C18";
NET "an<1>" LOC = "H17";
NET "an<0>" LOC = "F17";

NET "reset" LOC = "H13";

NET "x[0]" LOC = "G18";
NET "x[1]" LOC = "H18";
NET "x[2]" LOC = "K18";
NET "x[3]" LOC = "K17";
NET "x[4]" LOC = "L14";
NET "x[5]" LOC = "L13";
NET "x[6]" LOC = "N17";

NET "en" LOC = "R17"; //dozvola

NET "led1" LOC = "J14";
NET "led2" LOC = "J15";
NET "led3" LOC = "K15";
NET "led4" LOC = "K14";
NET "led5" LOC = "E17";
NET "led6" LOC = "P15";
NET "led7" LOC = "F4";
NET "led8" LOC = "R4";
```

6. Literatura

1. Digilent Nexys2 Board Reference Manual

2.https://www.xilinx.com/support/documentation/data_sheets/ds312.pdf

3.https://www.wikipedia.org/