Dodatno LED sklopovlje za E2LP razvojni sustav

Birka, Toni

Undergraduate thesis / Završni rad

2017

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:200:729294

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij računarstvo

DODATNO LED SKLOPOVLJE ZA E2LP RAZVOJNI SUSTAV

Završni rad

Toni Birka

Osijek, 2017.

SADRŽAJ

1. UVOD 2
1.1. Zadatak završnog rada 2
2. E2LP RAZVOJNO SUČELJE 3
2.1. FPGA 4
2.1.1. Konfigurabilni logički blokovi 5
2.1.2. Konfigurabilni I/O blokovi 5
2.1.3. Programibilne interkonekcije 5
2.1.4. Konfiguracija FPGA sklopa 6
2.2. VHDL 7
3. RAZVOJ DODATNOG SKLOPOVLJA 8
3.1. Sklopke 8
3.2. 7-segmentni pokaznici 9
3.3. LED matrica 11
3.4. RGB diode 12
3.5. Tipkalo 13
3.6. Shema sklopovlja 13
3.7. Nožice FPGA sklopa za UCF datoteku 14
4. IMPLEMENTACIJA I TESTIRANJE NA EKSPERIMENTALNOJ PLOČICI 15
4.1. VHDL kôd 15
4.2. Spajanje sklopovlja na eksperimentalnoj pločici 21
5. DIZAJNIRANJE I TESTIRANJE TISKANE PLOČICE 22
6. ZAKLJUČAK 24
LITERATURA 25
SAŽETAK 26
ABSTRACT 27
ŽIVOTOPIS 28
PRILOZI 29

1. UVOD

Na kolegiju Digitalna elektronika koristi se E2LP razvojni sustav (eng. *Embedded Engineering Learning Platform*) koji omogućuje studentima programiranje različitih digitalnih sklopova. Međutim, za interakciju s korisnikom na E2LP platformi je dostupno samo nekoliko elemenata kao što su sklopke, tipkala, LCD pokaznik i LE diode.

U svrhu povećanja broja programibilnih elemenata osmišljeno je dodatno LED sklopovlje za E2LP razvojni sustav kako bi se programiranje na kolegiju Digitalna elektronika učinilo zanimljivijim.

Dodatno LED sklopovlje za E2LP razvojni sustav sadrži 16 sklopki, četiri 7-segmentna pokaznika, 8x8 LED matricu, tri RGB diode i jedno tipkalo koje je predviđeno koristiti kao *reset*. Isto kao i originalna pločica, dodatno LED sklopovlje se spaja 50-pinskim konektorom s osnovnom pločom E2LP razvojnog sustava. Dimenzije tiskane pločice dodatnog LED sklopovlja su 10 $cm \times 10\ cm$.

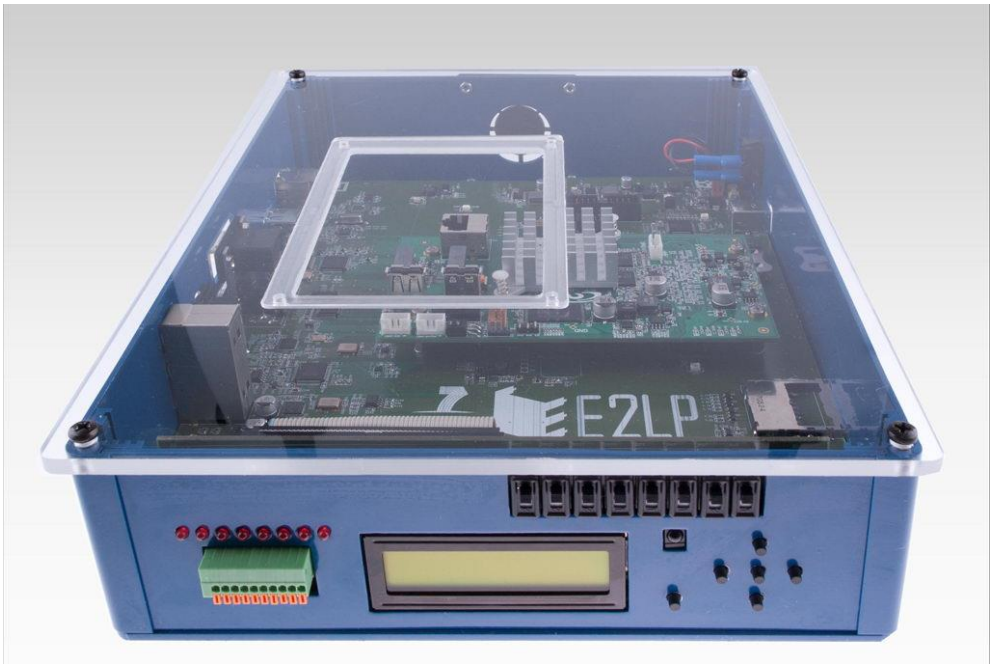
Na početku se, u drugom poglavlju, detaljnije opisuje E2LP razvojni sustav, najvažniji dio tog razvojnog sustava FPGA sklop te VHDL jezik odnosno jezik, koji služi za opisivanje digitalnih sklopova. Nakon toga je u trećem poglavlju podrobnije opisana svaka komponenta dodatnog LED sklopovlja i shema sklopovlja. U četvrtom poglavlju se opisuju glavni dijelovi koda koji opisuje rad pojedinih komponenti dodatnog LED sklopovlja i prikazan je izgled sklopovlja na eksperimentalnoj pločici. Na kraju je prikazan izgled tiskane pločice i završno testiranje.

1.1. Zadatak završnog rada

U ovom radu potrebno je istražiti mogućnosti dodavanja dodatnog LED sklopovlja na E2LP razvojni sustav.

2. E2LP RAZVOJNO SUČELJE

Platforma pod nazivom E2LP služi za edukaciju inženjera o dizajnu ugradbenih sustava kroz razne eksperimente u stvarnom vremenu koji potiču studente da razumiju i grade svoje znanje na temelju tih eksperimenata. Tvrtka RT-RK je dizajnirala i sastavila E2LP osnovnu ploču (eng. *Base Bord)*, koja je prikazana na slici 2.1. [1]



Sl. 2.1: E2LP razvojno sučelje.[1]

E2LP ploča se temelji na *Xilinx Spartan* 6 FPGA sklopu koji omogućuje veliku fleksibilnost i dizajn jednostavnih i kompleksnih sustava koristeći *Xilinx Web Pack* program. Za razliku od mnogih drugih razvojnih sučelja, E2LP ima ugrađeni programator u samoj ploči tako da se komunikacija s računalom ostvaruje direktno preko USB sučelja. Osim FPGA na ploči se nalaze razni priključci kao što su HDMI, RS232, *Ethernet*, Audio, USB, itd. Korisničko sučelje sadrži 16×2 LCD pokaznik, LE diode, sklopke i tipkala.

Kako bi se proširila funkcionalnost razvojnog sučelja moguće je koristiti ploče *NXP ARM* i *Marvell Armada* 1500. *NXP ARM* ploča sadrži dodatne periferije kao što su termometar, akcelerometar, itd. *Marvell Armada 1500* ima bolji procesor pa se koristi pri naprednoj obradi *real-time* digitalnih signala.

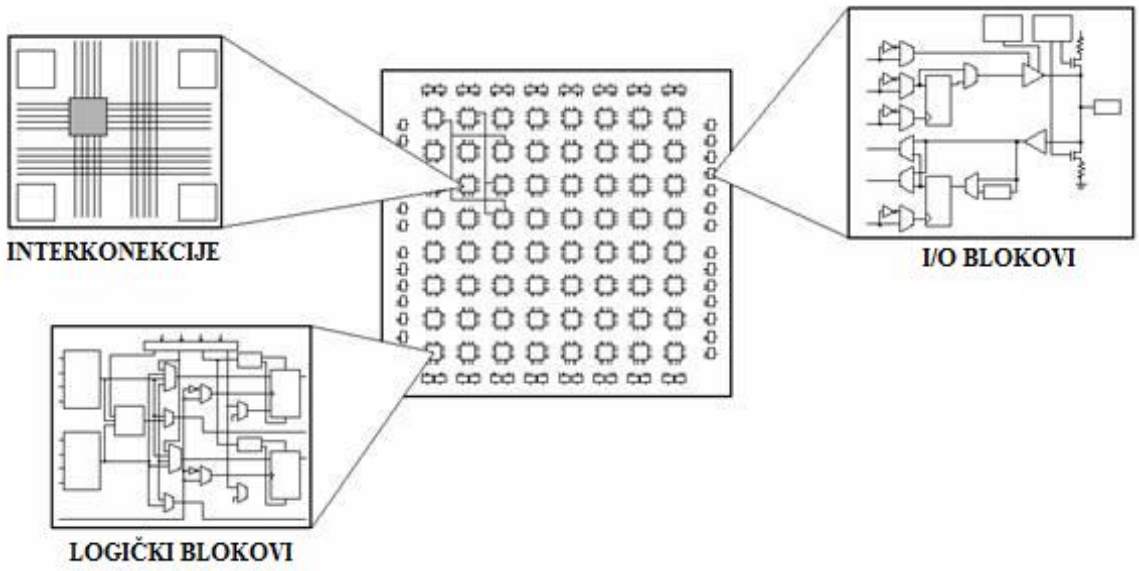
E2LP osnovna ploča obavlja sljedeće funkcije:

- napajanje za E2LP platformu,
- kontrolira programiranje FPGA sklopa i procesora na dodatnom sklopovlju,
- pruža osnovno korisničko sučelje,

- pruža pohranu, multimediju i komunikacijska sučelja za platformu,
- pruža platformu za dizajn digitalnih sustava,
- pruža testne točke za uklanjanje pogrešaka.

2.1. FPGA

Programibilna polja logičkih vrata (eng. *Field-Programmable Gate Array*) su programibilni integrirani sklopovi vrlo visokog stupnja integracije u kojima se veliki broj jednostavnih programibilnih logičkih elemenata može povezati u kompleksni sklop. Ross Freeman, suosnivač tvrtke Xilinx, predstavio je prvi FPGA 1985. godine. Za razliku od tradicionalnih procesora, FPGA je dizajniran da bi ga kupac mogao konfigurirati. [2]



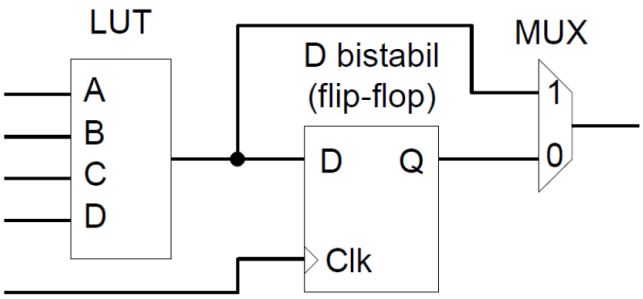
Sl. 2.2: Dijelovi FPGA sklopa.

Slika 2.2 prikazuje glavne dijelove FPGA sklopa, a to su:

- konfigurabilni logički blokovi (eng. *Configurable Logic Block*),
- konfigurabilni I/O blokovi,
- programibilne interkonekcije,
- sklopovi za generiranje signala takta,
- dodatni logički sklopovi kao što su memorija, dekoderi, itd. [3]

2.1.1. Konfigurabilni logički blokovi

Temeljne komponente konfigurabilnih logičkih blokova su pregledna tablica (eng. *Look-Up Table* – LUT) i D bistabil. Pregledna tablica omogućuje izvedbu proizvoljne logičke funkcije od određenog broja ulaza. Izlaz iz jedne pregledne tablice može se direktno dovesti na ulaz memorijskog elementa odnosno bistabila ili ostvariti kompleksnije logičke funkcije povezivanjem s ulazima drugih preglednih tablica. Model konfigurabilnog logičkog bloka je prikazan na slici 2.3. [4]



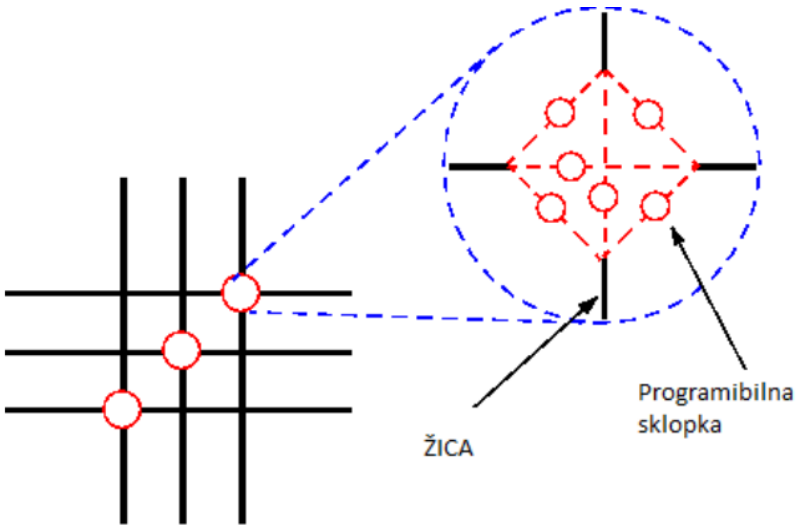
Sl. 2.3: Model programibilnog logičkog elementa. [4]

2.1.2. Konfigurabilni I/O blokovi

Dio FPGA sklopa koji služi za ulaz ili izlaz signala, odnosno za spajanje FPGA sklopa s ostalim elementima. I/O blokovi su spojeni s CLB blokovima putem programibilne interkonekcije. Sadrži izlazni i ulazni međuspremnik, na izlazima obično ima *pull up* otpornike, a ponekad i *pull down* otpornike. Izlazi I/O blokova mogu biti aktivni u visokom ili niskom naponskom stanju. [3]

2.1.3. Programibilne interkonekcije

Programibilne interkonekcije omogućuju vezu između logičkih blokova i I/O blokova kako bi se mogao implementirati korisnički definiran sklop. Sastoji se od žica i programibilnih sklopki koje formiraju željenu vezu. Slika 2.4 prikazuje dijelove programibilne interkonekcije. [3]

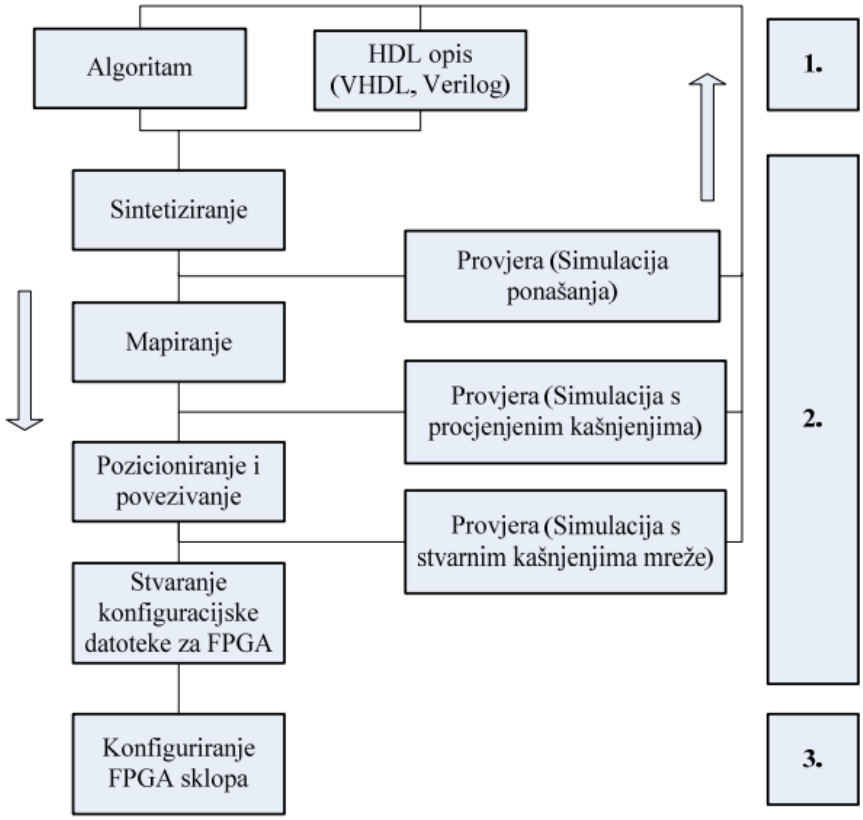


Sl. 2.4: Programibilna interkonekcija. [3]

2.1.4. Konfiguracija FPGA sklopa

Svaki logički blok FPGA sklopa radi paralelno i potpuno nezavisno od drugih, za razliku od mikroračunala u kojima jezgra mikroprocesora slijedno izvodi programske instrukcije jedu za drugom.

Konfiguracija FPGA sklopova se generira strojno uz pomoć specijaliziranih programskih alata, tzv. sintetizatora, a izvodi se na temelju opisa sklopa shemom ili korištenjem jezika za opis digitalnih sklopova (eng. *Hardware Description Language* – HDL). Najpoznatiji jezici za opis digitalnih sklopova su Verilog i VHDL (eng. *Very High Speed Integrated Circuit HDL*). Koraci sinteze, programiranja i konfiguracije FPGA sklopa mogu se podijeliti na tri glavna dijela, što je prikazano na slici 2.5. [4]



Sl. 2.5: Koraci sinteze, konfiguracije i programiranja FPGA sklopa.

Prvo se opisuje željeni sklop na način koji je prikladan za daljnju strojnu obradu, algoritmom ili korištenjem jezika za opis digitalnih sklopova.

Pri sintetiziranju se analizira opis sklopa te se opisani blokovi zamjenjuju funkcijski ekvivalentnim strukturnim modelima koji su izgrađeni od primitiva odnosno osnovnih logičkih sklopova kao što su *AND* i *OR*, ali primitivima također pripadaju kompliciraniji sklopovi kao što su posmačni registri i aritmetičke jedinice. Dobivena mreža primitiva naziva se i *netlist* i može se analizirati korištenjem određenog simulatora.

Nakon toga se primjenjuje mapiranje, pri čemu se *netlist* odnosno mreža dobivena sintetiziranjem mijenja mrežom preglednih tablica, memorijskim elementima te ostalim specijaliziranim logičkim blokovima koji odgovaraju FPGA platformi. U ovom koraku se može napraviti i simulacijski model s grubom procjenom kašnjenja pri propagaciji signala iz razloga što još nisu poznata kašnjenja mreže.

Tijekom izvođenja pozicioniranja i povezivanja sintetizator pokušava rasporediti logičke blokove na FPGA sklopu i povezati ih tako da omogući ispravan rad sklopa pri najvišoj mogućoj frekvenciji takta ili najmanjoj potrošnji, ovisno o postavkama sintetizatora. Nakon ovog koraka moguće je napraviti simulacijski model koji uključuje vrlo preciznu procjenu kašnjenja svih logičkih elemenata i mreže.

Krajnji rezultat procesa sinteze je konfiguracijska datoteka (eng. *Configuration bitstream*) kojom se uz pomoć odgovarajućeg alata može programirati FPGA sklop.

2.2. VHDL

VHDL je jezik kojim se opisuje ponašanje elektroničkih sklopova. Standardiziran je kroz *IEEE* 1076 standard, a koristi se za sintezu i simulaciju sklopova i primjenjuje se u području ASIC sklopova (eng. *Application Specific Integrated Circuit*) i programibilnih logičkih sklopova (eng. *Programmable Logic Devices* – PLD) u koje pripadaju CPLD sklopovi (eng. *Complex Programmable Logic Device*) i FPGA sklopovi. VHDL je paralelan jezik, za razliku od standardnih programskih jezika (C/C++, Basic, itd.), koji su sekvencijalni. [4]

VHDL kôd se sastoji od tri osnovne strukture:

- *Library* deklaracije – sadrži popis svih biblioteka koje se koriste u dizajnu,
- *Entity* - sadrži popis sa specifikacijama svih ulaznih i izlaznih pinova digitalnog sklopa,
- *Architecture* – sadrži VHDL kôd koji opisuje ponašanje sklopa. [4]

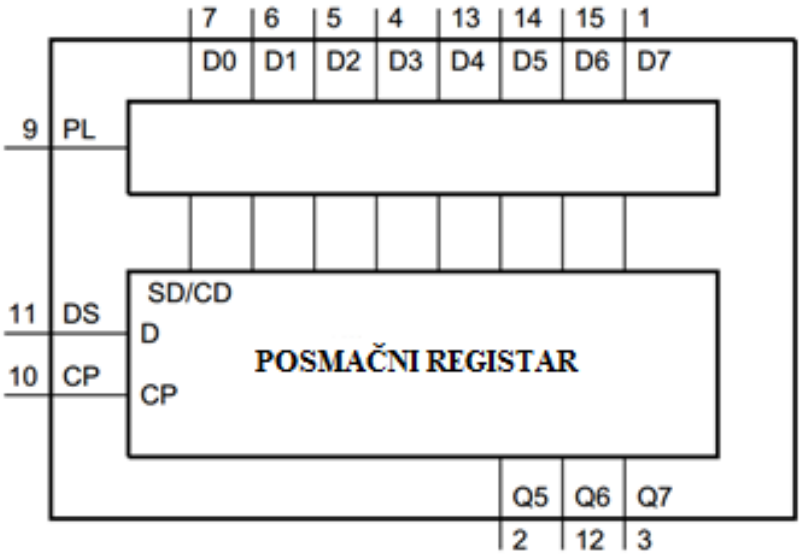
3. RAZVOJ DODATNOG SKLOPOVLJA

Cilj razvoja dodatnog sklopovlja je bio iskorištenje što većeg broja ulaza/izlaza s FPGA sklopa da bi sklopovlje imalo što veći broj programibilnih elemenata kako bi se proširilo korisničko sučelje E2LP platforme. Budući da korištenje svih željenih elemenata zahtijeva više nožica od mogućih 38, koji su dostupni na 50 pinskom konektoru, u sklopovlje su ukomponirani elementi koji su bazirani na serijskoj komunikaciji, tako da je u konačnici iskorišteno 34 nožice.

3.1. Sklopke

Kako bi se ostvarila interakcija između korisnika i E2LP platforme dostupno je ukupno 16 sklopki kojima je moguće unijeti 16-bitnu informaciju. S ciljem smanjenja broja potrebnih pinova korišteni su 4021 posmačni registri (eng. *Shift register*) koji su smanjili broj pinova sa 16 na 3.

Integrirani krug 4021 je 8-bitni posmačni registar koji paralelno čita stanja na sklopkama i nakon toga prosljeđuje pročitanu informaciju na sljedeći posmačni registar ili na neki drugi sklop kojem je namijenjena ta informacija. [5]



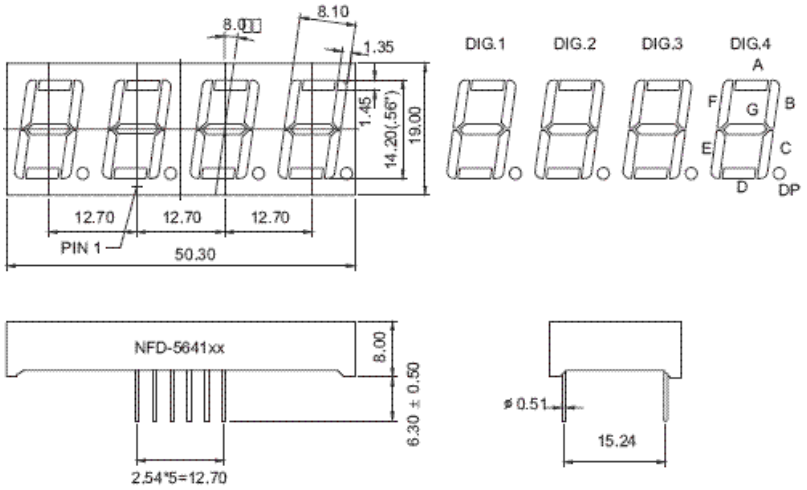
Sl. 3.1: Funkcijski dijagram 4021 posmačnog registra. [5]

Na slici 3.1 je prikazan funkcijski dijagram 4021 posmačnog registra. Nožice od D0 do D7 su paralelni ulazi posmačnog registra na koje se mogu spojiti sklopke ili tipkala. Kada se PL postavi u stanje *HIGH* učitavaju se stanja s paralelnih ulaza u posmačni registar. DS služi za serijski unos podataka i omogućuje spajanje više posmačnih registara, što je pogodno za realizaciju ovog sklopa. CP je signal takta ulaza dok su pinovi od Q5 do Q7 paralelni izlazi zadnja tri stanja u posmačnom registru.

Osim sklopki, na ulaze posmačnih registara se spajaju *pull down* otpornici koji služe za postavljanje ulaza u stanje logičke nule dok je sklopka isključena, kako ulaz ne bi ostao u neodređenom stanju. Vrijednost otpora tih otpornika je 10 *kΩ* i kako bi se uštedilo na prostoru koriste se mreže otpornika (eng. *Resistor network*) umjesto običnih otpornika.

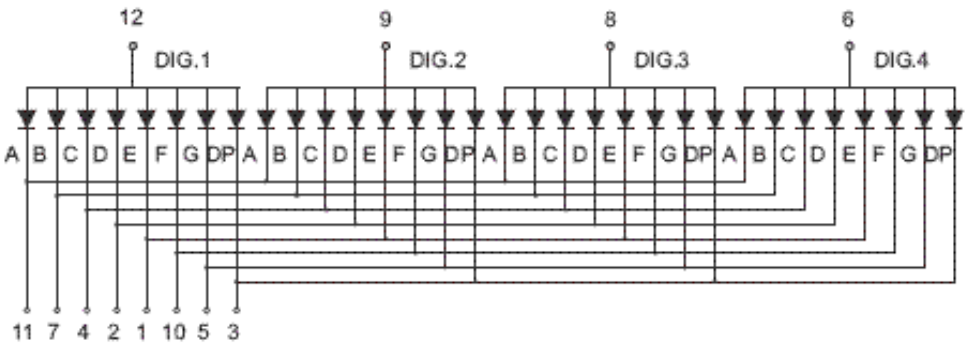
3.2. 7-segmentni pokaznici

Za prikaz brojeva koriste se 7-segmentni pokaznici. Na jednoj komponenti, koja je prikazana na slici 3.2, nalazi se četiri 7-segmentna pokaznika na kojima je moguće prikazati četiri znamenke. Budući da sva četiri pokaznika imaju zajednički spojenu anodu, pri programiranju treba primijeniti vremensko multipleksiranje kako bi se moglo prikazati četiri različite znamenke.



Sl. 3.2: Dimenzije NFD-5641xx serije.[1]

Na slici 3.3 je prikazan shematski prikaz komponente NFD-5641BH, na kojoj možemo vidjeti da svaki pokaznik ima zajednički spojenu anodu i da se sastoji od ukupno osam LE dioda (eng. *Light Emitting Diode*), od kojih LE diode od A do G služe za prikaz određene znamenke, dok DP LE dioda služi za prikaz decimalne točke.

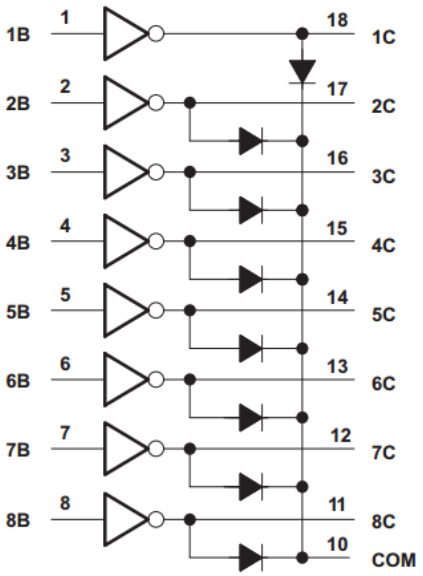


Sl. 3.3: Shematski prikaz komponente NFD-5641BH.[1]

[1]http://www.futurlec.com/LED/7FR5641BS.shtml

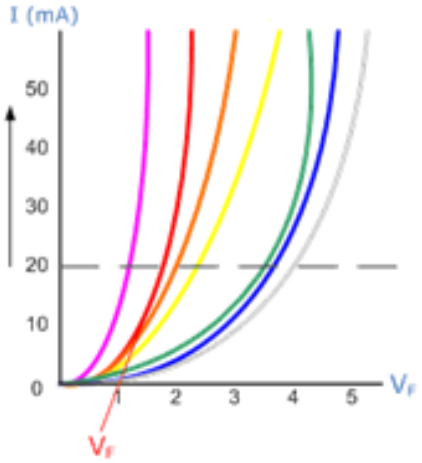
Prilikom testiranja primijećeno je da određeni ulazi na FPGA sklopu ne mogu primiti željenu količinu struje, stoga je ukomponiran integrirani krug ULN2803 čiji je logički dijagram prikazan na slici 3.4.

ULN2803 sadrži polje tranzistora odnosno 8 Darlingtonovih spojeva, koji, ovisno o stanju upravljačkih pinova od 1B do 8B, određuju hoće li kroz kolektore od 1C do 8C teći struja. Pojedini Darlingtonov spoj može provesti struju do 500 *mA*, a u slučaju potrebe za većom strujom moguće je više Darlingtonovih spojeva spojiti paralelno. [6]

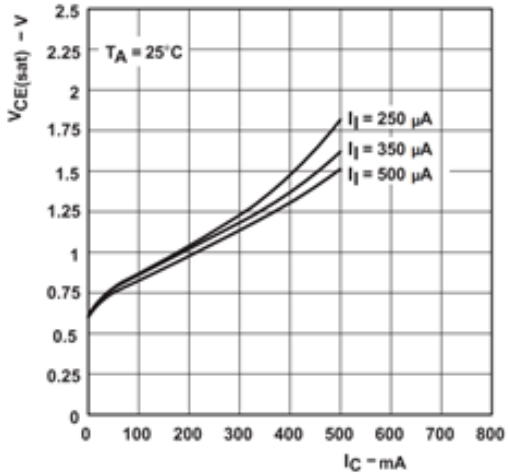


Sl. 3.4: Logički dijagram ULN2803. [6]

Kako bi se ograničila struja kroz pojedine diode, na pokaznicima su dodani otpornici. Iznos otpora je izračunat upotrebom formule 3-1, gdje je U napon koji daje izlaz na FPGA koji iznosi 3,3 *V*. U_D je prosječan pad napona na crvenoj LE diodi određuje se iz grafa na slici 3.5 i iznosi 1,2 *V* pri struji od 10 *mA*. U_{CE} je pad napona na jednom Darlingtonovom spoju, određuje se iz grafa na slici 3.6 i iznosi 0,6 *V* pri struji od 10 *mA*.



Sl. 3.5: U-I karakteristike LE dioda.[2]



Sl. 3.6: Odnos pada napona između kolektora i emitera sa strujom kroz jedan Darlingtonov spoj. [6]

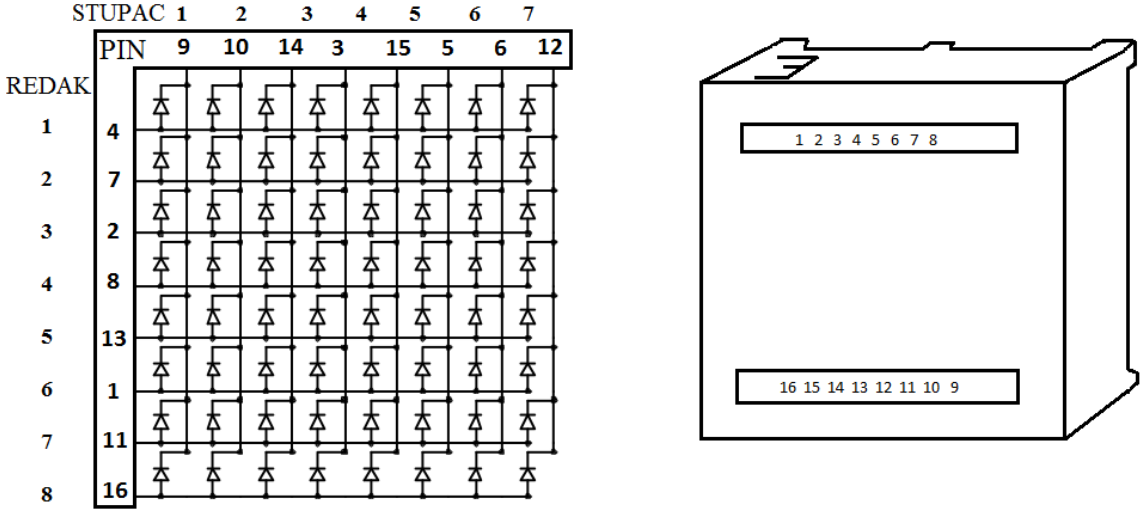
$$R = \frac{U - U_D - U_{CE}}{I} \qquad (3\text{-}1)$$

Nakon uvrštavanja zadanih vrijednosti dobije se vrijednost otpornika koja iznosi 150 *Ω*. Zbog prevelikog intenziteta svjetla kojeg su LE diode emitirale koriste se otpornici od 270 *Ω* nakon čega se struja smanjila na oko 5 *mA*.

[2] http://www.electronics-tutorials.ws/diode/diode_8.html

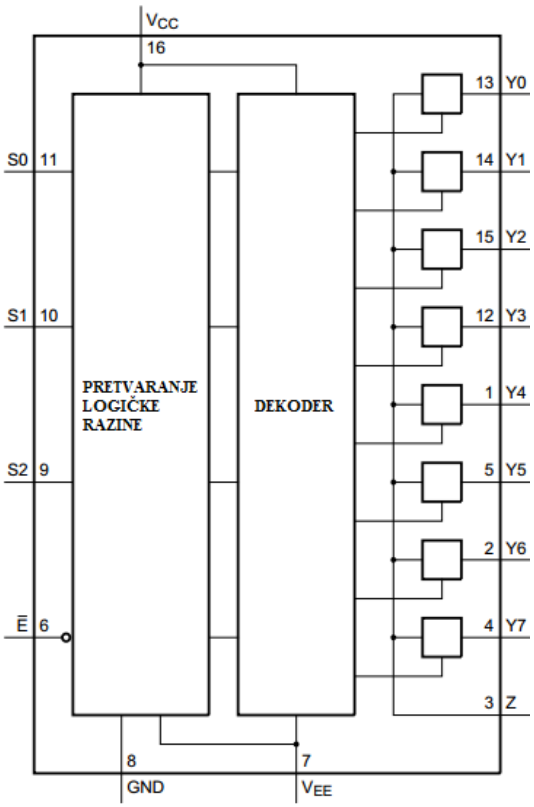
3.3. LED matrica

LED matrica (eng. *LED matrix*) sadrži ukupno 64 crvene LE diode za koje treba osigurati 16 nožica s FPGA kako bi se njima moglo upravljati. Raspored pinova LED matrice je prikazan na slici 3.7. Međutim, da bi se smanjio broj potrebnih nožica koriste se 4051 integrirani krugovi pa se broj upravljačkih pinova smanjio na 7.



Sl. 3.7: Raspored pinova LED matrice.

Integrirani krug 4051 je oktalna analogna sklopka pogodan za korištenje kao 8:1 multiplekser ili demultiplekser. Funkcijski dijagram tog integriranog kruga je prikazan na slici 3.8. Dovođenjem logičke jedinice na tri digitalna ulaza (S0, S1 i S2) određuje se koji će od osam nezavisnih ulaza/izlaza (od Y0 do Y7) biti uključen. Z pin je zajednički ulaz/izlaz. Postavljanjem pina E u stanje logičke jedinice sve sklopke će biti isključene. [7]



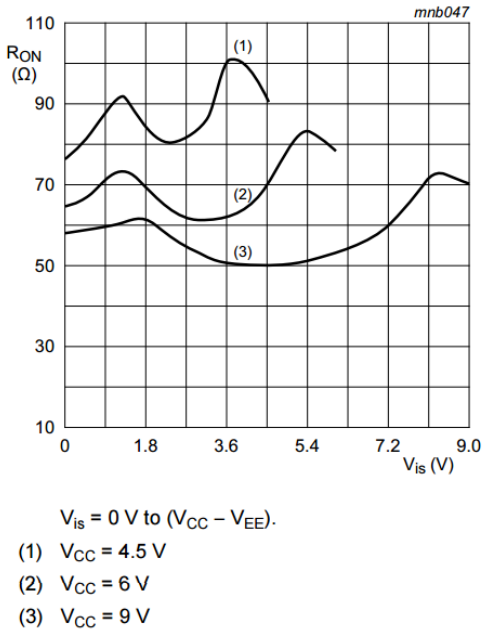
Sl. 3.8: Funkcijski dijagram 4051 multipleksera/demultipleksera. [7]

Dva 4051 integrirana sklopa su spojena na način da se na jednom određuje broj retka u kojoj će svijetliti LE dioda i taj daje 5 *V* na Z izlazu, dok drugi određuje u kojem će stupcu svijetliti dioda i taj ima na Z ulazu spojen *GND* uz otpornik koji ograničava struju. U jednom trenutku može svijetliti samo jedna LE dioda što znači da treba koristiti vremensko multipleksiranje kako bi se dobila željena slika na LED matrici.

$$R = \frac{U - U_D}{I} - 2 * R_{ON} \quad (3\text{-}2)$$

Korištenjem formule 3-2 se određuje potreban iznos otpora. U je napon na Z izlazu prvog 4051 integriranog kruga i iznosi 5 *V*. U_D je prosječa pad napona na crvenoj LE diodi koji se određuje iz grafa na slici 3.5 i iznosi 1,5 *V* pri struji od 10 *mA*. R_{ON} je otpor analogne sklopke koji se može iščitati s grafa na slici 3.9 i pri naponu od 5 *V* taj otpor iznosi oko 90 *Ω*, a budući da su u strujni krug uključene dvije sklopke istovremeno, treba poduplati otpor.

Nakon što se uvrste vrijednosti dobije se vrijednost otpora koja iznosi 170 *Ω*. Zbog dostupnosti se koristi otpor vrijednosti 150 *Ω*, a u tom slučaju struja iznosi 10,6 *mA*.

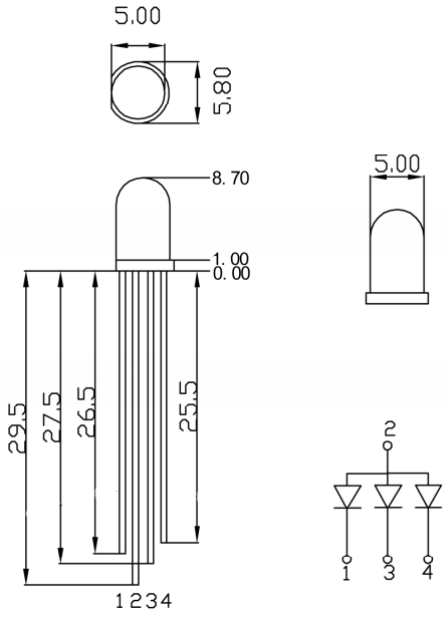


Sl. 3.9: Otpor analogne sklopke integriranog kruga 4051 pri naponu V_{is} [7]

3.4. RGB diode

Na sklopovlju su spojene tri RGB diode (eng. *Red, Green, Blue*) uz pomoć kojih se može programirati semafor i slično. Diode imaju zajedničku anodu i promjera su 5 *mm*, a dimenzije diode su prikazane na slici 3.10.

U ovom slučaju se za izračun otpornika koristi formula 3-3, ali se koriste različiti padovi napona za crvenu, zelenu i plavu LE diodu unutar RGB diode. Na slici 3.5 se vidi da je uobičajen pad napona pri 5 *mA* na crvenoj diodi iznosi 1,1 *V*, na zelenoj 2,2 *V*, a na plavoj oko 2,55 *V*. Uvrštavanjem vrijednosti padova napona, vrijednosti otpora iznose 270 *Ω* za crvenu, 220 *Ω* za zelenu i 150 *Ω* za plavu LE diodu.



Sl. 3.10: Dimenzije RGB diode.[3]

$$R = \frac{V - V_D}{I} \quad (3\text{-}3)$$

[3] https://cdn.sparkfun.com/datasheets/Components/LED/YSL-R596AR3G4B5C-C10.pdf

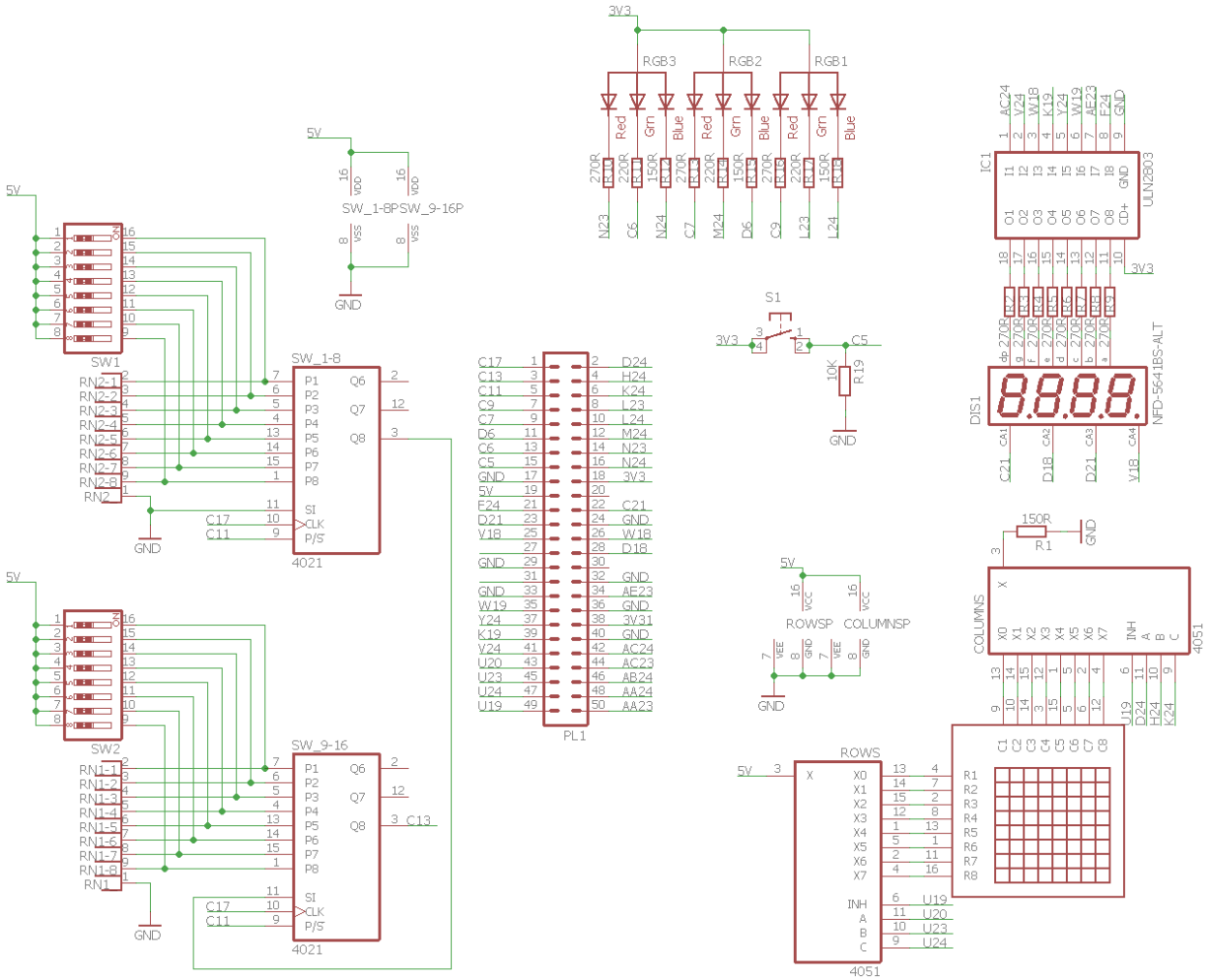
3.5. Tipkalo

Jedno tipkalo je spojeno na C5 nožicu i predviđeno je da se koristi kao *reset*. Da nožica ne bi ostala u neodređenom stanju dok tipkalo nije pritisnuto, spojen je *pull down* otpornik s otporom od 10 *k*Ω.

3.6. Shema sklopovlja

Shema sklopovlja izrađena u programu *EAGLE* je prikazana na slici 3.11. Pojedini elementi su spojeni kao što je objašnjeno u prijašnjim potpoglavljima.

Bitno je napomenuti da se na sredini sheme nalazi 50-pinski konektor kojim se sklopovlje spaja s E2LP razvojnim sučeljem. Na taj konektor su spojeni svi elementi sklopovlja, a nazivi svake nožice konektora odgovaraju nazivu nožica na FPGA sklopu, stoga je na shemi moguće vidjeti koje je nožice potrebno koristiti kako bi se upravljalo određenim elementom sklopovlja.



Sl. 3.11: Shema sklopovlja izrađena u programu EAGLE.

3.7. Nožice FPGA sklopa za UCF datoteku

U tablici 3.1 nalaze se nožice FPGA sklopa za pojedine elemente sklopovlja koje treba navesti u UCF datoteci pri programiranju. Signal takta frekvencije 24 *MHz* nalazi se na nožici M21.

Tab. 3.1: Nožice FPGA sklopa za pojedine elemente sklopovlja.

7 segmentni pokaznici		RGB diode		LED matrica		Sklopke	
CA1	C21	**RGB1**		**E**	U19	**CLK**	C17
CA2	D18	**Crvena**	C9	**Retci**		**PL**	C11
CA3	D21	**Zelena**	L23	**S0**	U20	**Q**	C13
CA4	V18	**Plava**	L24	**S1**	U23	**Tipkalo**	
a	F24	**RGB2**		**S2**	U24	**RST**	C5
b	AE23	**Crvena**	C7	**Stupci**			
c	W19	**Zelena**	M24	**S0**	D24		
d	Y24	**Plava**	D6	**S1**	H24		
e	K19	**RGB3**		**S2**	K24		
f	W18	**Crvena**	N23				
g	V24	**Zelena**	C6				
dp	AC24	**Plava**	N24				

4. IMPLEMENTACIJA I TESTIRANJE NA EKSPERIMENTALNOJ PLOČICI

Prije testiranja sklopovlja potrebno je napisati kôd u VHDL-u kojim će biti testiran svaki element sklopovlja i koji će potom biti implementiran na E2LP platformu. Xilinx je 2013. godine izdao posljednju inačicu programa *ISE Design Suite* 14.7 u kojem je napisan VHDL kôd. [8]

4.1. VHDL kôd

Na početku VHDL kôda su deklarirane biblioteke kao što je prikazano na slici 4.1. Nakon toga je napisan *Entity* u kojem su definirani svi ulazi i izlazi u odnosu na FPGA.

Linija ***Kod***

```
1:  library IEEE;
2:  use IEEE.STD_LOGIC_1164.ALL;
3:  entity All_in_one is
4:  Port (
5:          Clock           : in     STD_LOGIC;
6:          Reset           : in     STD_LOGIC;
7:          -- Switches
8:          CLK             : out    STD_LOGIC := '0';
9:          Input_data      : in     STD_LOGIC;
10:         Parallel_load   : buffer STD_LOGIC := '0';
11:         -- Matrix
12:         Columns         : out    STD_LOGIC_VECTOR (2 downto 0);
13:         Rows            : out    STD_LOGIC_VECTOR (2 downto 0);
14:         Enable          : out    STD_LOGIC := '1';
15:         -- RGB
16:         RGB1            : out    STD_LOGIC_VECTOR (2 downto 0);
17:         RGB2            : out    STD_LOGIC_VECTOR (2 downto 0);
18:         RGB3            : out    STD_LOGIC_VECTOR (2 downto 0);
19:         -- 7-seg
20:         Common_anodes   : buffer STD_LOGIC_VECTOR (3 downto 0);
21:         Segments        : out    STD_LOGIC_VECTOR (7 downto 0));
22: end All_in_one;
```

Sl. 4.1: Library i Entity dio VHDL kôda.

U liniji 5 je deklariran ulazni signal *Clock* odnosno signal takta koji je dostupan na E2LP platformi i koji mijenja stanja frekvencijom od 24 *MHz*. Nakon toga je deklariran signal *Reset* čije stanje ovisi o tomu je li tipkalo na sklopovlju pritisnuto ili ne. Upotrebom operatora ':=' se postavlja početna vrijednost signala, ali ona vrijedi samo za simulaciju. Ova dva signala su *STD_LOGIC* tip

podatka, odnosno logički sustav koji može poprimiti osam različitih vrijednosti (prema standardu IEEE 1164), a te vrijednosti su sljedeće:

- 'X' prisilno nepoznato stanje (eng. *forcing unknown*),
- '0' prisilno nisko stanje,
- 1' prisilno visoko stanje,
- 'Z' visoka impedancija,
- 'W' slabo nepoznato stanje (eng. *weak unknown*),
- 'L' slabo nisko stanje,
- 'H' slabo visoko stanje,
- '-' nije bitno. [9]

U linijama od 8 do 10 su deklarirani signali koji su potrebni kako bi se upravljalo unosom stanja sa sklopki. Bitno je napomenuti da *Parallel_load* signal ima mod *buffer*, što omogućuje čitanje stanja na tom izlazu, što nije moguće kada je mod signala *out*.

Nakon toga su deklarirani signali za upravljanje LED matricom. Signali *Columns* i *Rows* su definirani kao *STD_LOGIC_VECTOR (2 downto 0)* što znači da je riječ o 3-bitnom vektoru s krajnjim lijevim bitom kao najznačajnijim. Moguće je koristiti i *0 to 2* soluciju, ali u tom slučaju bi najznačajniji bit bio krajnji desni.

U slijedeće tri linije su deklarirana tri 3-bitna vektora kojima se upravlja RGB diodama, tako što svaki bit u signalu određuje hoće li određena boja na RGB diodi svijetliti ili ne.

Naposljetku su definirani 4-bitni vektor *Common_Anodes* kojim se uključuje određeni pokaznik postavljanjem određenog bita u logičku jedinicu i 8-bitni vektor *Segments* kojim se definira koje će LE diode na pokazniku svijetliti.

Na slici 4.2 je prikazan deklarativni dio strukture *Architecture*. Deklarirano je nekoliko signala *STD_LOGIC* tipa podatka za različite frekvencije signala takta, dva vektora veličine 16 *b* i 64 *b* kako bi se mogla pohraniti stanja pojedinih sklopki i stanja pojedinih LE dioda LED matrice. Na kraju je deklariran signal tipa *INTEGER* koji može poprimiti vrijednosti od 0 do 9.

Linija ***Kod***

```
1:  Architecture Behavioral of All_in_one is
2:          signal clk_10kHz:       STD_LOGIC;
3:          signal clk_1Hz:         STD_LOGIC;
4:          signal clk_10Hz:        STD_LOGIC;
5:          signal SW_data:         STD_LOGIC_VECTOR (15 downto 0);
6:          signal LEDs:            STD_LOGIC_VECTOR (0 to 63);
7:          signal Number:          INTEGER range 0 to 9;
8:          begin
```

Sl. 4.2: Architecture – deklarativni dio.

U dijelu u kojem se piše kôd koji opisuje ponašanje sklopovlja se na više mjesta izvodi povezivanje više VHDL kodova iz razloga da bi kôd bio pregledniji. Budući da se koriste signali takta različitih frekvencija, treba se koristiti djelitelj frekvencije kako bi se od ulaznog signal takta dobilo signal željene frekvencije. Primjer na kojem je vidljivo dobivanje signala takta 10 *kHz* je prikazan na slici 4.3. U *generic map* se upisuje broj kojim želimo podijeliti ulaznu signal takta, a u *port map* upisujemo redom ulazni signal, izlazni signal i *Reset*.

Linija	***Kod***
1:	`f10khz: entity work.freq_div generic map(2_400) port map(Clock, clk_10kHz, Reset);`

Sl. 4.3: Pozivanje djelitelja frekvencije.

Na slici 4.4 je prikazan kôd koji upravlja 7-segmentnim pokaznicima na način da se u procesu, koji se izvodi između linija 1 i 10, signal *Number* mijenja deset puta u sekundi, a taj signal određuje što će biti prikazano na pokaznicima. To određivanje se provodi uz pomoć *with select* naredbe koja je napisana nakon linije 12. Postavljanje signala *Common_anodes* na „1111“ označuje istovremeni rad sva 4 pokaznika.

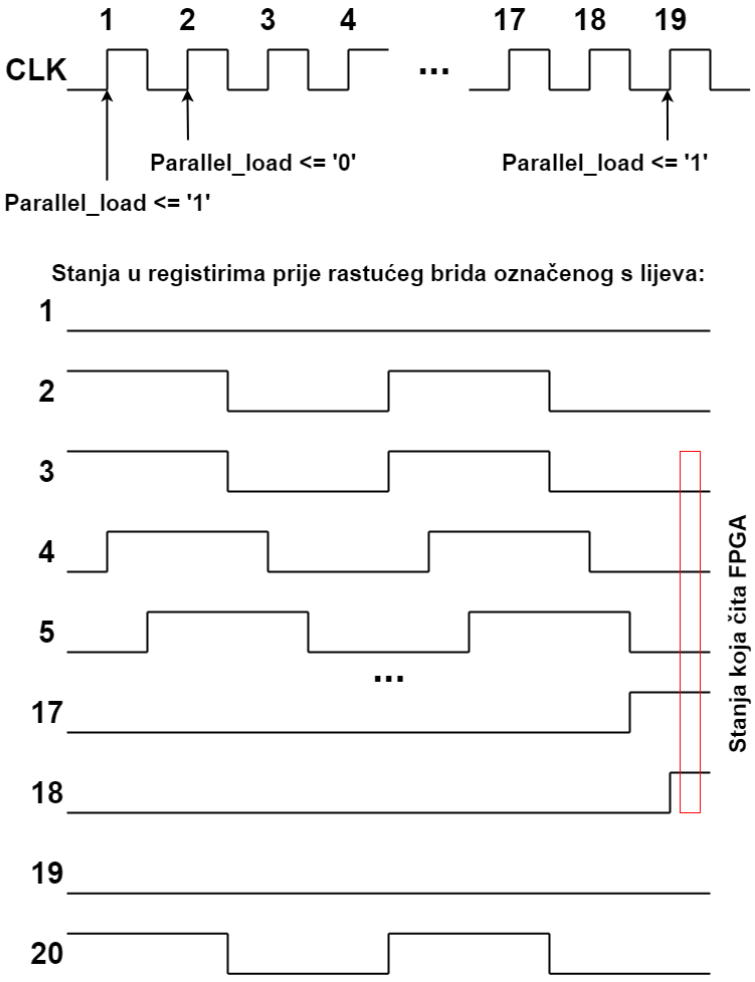
Linija	***Kod***
1:	`process(clk_10Hz, Reset)`
2:	`variable tmp: integer range 0 to 10 := 0;`
3:	`begin`
4:	`if clk_10Hz'event AND clk_10Hz = '1' then`
5:	`tmp := tmp+1;`
6:	`if tmp > 9 then`
7:	`tmp := 0;`
8:	`end if;`
9:	`end if;`
10:	`Number <= tmp;`
11:	`end process;`
12:	`Common_anodes <= "1111";`
13:	`with Number select`
14:	`Segments <= "10000000" when 0,`
15:	`"01000000" when 1,`
16:	`"00100000" when 2,`
17:	`"00010000" when 3,`
18:	`"00001000" when 4,`
19:	`"00000100" when 5,`
20:	`"00000010" when 6,`
21:	`"00000001" when 7,`
22:	`"00000000" when 8,`
23:	`"11111111" when 9;`

Sl. 4.4: Kôd koji opisuje ponašanje 7-segmentnih pokaznika.

Linija	*Kod*
1:	`read_states_of_switches: entity work.SW_read port map(clk_10kHz, Input_data, Parallel_load, SW_data, Reset);`

Sl. 4.5: Instanciranje sklopa za čitanje stanja na sklopkama.

Na slici 4.5 je prikazana linija koda u kojoj se izvodi instanciranje sklopa za čitanje stanja na sklopkama. U *port map*-u se predaju signali koji su potrebni za rad tog sklopa. *clk_10kHz* određuje brzinu čitanja sklopki. U ovom slučaju radi se o signalu takta frekvencije 10 *kHz*. U signal *Input_data* se sprema trenutna vrijednost koja je pročitana na određenoj sklopki, dok se u *SW_data* spremaju vrijednosti svih sklopki. *Parallel_load* je signal koji se postavlja u logičku jedinicu kada se žele učitati vrijednosti sklopki u posmačni registar 4021. Sklop radi tako što na svaki rastući brid očita vrijednost sklopke i tako šesnaest puta, nakon čega postavlja signal *Parallel_load* u logičku jedinicu te se nakon toga ponavlja isti postupak. Primjer čitanja sklopki koje su postavljene u stanja „1111000011110000“ je vidljiv na slici 4.6.



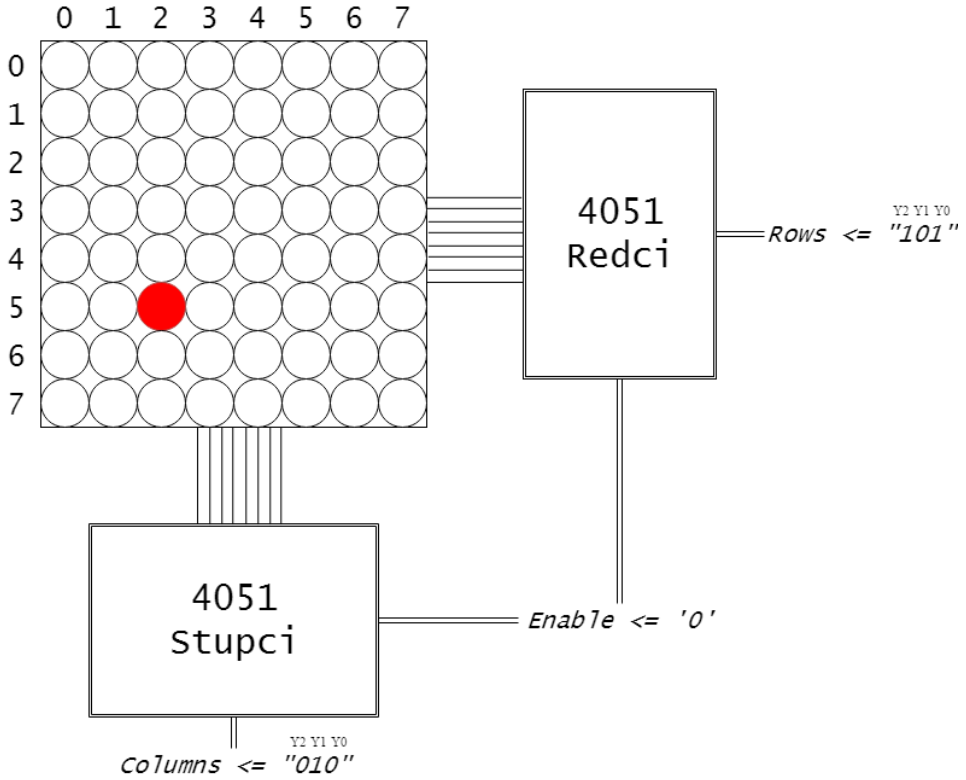
Sl. 4.6: Primjer čitanja stanja na sklopkama.

Linija ***Kod***

```
1:  update_matrix: entity work.Matrix port map(clk_10kHz, LEDs,
    Columns, Rows, Enable, Reset);
```

Sl. 4.7: Instanciranje sklopa za upravljanje LED matricom.

Slika 4.7 prikazuje liniju u kojoj se instancira sklop za upravljanje LED matricom, koja također prima signal takta frekvencije 10 *kHz* te se pri toj frekvenciji upravlja integriranim krugom 4051. Također prima signal *LEDs* koji je zapravo vektor veličine 64 bita, a svako stanje u tom vektoru predstavlja stanje određene LE diode matrice. Signali *Columns* i *Rows* određuju u kojem stupcu i u kojem retku će LE dioda biti uključena postavljanjem ulaza 4051 integriranog kruga Y0, Y1 i Y2 u određeno stanje, dok *Enable* služi za uključivanje i isključivanje cijele matrice. Kada su sve LE diode matrice isključene, signal *Enable* se postavlja u logičku jedinicu. Primjer rada LED matrice prikazan je na slici 4.8.



Sl. 4.8: Primjer rada LED matrice.

U liniji na slici 4.9 se izvodi instanciranje sklopa za upravljanje RGB diodama. Prvi signal koji prima određuje brzinu kojom će se izvoditi promjene na RGB diodama. U ovom slučaju će se promjene izvoditi svake sekunde. U *RGB1*, *RGB2* i *RGB3* signale se spremaju stanja svake pojedine boje na RGB diodi.

Linija ***Kod***

```
1:  RGB_leds: entity work.RGB_animation port map(clk_1Hz, RGB1,
    RGB2, RGB3, Reset);
```

Sl. 4.9: Instanciranje sklopa za upravljanje RGB diodama.

Pri kodiranju animacije za RGB diode korištena je tehnika za kodiranje automata s konačnim brojem stanja (eng. *Finite State Machine*) odnosno FSM dizajn. Ta tehnika se koristi kada logički sklop ima jasno definirana stanja. Postoje dva tipa FSM-a, Maleyev i Mooreov. Kod Mooreovog tipa izlaz ovisi samo o trenutnim stanjima, dok kod Maleyevog ovisi i o trenutnim ulazima. FSM se sastoji od sekvencijalnog dijela, koji određuje trenutno stanje sklopa i kombinacijskog dijela koji određuje izlaze i buduće stanje na temelju ulaza i trenutnog stanja. [10]

Slika 4.10 prikazuje VHDL kod koji izvodi animaciju za RGB diode s FSM pristupom. Prvi proces, koji se nalazi između prve i osme linije, je sekvencijalni dio koda, dok je drugi proces kombinacijski dio koda. Svaka boja pojedine RGB diode je aktivna na logičku nulu. Pri izvođenju animacije prvo se redom uključuje svaka boja svake diode pojedinačno te se nakon toga uključuje bijela boja na sve tri RGB diode, odnosno sve boje na RGB diodama se postavljaju u logičku nulu.

Linija ***Kod***

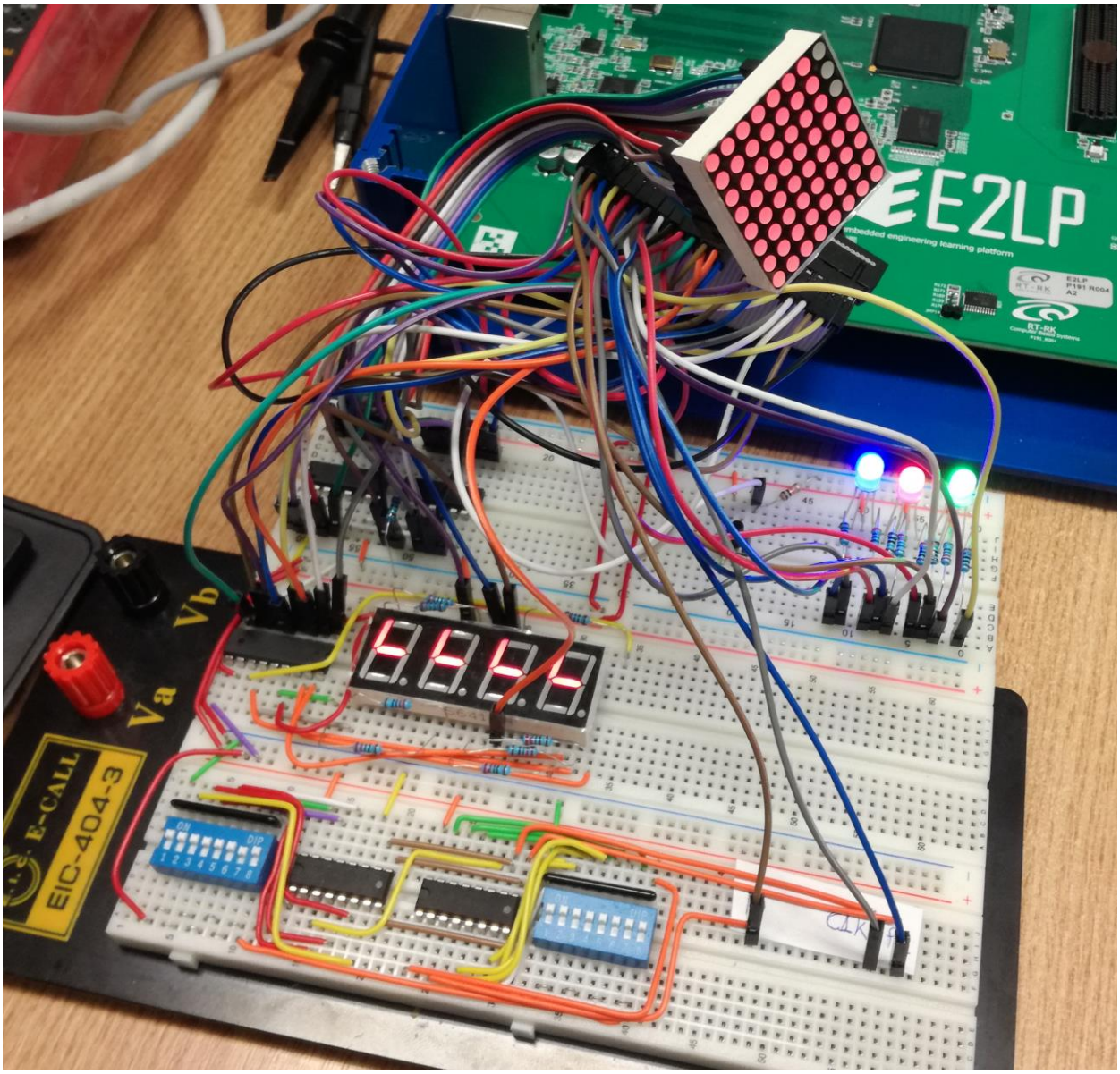
```
1:  process(Clock, Reset)
2:          begin
3:          if Reset = '1' then
4:                  sad_st <= S1;
5:          elsif (Clock'event and Clock = '1') then
6:                  sad_st <= slj_st;
7:          end if;
8:  end process;
9:  process(sad_st)
10:         begin
11:         case sad_st is
12:         when S1 => L1 <= "111"; L2 <= "111"; L3 <= "111"; slj_st<=S2;
13:         when S2 => L1 <= "011"; L2 <= "111"; L3 <= "111"; slj_st<=S3;
14:         when S3 => L1 <= "101"; L2 <= "111"; L3 <= "111"; slj_st<=S4;
15:         when S4 => L1 <= "110"; L2 <= "111"; L3 <= "111"; slj_st<=S5;
16:         when S5 => L1 <= "000"; L2 <= "011"; L3 <= "111"; slj_st<=S6;
17:         when S6 => L1 <= "000"; L2 <= "101"; L3 <= "111"; slj_st<=S7;
18:         when S7 => L1 <= "000"; L2 <= "110"; L3 <= "111"; slj_st<=S8;
19:         when S8 => L1 <= "000"; L2 <= "000"; L3 <= "011"; slj_st<=S9;
20:         when S9 => L1 <= "000"; L2 <= "000"; L3 <= "101"; slj_st<=S10;
21:         when S10 => L1 <= "000"; L2 <= "000"; L3 <= "110"; slj_st<=S11;
22:         when S11 => L1 <= "011"; L2 <= "101"; L3 <= "110"; slj_st<=S12;
23:         when S12 => L1 <= "101"; L2 <= "110"; L3 <= "011"; slj_st<=S13;
24:         when S13 => L1 <= "110"; L2 <= "011"; L3 <= "101"; slj_st<=S14;
25:         when S14 => L1 <= "000"; L2 <= "000"; L3 <= "000"; slj_st<=S1;
26:         end case;
27: end process;
```

Sl. 4.10: FSM dizajn – animacija za RGB diode.

4.2. Spajanje sklopovlja na eksperimentalnoj pločici

Prije izrade tiskane pločice potrebno je testirati sklopovlje na eksperimentalnoj pločici kako bi se utvrdilo radi li sve kako je zamišljeno. Testiranje se provodi spajanjem sklopovlja na E2LP razvojno sučelje korištenjem 50-pinskog konektora. Nakon spajanja sklopovlja se implementira VHDL program na FPGA sklopu. U slučaju da nešto ne funkcionira, treba napraviti izmjene na sklopovlju i/ili VHDL kodu i ponovno provesti testiranje.

Nakon nekoliko izmjena sve komponente sklopovlja su radile kao što je vidljivo na slici 4.11.

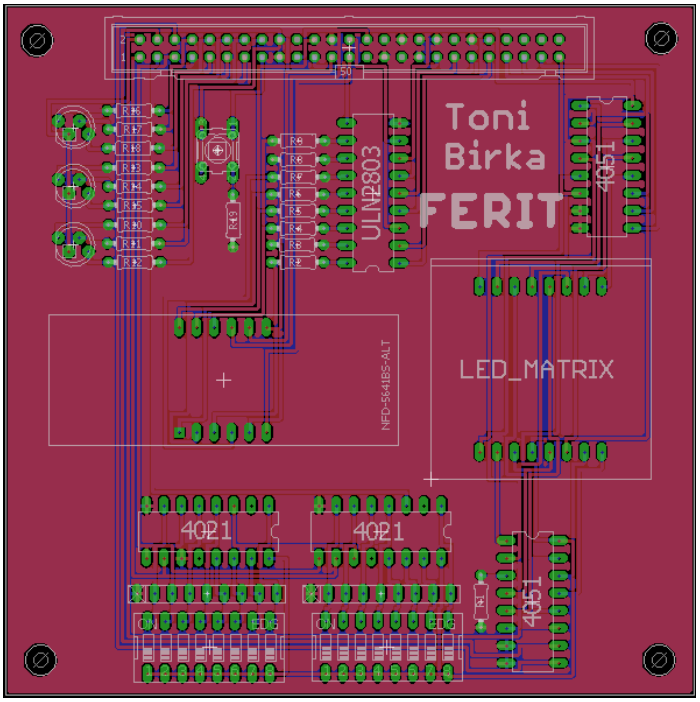


Sl. 4.11: Testiranje na eksperimentalnoj pločici.

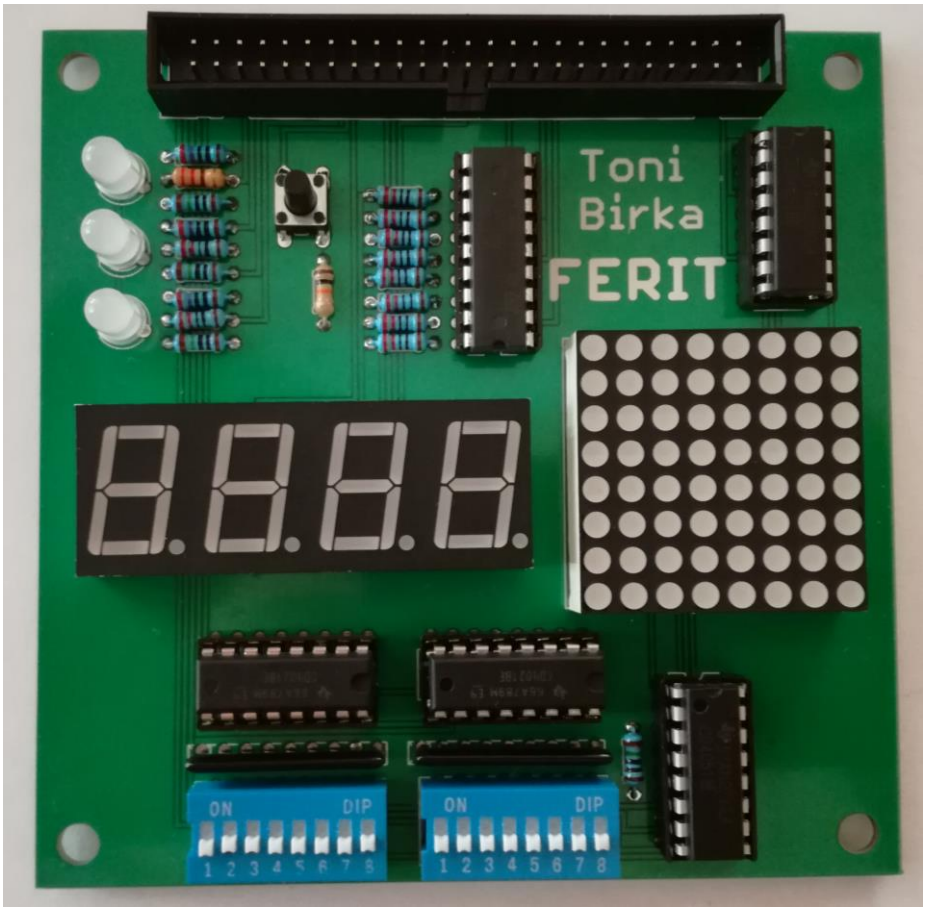
5. DIZAJNIRANJE I TESTIRANJE TISKANE PLOČICE

Nakon što je sklopovlje uspješno testirano na eksperimentalnoj pločici, dizajnirana je tiskana pločica u programu *EAGLE* što je prikazano na slici 5.1. Dizajnirana je na dva sloja kako bi se smanjila površina same pločice. Konačne dimenzije pločice su 10 *cm* × 10 *cm*.

Za izradu pločica je korištena usluga *ITEAD Studio* koji su izradili deset komada. Nakon isporuke naručenih pločica, svaki element je ručno zalemljen. Na slici 5.2 je prikazana tiskana pločica sa zalemljenim svim elementima.

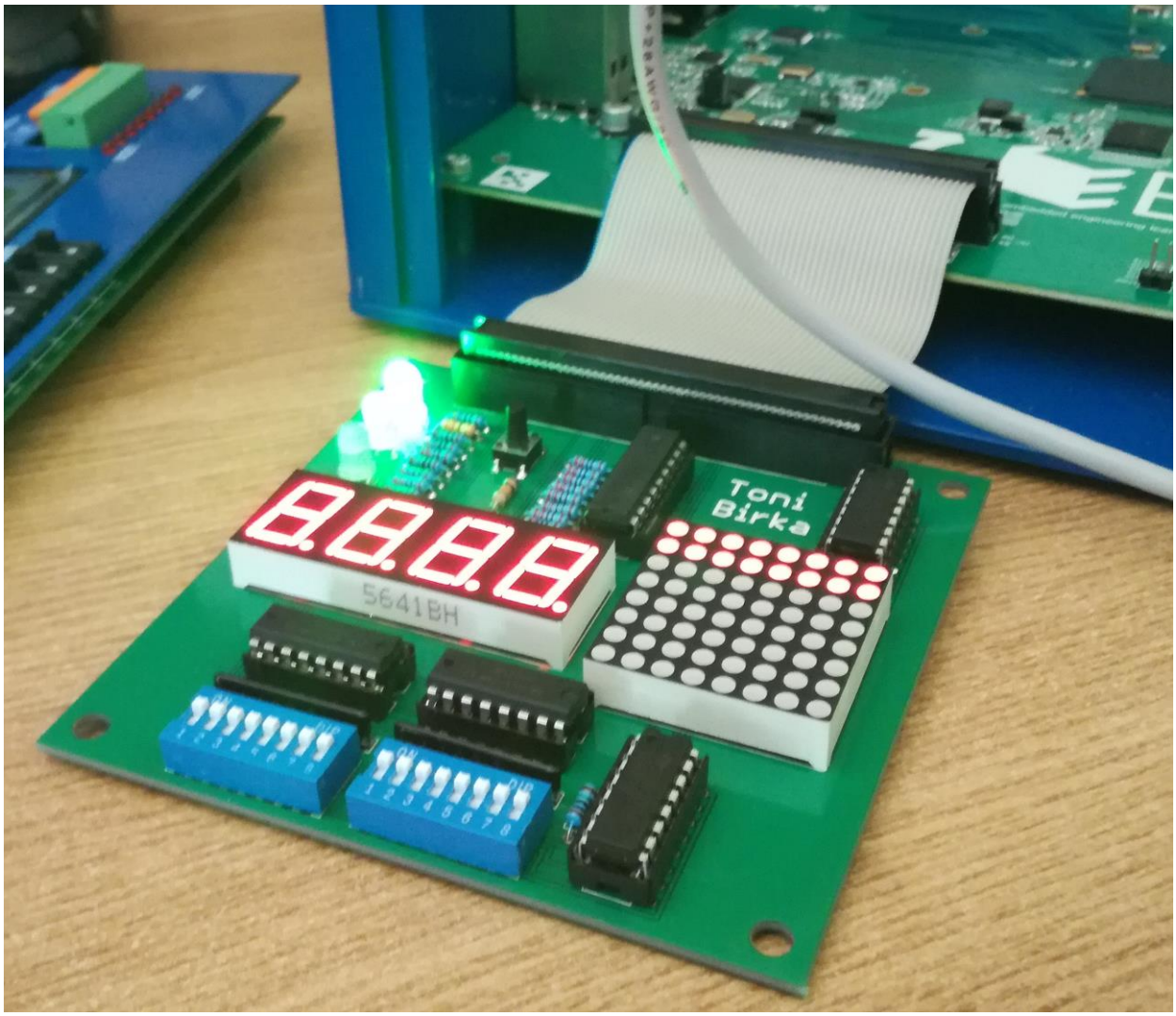


Sl. 5.1: Dizajn pločice u programu EAGLE.



Sl. 5.2: Tiskana pločica.

Na kraju treba provesti testiranje kako bi se utvrdilo je li sve funkcionalno. Testiranje se provodi spajanjem sklopovlja na E2LP razvojno sučelje koristeći 50-pinski konektor. Nakon toga se E2LP povezuje s računalom preko USB sučelja pa se vrši konfiguriranje FPGA sklopa. Svi elementi na dodatnom sklopovlju su bili funkcionalni pa nije bilo potrebno raditi dodatne izmjene. Rezultati testiranja se vide na slici 5.3.



Sl. 5.3: Testiranje tiskane pločice.

6. ZAKLJUČAK

U ovom završnom radu osmišljeno je dodatno LED sklopovlje za E2LP razvojni sustav koje je predviđeno za edukaciju studenata kroz različite eksperimente. Sklopovlje omogućuje korisnicima upotrebu 16 sklopki, 8x8 LED matrica, četiri 7-segmentna pokaznika i tri RGB diode pri programiranju FPGA sklopa. Time se proširuje korisničko sučelje u odnosu na osnovno korisničko sučelje koje je dostupno na E2LP razvojnom sustavu.

U radu su opisane funkcije osnovne E2LP ploče i pripadajućeg FPGA sklopa. Opisan je FPGA sklop i svi njegovi glavni dijelovi. Objašnjeni su svi koraci tijeka konfiguracije sklopa, a potom je definiran VHDL jezik za opisivanje elektroničkih sklopova.

Nakon toga su detaljnije opisane karakteristike svakog elementa sklopovlja, na koji način funkcioniraju i zašto se koriste elementi koji su bazirani na serijskoj komunikaciji, odnosno posmačni registri. Provedeni su proračuni za izračun vrijednosti otpornika za ograničenje struje kroz LE diode. Pojedini dijelovi razvijenog VHDL koda detaljno su objašnjeni te je provedeno testiranje na eksperimentalnoj pločici.

Na kraju je prikazan dizajn tiskane pločice izrađen u *EAGLE* programu. Izrađena tiskana pločica uspješno je testirana s razvijenim VHDL kodom.

LITERATURA

[1] E2LP, http://www.rt-rk.com/news /142-embedded-engineering-learning-platform, pristupljeno 30. travnja 2017.

[2] FPGA osnove, http://www.ni.com/white-paper/6983/en/, pristupljeno 30. travnja 2017.

[3] T. Matić, Auditorne vježbe 4, Digitalna elektronika

[4] V. A. Pedroni, Circuit Design with VHDL, MIT Press, 2004.

[5] Posmačni registar 4021, http://assets.nexperia.com/documents/data-sheet/HEF4021B.pdf, pristupljeno 10. prosinca 2016.

[6] Polje tranzistora, http://www.ti.com/lit/ds/symlink/uln2803a.pdf, pristupljeno 15. veljače 2017.

[7] Analogni multiplekser/demultiplekser 4051, http://assets.nexperia.com/documents/data-sheet/74HC_HCT4051.pdf, pristupljeno 10. prosinca 2016.

[8] ISE Design Suite 14.7, https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html, pristupljeno 23. siječnja 2017.

[9] STD LOGIC tip podatka, http://www.vhdl.renerta.com/mobile/source/vhd00067.htm, pristupljeno 30. travnja .2017.

[10] T. Matić, Auditorne vježbe 6, Digitalna elektronika

SAŽETAK

Cilj razvoja dodatnog LED sklopovlja za E2LP razvojni sustav je bio povećati broj programibilnih elemenata kako bi se proširilo korisničko sučelje navedenog razvojnog sustava. To je ostvareno korištenjem oktalnih analognih sklopki i komponenti koje se baziraju na serijskoj komunikaciji odnosno korištenjem posmačnih registara. E2LP je platforma koja služi za edukaciju inženjera o dizajnu ugradbenih sustava koji stječu znanje kroz razne eksperimente. E2LP se temelji na Xilinx Spartan 6 FPGA sklopu, a FPGA je programibilni integrirani sklop kojim se veliki broj jednostavnih programibilnih logičkih elemenata može povezati u kompleksni sklop. FPGA se konfigurira VHDL kôdom koji služi za opisivanje ponašanja elektroničkih sklopova. Razvijeni VHDL kod i pločica omogućuju korisniku programiranje 8x8 LED matrice, četiri 7- segmentna pokaznika, tri RGB diode te korištenje 16 sklopki i jednog tipkala.

Ključne riječi: E2LP platforma, FPGA, VHDL, dodatno sklopovlje, *EAGLE*.

ABSTRACT

Title: Expansion board for E2LP

The goal of developing expansion LED board for E2LP platform was to increase the number of programmable elements to expand the user interface of the said development system. That was achieved using octal analog switches and components based on serial communication or in other words, using shift registers. E2LP is a platform which is used to educate engineers about embedded systems design who gain knowledge through various experiments. E2LP is based on Xilinx Spartan 6 FPGA, and FPGA is a programmable integrated circuit that allows a large number of simple programmable logic elements to connect into a complex circuit. The FPGA is configured with a VHDL code which is used to describe the behavior of electronic circuits. Ultimately, the expansion board enables the user to program 8x8 LED matrix, four 7-segment displays, three RGB diodes and the use of 16 switches and one pushbutton.

Keywords: E2LP, FPGA, VHDL, expansion board, EAGLE.

ŽIVOTOPIS

Toni Birka je rođen 16. rujna 1995. godine u Virovitici. Živi u Končanici, u blizini Daruvara. Završio je Češku osnovnu školu J. A. Komenskog u Daruvaru, nakon čega upisuje Tehničku školu Daruvar, smjer tehničar za računarstvo. U četvrtom razredu srednje škole je osvojio treće mjesto na županijskom natjecanju iz matematike. Radom „RGB Display“ 5. svibnja 2014. godine je osvojio treće mjesto na Državnoj smotri radova u Obrazovnom sektoru elektrotehnika i računarstvo za školsku godinu 2013./2014., koja je bila održana na FER-u u Zagrebu. Istim radom je 10. svibnja 2014. godine osvojio zlato za uspješan nastup na izložbi INOVA MLADI 2014. na FSB-u u Zagrebu. Nakon završetka srednjoškolskog obrazovanja je izravno upisao preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Dobiva priznanje i nagradu za postignut uspjeh u studiranju 20. svibnja 2016. godine. Sudjelovao je 22. listopada 2016. godine na dvadesetčetverosatnom natjecanju u programiranju IEEE Extreme. Sudjelovao je na natjecanju iz informatike na Elektrijadi u Budvi 2017. godine. Dvije i pol godine je primao stipendiju Bjelovarsko-bilogorske županije, a od ožujka 2017. godine je stipendist tvrtke Comping iz Zagreba.

PRILOZI

P1 VHDL kôd koji se implementira na E2LP razvojni sustav, nalazi se na CD-u.

P2 .sch datoteka koja se otvara uz pomoć *EAGLE* programa, u kojoj je prikazana shema dodatnog LED sklopovlja za E2LP, nalazi se na CD-u.

P3 .brd datoteka koja se otvara uz pomoć *EAGLE* programa, u kojoj je prikazan dizajn tiskane pločice dodatnog LED sklopovlja za E2LP, nalazi se na CD-u.