

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Seminarski rad Osnove računarske tehnike 2

Tema:
Simulacija parkinga na Spartan 3E FPGA ploči

Studenti:
Vukašin Paunović 552/2015
Kosta Erić 561/2015

Predmetni profesor:
Aleksandar Peulić

SADRŽAJ

1. UVOD	2
2. ARHITEKTURA	4
2.1. IZVORI NAPAJANJA	5
2.2. MEMORIJA.....	6
2.3. ULAZNI/IZLAZNI UREĐAJI (LED diode, prekidači i tasteri)	7
2.4. SEDMO-SEGMENTNI DISPLEJ	8
2.5. VGA PORT.....	10
2.6. USB PORT	11
2.7. CLOCK.....	11
3. PROJEKTNII ZADATAK.....	12
4. REALIZACIJA PROJEKTOG ZADATKA.....	12
5. ZAKLJUČAK	13
6. LITERATURA.....	14
7. PRILOG KODOVI	15

1. UVOD

Spartan 3 generacija FPGA ploča nudi izbor od pet platformi, gde svaka omogućava jedinstven balans programabilne logike i načina povezivanja za programiranje jednostavnih aplikacija.

- **Spartan-3A-DSP**-(DSP optimizovan) idealan je za kreiranje jednostavnih aplikacija za obradu signala, kao što su vojni radio, nadzorne kamere, i razne primene u medicini,
- **Spartan-3AN**-najviše se koristi za sigurnosne aplikacije i ima veliku primenu u ugrađenim sistemima,
- **Spartan-3A**-optimizovan za ulazne i izlazne uređaje, koristi se za aplikacije koje se baziraju na radu sa memorijom i koje zahtevaju spektar različitih pristupa,
- **Spartan-3E**-logički optimizovan-koristi se za integrisanje logičkih kola, kontrolu ugrađenih sistema,
- **Spartan-3**-koristi se za visoko integrisane aplikacije za procesovanje podataka.

Spartan 3E familija FPGA ploča je zasnovana na strukturi uređaja ranije Spartan 3 familije, kojoj je značajnim poboljšanjima na polju ulazne i izlazne logike, smanjena cena po ulaznoj ćeliji. Nove karakteristike su unapredile performanse sistema i smanjile cenu konfiguracije. Ovakva Spartan 3E FPGA poboljšanja u kombinaciji sa 90 nanometarskom procesnom tehnologijom su omogućile bolju funkcionalnost, postavljajući nove standarde u proizvodnji programabilne logike.

Zbog niskih cena, Spartan-3E FPGA uređaji su idealni za programiranje širokog spektra korisničkih elektronskih aplikacija uz sam pristup ploči. Mogu da se koriste za povezivanje na internet, projektovanje kao i dodatak digitalne televizije.

Glavne karakteristike Spartan 3E familije FPGA uređaja:

3

- Niska cena, visoko-performansna rešenja za korisničke aplikacije.
- Pouzdana 90 nanometarska procesna tehnologija.
- Više-voltni, standardni SelectIO interfejs za pinove.
 - Do 376 I/O pinova, odnosno 156 različitih signalnih parova.
- LVCMOS, LVTTL, HSTL i SSTL jednosmerni signalni standardi..
- 3.3V, 2.5V, 1.8V, 1.5V i 1.2V signalni opseg
- Poboljšana Double Data Rate (DDR) podrška / DDR SDRAM podržava brzinu do 333 Mb/s.
- Efikasni multiplexeri i unapređena logika .
- IEEE JTAG port za debugovanje.
- Hijerarhijska arhitektura RAM memorije.
 - Do 648 Kbita brzih RAM blokova
 - Do 231 Kbita efikasne RAM memorije
- Do 8 digitalnih clock menadžera (DCM)
 - Sinteza frekvencije, množenje, deljenje
 - Fazno šiftovanje visoke rezolucije
 - Širok spektar frekvencije (od 5 Mhz do 300 Mhz)
- 8 globalnih CLOCK-ova plus 8 dodatnih na svakoj polovini uređaja
- Prostorno jeftin SPI serijski Flash Prom
- Kompletan Xilinx ISE i WebPack softver za potpunu manipulaciju nad uređajima ove vrste.

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

Tabela 1: Prikaz svih uređaja i njihovih karakteristika Spartan 3E familije

2. ARHITEKTURA

Nexys 2 (*Slika 1*) je kompletna, funkcionalna i veoma efikasna razvojna platforma zasnovana na Xilinx Spartan 3E FPGA porodici uređaja. Omogućava razvijanje raznovrsnih dizajnova digitalnih sistema, i što je veoma bitno, za to joj nisu potrebne nikakve dodatne komponente. Glavne karakteristike ove ploče su:

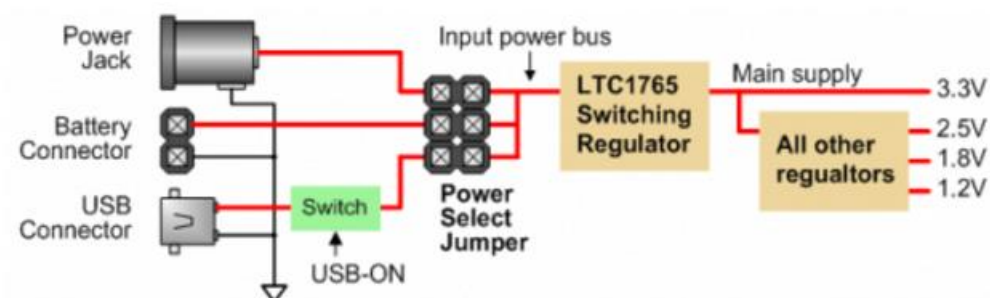
- Xilinx Spartan 3E FPGA 500k gate procesor
- 16 MB brze PSDRAM memorije
- 16 MB Intel Flash ROM memorije
- Razvija se pomoću Xilinx ISE/Web Pack softvera
- 50 Mhz oscilator, plus postolje za sekundarni oscilator
- Svi ulazno/izlazni (I/O) signali su zaštićeni od kratkog spoja i omogućavaju dugotrajnu funkcionalnost u bilo kom okruženju
- USB 2.0 port - može da se koristi kao izvor napajanja ploče , za konfiguraciju uređaja i veoma brzu razmenu podataka
- 8 LED dioda
- 4 1-cifarna sedmo-segmentna displeja
- 4 tastera
- 8 prekidača
- Ukupno 60 I/O konektora za razne dodatne komponente (brzi Hirose FX2 konektor sa 43 signala i 4 2x6 Pmod konektora, VGA konektor, RS232, PS/2 port)



Slika 1: Prikaz Nexys 2 FPGA ploče

2.1. IZVORI NAPAJANJA

Izvori napajanja Nexys 2 ploče (Tabela 2) koji se povezuju na ulazne konektore namenjene za napajanje ploče, mogu da budu USB kabl, 2.1 milimetarski zidni priključak ili baterija. "Power select" blok (Slika 2) na samoj ploči određuje koja će vrsta napajanja biti omogućena. Električno kolo za USB je pod naponom samo kada je USB kabl prključen, čim se USB kabl otkāči, kolo ostaje bez napona.



Slika 2: "Power select" blok

Nexys2 Power Supplies		
Supply	Device	Amps (max/typ)
3.3V main	IC6: LTC1765	3A/100mA
2.5V FPGA	IC7: LTC3417	1.4A/50mA
1.2V FPGA	IC7: LTC3417	1.4A/200mA
1.8V SRAM	IC8: LTC1844	150mA/90mA
3.3V USB	IC5: LTC1844	150mA/60mA

Tabela 2: Izvori napajanja za Nexys 2 ploču

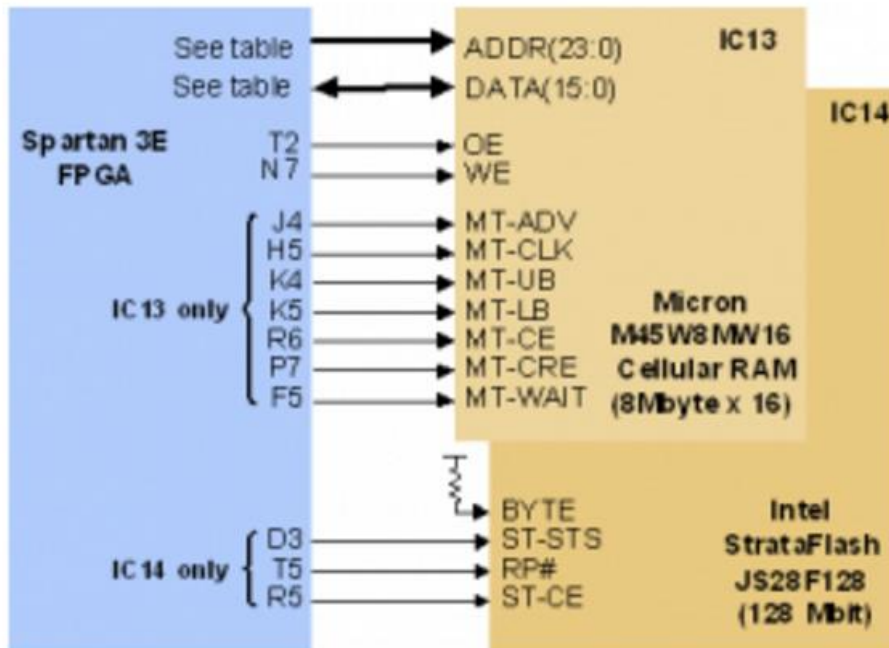
Konektor za izvor napajanja koristi 3.3 voltni regulator koji obezbeđuje struju za kompletnu ploču. Neki uređaji zahtevaju 2.5V, 1.8V, i 1.2V preko osnovna 3.3V, i takva dodatna napajanja su omogućena kreiranjem regulatora koji dovode svoje napajanje na glavno napajanje od 3.3V. Ovi regulatori ne samo da efikasno koriste snagu USB-a, već i omogućavaju da se Nexys 2 ploča pokreće samo preko baterija i to na duži vremenski period.

2.2. MEMORIJA

Nexys 2 platforma ima eksterne RAM i ROM uređaje. Eksterni RAM je 128 Mbitni Micron Cellular pseudo-static DRAM uređaj organizovan kao 8 Mbajta x 16 bitova. Može da se pokreće kao tipični asinhroni SRAM sa ciklusima upisivanja i čitanja podataka reda 70 nano-sekundi, ili kao sinhrona memorija sa 80 Mhz adresnom magistralom. Kada se koristi kao asinhroni SRAM, Cellular RAM automatski osvežava svoje unutrašnje DRAM nizove memorije, omogućavajući pojednostavljen dizajn memorije na FPGA platformi. Kada se koristi kao sinhroni SRAM, moguće je razmenjivanje podataka maksimalne brzine od 80 Mhz.

Eksterni ROM je 128Mbitni Intel StrataFlash uređaj, organizovan kao 8 Mbajta x 16 bitova. Sadrži 128 blokova koji mogu pojedinačno da se brišu, a podržava i 110 nano-sekundni ciklus čitanja podataka, sa 25 nano-sekundnim modom čitanja u okviru blokova. Ima interni 32- bajtni bafer koji može da se upisuje maksimalnom brzinom od 70 nano-sekundi po ciklusu, a može biti pretvoren i u Flash niz.

Oba uređaja (Slika 3) podržavaju 16- bitnu magistralu podataka i 24- bitnu adresnu magistralu.



Slika 3: Prikaz RAM i ROM memorije

2.3. ULAZNI/IZLAZNI UREĐAJI (LED diode, prekidači i tasteri)

Na Nexys 2 ploči se nalaze 4 tastera, 8 LED dioda i 8 prekidača.

Taster (*Slika 4*) funkcioniše tako što je u normalnom stanju neaktivan, a sve vreme dok je pod pritiskom poprima stanje jedinice. Međutim, zbog mehaničke implementacije, moguće je da taster prilikom prelaska sa logičke nule na logičku jedinicu, i obrnuto (pritiskom i puštanjem tastera), generiše nestabilan signal koji može negativno da utiče na izvršavanje programa. Debaunsiranje tastera je jedan od načina prevazilaženja ovog problema.



Slika 4: Prikaz 4 tastera na Nexys 2 ploči

Prekidači (*Slika 5*) su uređaji koji generišu konstantan signal. Signal ima vrednost logičke jedinice kada je prekidač pomeren na gore (slide up), a vrednost logičke nule kada je u normalnom položaju.

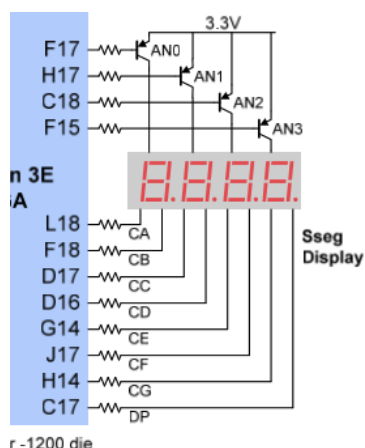


Slika 5: Prikaz 8 (slide) prekidača i 8 LED dioda

Osam LED dioda (*Slika 5*) su na ploči postavljene kao izlazne komponente. Anode LED dioda su povezane sa FPGA preko otpornika od 390 Ohm-a, tako da će izlaz diode da postane logička jedinica sa oko 3-4 mA struje. Deveta LED dioda prikazuje kada je ploča upaljena, dok deseta dioda označava FPGA status programiranja. Dioda 4-7 su dodeljene različitim pinovima kako bi se razlikovale među verzijama ploča (500 i 1200).

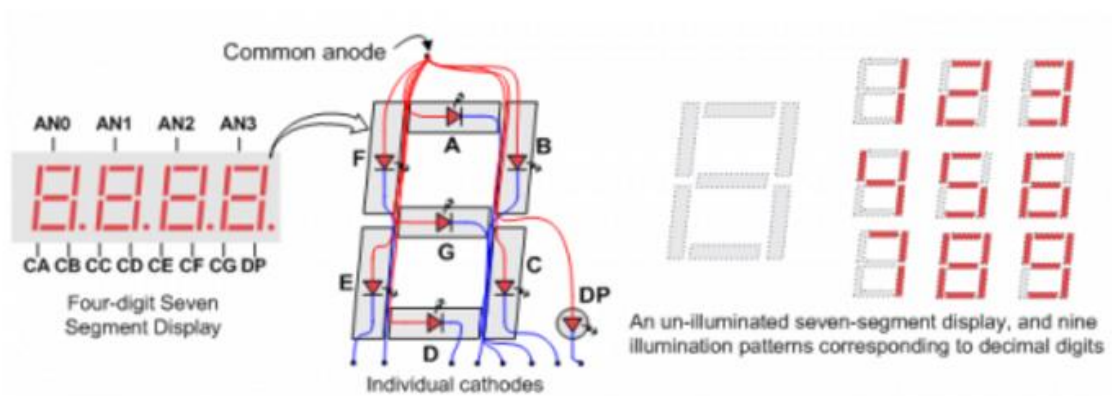
2.4. SEDMO-SEGMENTNI DISPLEJ

Nexys 2 ploča sadrži četiri 1-cifarna uobičajena sedmo-segmentna LED displeja. Svaki od 4 displeja je sastavljen od 7 segmenata poređanih kao na *Slici 6*, gde svaki segment predstavlja jednu LED diodu. Svaka od tih dioda može samostalno da se pali i gasi, tako da je moguće predstaviti svih 128 kombinacija na displeju, gde su određene diode upaljene i svetle, dok su sve ostale ugašene. Od svih 128 kombinacija, najčešće korišćene su one koje imaju veze sa predstavom decimalnih brojeva.



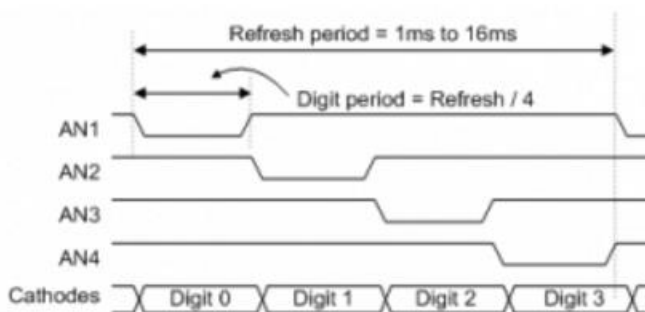
Slika 6: Prikaz šeme segmentnih displeja

Anode povezane na 7 LED dioda, su spojene u jednu "zajedničku anodu", dok su katode LED dioda odvojene. Signal zajedničke anode je sastavljen od 4 ulazna signala, gde svaki signal pali posebni segmentni displej. Katode na svakom od tih segmenata su spojene u kolo čvorova, označenih kao CA do CG (*Slika 7*). Signali ovih sedam katoda su omogućeni kao ulazi za svaki od 4 displeja. Ovakva šema povezanih signala pravi multipleksirani displej, gde su signali katoda zajednički za sve displeje ali mogu da se pale samo na jednom displeju i to na onom čija je anoda aktivna.



Slika 7: Prikaz rasporeda katoda na svakom displeju

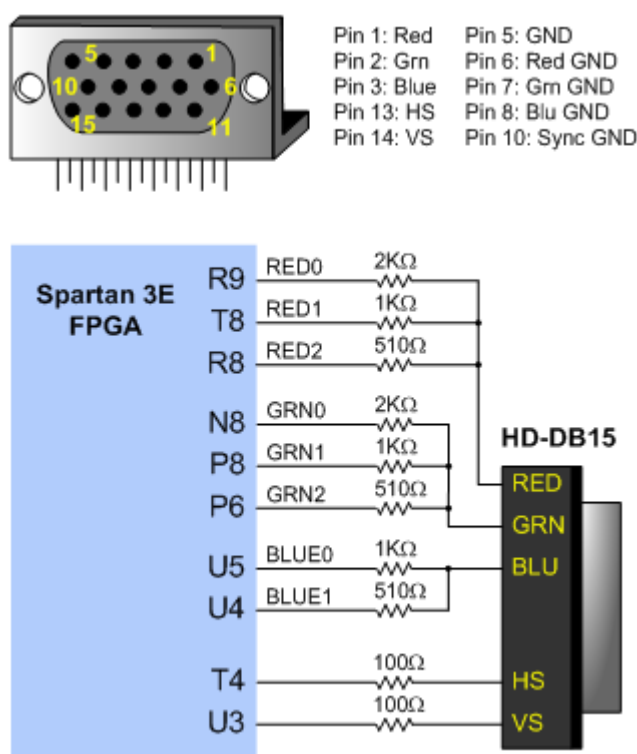
Kako bi bilo omogućeno da više od jednog displeja bude upaljeno u istom trenutku, potrebno je podesiti frekvenciju osvežavanja svakog displeja tako da se osvežavanje odvija dovoljno velikom brzinom da ljudsko oko nema utisak da displeji rade naizmenično, već konstantno. Potrebna je frekvencija od 1Khz do 60Hz, gde bi se displeji naizmenično osvežavali na svakih 1-16 milisekundi (Slika 8).



Slika 8: Naizmenično paljenje segmentnih displeja

2.5. VGA PORT

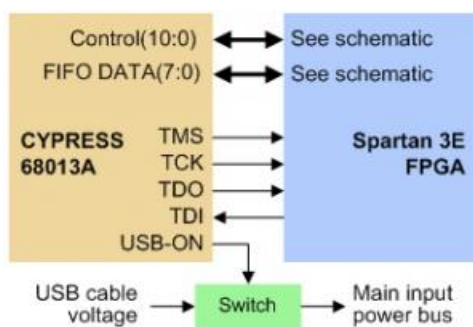
Nexys 2 ploča koristi 10 FPGA signala za kreiranje VGA porta (Slika 9) sa 8-bitnim kolor i dva standardna sinhronizujuća signala (HS – Horizontalno sinhronizovanje, i VS – Vertikalno sinhronizovanje). Kolor signali koriste kola razdelnika otpora koja rade zajedno sa 75-Ohm-nim otpornikom VGA displeja za kreiranje osmosignalnih nivoa na crvenim i zelenim VGA signalima, i četiri na plavim (ljudsko oko je manje osetljivo na plave nivoe). Ovo kolo, pokazano na Slici 9, proizvodi video kolor signale koji postupaju u jednakim inkrementovanjima između 0V (isključeno) i 0.7V (uključeno). Koristeći ovo kolo, 256 različitih boja može biti prikazano, jedan za svaki pojedinačni 8-bitni obrazac. Kolo video kontrolera mora biti kreirano u FPGA da bi pokretalo sinhronizaciju i kolor signale sa tačnim tajmingom da bi proizveo displej sistem koji radi.



Slika 9: Prikaz VGA Porta

2.6. USB PORT

Nexys 2 uključuje i brzi USB 2.0 port (*Slika 10*) baziran na Cypress CY7C68013A USB kontroleru. USB port se može koristiti za programiranje na ploči Xilinx uređaja, za izvođenje korisnik-podatak transfera brzine do 38Mbyte/sec, i za obezbeđivanje napona na ploči. Programiranje je postignuto sa Digilentovim besplatnim Adept Suite softverom. Transferi korisničkih podataka mogu takođe biti postignuti korišćenjem Adept softvera, ili bilo koji drugi softver može biti napisan korišćenjem Digilentovih javnih aplikativnih programskih interfejsa (API) da se ostvari Nexys 2 USB konekcija. Informacije za korišćenje Adepta i/ili javnih API za transfer podataka se mogu naći na Digilentovom veb sajtu.

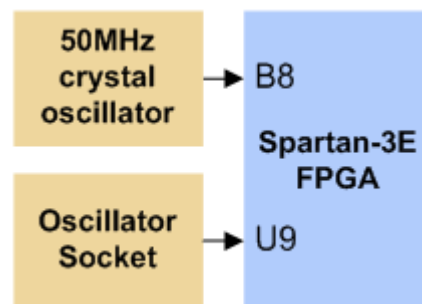


Slika 10: USB Port

USB port može takođe da obezbedi napon na Nexys2 ploči ako je džemper za selektovanje napona postavljen na „USB“. Kada je prvi put konektovana na USB host, Nexys2 ploča zahteva 500mA, i tada aktivira switch tranzistor da konektuje USB naponski kabl na glavni naponski ulaz.

2.7. CLOCK

Nexys 2 ploča uključuje i 50Mhz oscilator (*Slika 11*) i postolje za drugi oscilator. Clock signali oscilatora konektuju se na globalne clock ulazne pinove na FPGA tako da oni mogu pokrenuti sinhronizujuće blokove na FPGA. Clock sinhronizatori obezbeđuju clock menadžment mogućnosti koje uključuju dupliranje i kvadriranje ulazne frekvencije, deljenje ulazne frekvencije sa bilo kojim višestrukim intedžerima, i precizno definisanje veza faza i kašnjenja između različitih clock signala.



Slika 11: Prikaz Clock-a na Nexys 2 platformi

3. PROJEKTNII ZADATAK

Ideja za projektni zadatak je realna simulacija broja slobodnih parking mesta na nekom parkingu, gde bi se nakon svakog ulaska (izlaska) određenog broja vozila na (sa) parkinga, broj slobodnih mesta inkrementovao (dekrementovao), i pritiskom na taster vraćao na početni broj slobodnih mesta (ukoliko je parking prazan). Ukoliko je parking pun, na displeju bi pisala neka reč koja bi označavala da na parkingu više nema slobodnih mesta.

4. REALIZACIJA PROJEKTOG ZADATKA

Ideja za realizaciju nam je prvobitno bila da se samo korišćenjem različitih tastera tj. button-a resetuje ili inkrementira ili dekrementira broj parking mesta, u zavisnosti od toga šta je potrebno. Međutim zbog sintakse `always` petlje i `if-else` slučajeva to nikako nije bilo moguće uraditi, jer `sensitivity` lista u `always` petlji u Xilinx ISE Design Suite 14.7 ne radi kako treba za više od 2 uslova, tj. parametra, pa smo projekat realizovali korišćenjem dva switcha (prekidača) „SW0“, „SW1“ i jednog tastera „BTN2“, i to na način opisan u daljem tekstu. Kada neki auto uđe na parking broj parking mesta na displeju se dekrementuje podizanjem prvog prekidača SW0 na 1 i pritiskom na taster, a kada auto izađe sa parkinga broj parking mesta se povećava za 1, odnosno inkrementuje, pritiskom na taster, dok prekidač SW0 ima vrednost nula, tj. isključen je. Na displeju se ispisuje reč **FULL** kada je parking pun, tj. kada broj slobodnih parking mesta dođe na nulu. Kada dođe kraj ili početak radnog vremena parkinga, broj parking mesta treba da se resetuje na maksimalan broj slobodnih parking mesta, u našem slučaju to je 50. To smo postigli tako što smo realizovali da se podizanjem drugog prekidača SW1 tj. postavljanjem na jedinicu i pritiskom tastera broj parking mesta resetuje na 50, bez obzira da li je prvi prekidač SW0 uključen ili isključen. To će biti prikazano na displeju nakon što drugi prekidač SW1 vratimo u svoj položaj, tj. isključimo ga (vratimo na nulu). Postavili smo i granice tako da broj parking mesta nikada neće otići preko 50 ili ispod 0 (FULL).

5. ZAKLJUČAK

U ovom projektu je prikazana realna simulacija slobodnih parking mesta na nekom parkingu. Projekat je takođe primenljiv u praksi gde bi se umesto tastera i prekidača mogao koristiti senzor koji bi detektovao pomeranje rampe, ili kao implementacija u nekom programu koji bi smanjivao (povećavao) broj parking mesta na displeju prilikom svakog plaćenog (izdatog) računa.

6. LITERATURA

1. <https://reference.digilentinc.com/reference/programmable-logic/nexys-2/reference-manual>
2. <http://store.digilentinc.com/nexys-2-spartan-3e-fpga-trainer-board-retired-see-nexys-4-ddr/>
3. <https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>
4. http://hamsterworks.co.nz/mediawiki/index.php/Digilent_Nexys2
5. <https://www.wikipedia.org/>

7. PRILOG KODOVI

15

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    15:55:17 01/08/2017
// Design Name:
// Module Name:    delay_projekat
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module delay_projekat(
    input button_in,
    input button2,
    input clk,
    input n_reset,
    output a,
    output b,
    output c,
    output d,
    output e,
    output f,
    output g,
    output dp,
    output [3:0]an
);

    reg [3:0]first; // Registar za prvi AN
    reg [3:0]second; // Registar za drugi AN
    reg [3:0]third; // Registar za treci AN
    reg [3:0]fourth; // Registar za cetvrti AN

    wire b1, b2, b3;

    assign b1 = button_in; // Assign-ovanje inputa na promenljive
    assign b2 = button2;
    assign b3 = n_reset;
```

```
always @ (posedge b1)
begin
if(b3 & b1)begin    // Vraca broj parking mesta na maksimum
    first<=4'd0;
    second<=4'd5;
    third<=4'd0;
    fourth<=4'd0;
end
else if (b1 & ~b2) begin
    if(first==4'd0 & second==4'd5)begin    // Ako broj parking mesta dostigne 50 ostani na 50 (vise od toga se ne moze inkrementirati)
        first<=4'd0;
        second<=4'd5;
        third<=4'd0;
        fourth<=4'd0;
    end
    else if (first==4'd92 & second==4'd92)begin    // Ako stoji FULL i hocemo da inkrementujemo postavi broj slobodnih parking mesta na 1
        first<=4'd1;
        second<=4'd0;
        third<=4'd0;
        fourth<=4'd0;
    end
    else if (first==4'd9) begin    // x9 dostignuto
        first <= 0;    // Postavi prvi AN na 0
        second <= second + 1; // Drugi AN inkrementuj
    end
    else
        first <= first + 1;    // U suprotnom prvi AN inkrementuj
end

else if (b1 & b2)begin
    if(first==4'd0 & second==4'd0)begin
        first<=4'd0;
        second<=4'd0;
    end
    else if (first==4'd0) begin    // x0 dostignuto
        first <= 9;    // Postavi prvi AN na 9
        second <= second - 1; // Dekrementuj drugi AN
    end
    else
        first <= first - 1;    // U suprotnom prvi AN dekrementuj
end
end
```

```
// Postavljanje frekvencije kojom se segmentni displeji pale naizmenicno

localparam N = 18;      // Ovim se postize frekvencija oko 800Hz koja je dovoljna da covek ima utisak da segmenti ne prestaju da svetle

reg [N-1:0]count;

always @ (posedge clk or posedge n_reset)
begin

    if (n_reset)
        count <= 0;
    else
        count <= count + 1;
    end

reg [6:0]sseg;
reg [3:0]an_temp;      // Registar za AN-ove

always @ (*)
begin

    if(first==4'd0 & second==4'd0)
    begin
        case(count[N-1:N-2])

            2'b00 :
            begin
                sseg=4'd92 ;           // L
                an_temp = 4'b1110;
            end

            2'b01:
            begin
                sseg=4'd92 ;           // L
                an_temp = 4'b1101;
            end

            2'b10:
            begin
                sseg=4'd91;           // U
                an_temp = 4'b1011;
            end

            2'b11:
            begin
                sseg=4'd90;           // F
                an_temp = 4'b0111;
            end
        endcase
    end

    else
        case(count[N-1:N-2])

            2'b00 :
            begin
                sseg = first;
                an_temp = 4'b1110;
            end

            2'b01:
            begin
                sseg = second;
                an_temp = 4'b1101;
            end

            2'b10:
            begin
                sseg = third ;
                an_temp = 4'b1011;
            end

            2'b11:
            begin
                sseg = fourth;
                an_temp = 4'b0111;
            end
        endcase
    end
end
```

```
assign an = an_temp;

reg [6:0] sseg_temp;
always @ (*)
begin
    case (sseg)
        4'd0 : sseg_temp = 7'b1000000; // 0
        4'd1 : sseg_temp = 7'b1111001; // 1
        4'd2 : sseg_temp = 7'b0100100; // 2
        4'd3 : sseg_temp = 7'b0110000; // 3
        4'd4 : sseg_temp = 7'b0011001; // 4
        4'd5 : sseg_temp = 7'b0010010; // 5
        4'd6 : sseg_temp = 7'b0000010; // 6
        4'd7 : sseg_temp = 7'b1111000; // 7
        4'd8 : sseg_temp = 7'b0000000; // 8
        4'd9 : sseg_temp = 7'b0010000; // 9
        4'd90 : sseg_temp = 7'b0001110; // F
        4'd91 : sseg_temp = 7'b1000001; // U
        4'd92 : sseg_temp = 7'b1000111; // L
        default : sseg_temp = 7'b1000000; // 0
    endcase
end
assign {g, f, e, d, c, b, a} = sseg_temp;
assign dp = 1'b1; // Ne koristimo decimale pa nam ni tacka/e nisu potrebne

endmodule
```