

Univerzitet u Kragujevcu  
Fakultet inženjerskih nauka



Seminarski rad  
Osnove računarske tehnike 2

Tema:  
Košarkaški semafor na Spartan 3E FPGA ploči

Student:  
Aleksa Veličković

Predmetni profesor:  
Dr. Aleksandar Peulić

Kragujevac, februar 2017.

**Sadržaj:**

1. UVOD.....	2
2. ARHITEKTURA.....	3
2.1. Osnovne karakteristike ploče Nexus2:.....	4
2.2. Izvori napajanja .....	4
2.3. Klok.....	5
2.4. U/I portovi.....	5
2.5. USB port .....	7
2.6. PS/2 port .....	7
2.7. VGA port.....	7
2.8. Serijski port .....	8
2.9. Memorija.....	8
2.10. Periferni konektori .....	9
3. PROJEKTNII ZADATAK.....	10
4. REALIZACIJA PROJEKTOG ZADATKA.....	10
5. ZAKLJUČAK .....	13
6. LITERATURA .....	13
7. PRILOG KODOVI .....	14

## 1. UVOD

U ovom radu će biti opisana realizacija košarkaškog semafora, korišćenjem Nexus 2 razvojnog sistema baziranog na Xilinx Spartan 3E FPGA.

Semafor odbrojava i zaustavlja vreme utakmice, prikazuje i omogućava promenu rezultata. Naravno, ovakva ideja, uz malo izmena, može naći svoju primenu u različitim oblastima; uz dodatne ulazno/izlazne portove (tastere, displeje...) koncept bi se lako proširio, zavisno od zahteva koje imamo pred sobom, na druge sportove i takmičenja. Opisana je arhitektura korišćenog FPGA uređaja i objašnjeno njegovo programiranje u Verilogu (jezik za opis hardvera).

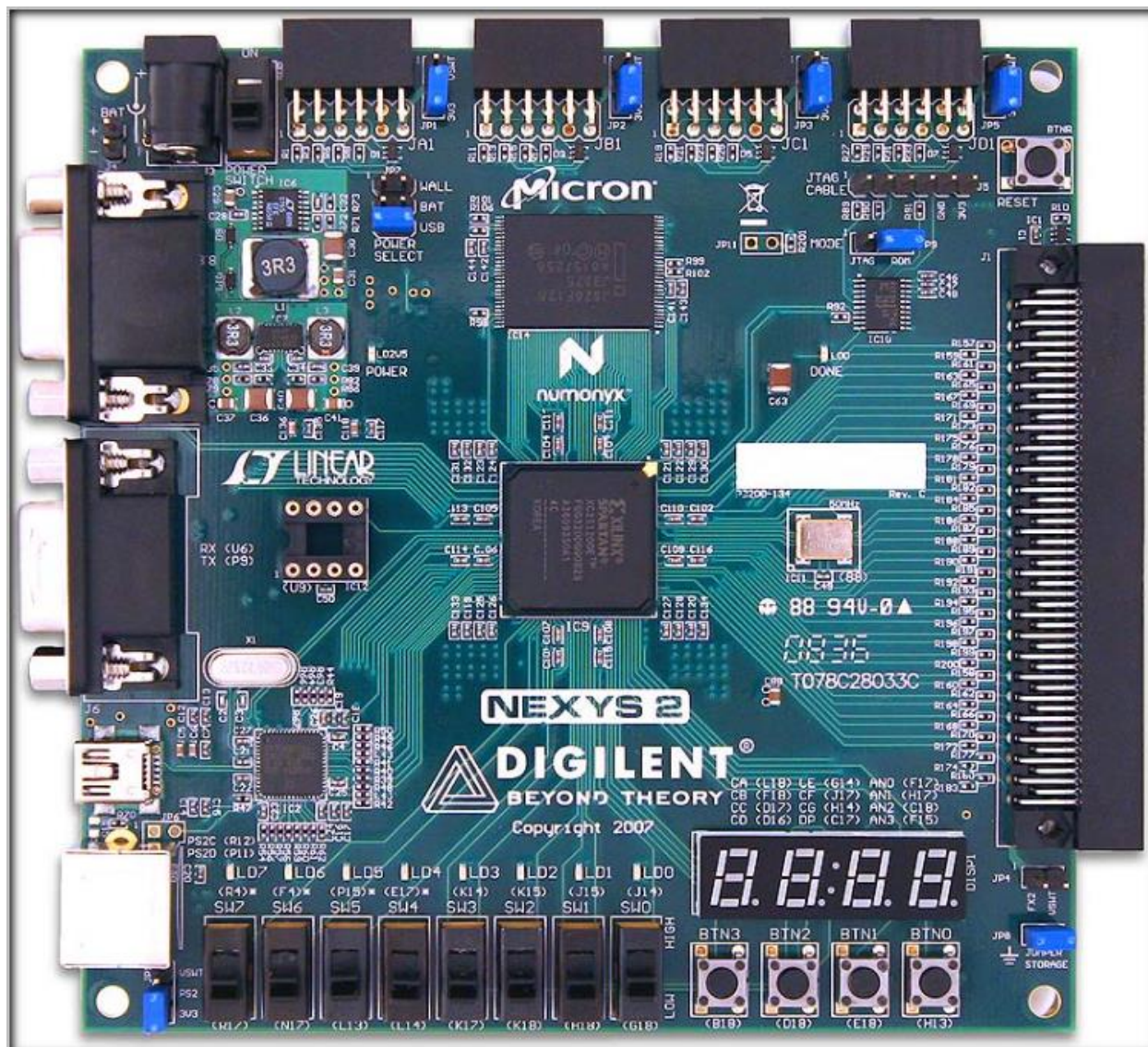
Spartan 3 generacija FPGA ploča nudi izbor od pet platformi:

- **Spartan-3A-DSP** - idealan za primene u obradi signala, kamere...
- **Spartan-3AN** - koristi se u bezbedonosne svrhe.
- **Spartan-3A** - pogodan kada je broj U/I portova važniji od gustine logike; idealan za primene u povezivanju, a pritom pristojnom obradom.
- **Spartan-3E** - suprotno od prethodnog, je pogodan kada je gustina logike važnija od broja U/I; kada postoji značajna obrada, procesiranje, i nekoliko interfejsa.
- **Spartan-3** - koristi se kada postoji potreba, kako za U/I portovima, tako i za gustom logikom.

Spartan-3E FPGA je idealan za širok spektar primena, uključujući i povezivanje na internet, kao i oprema digitalne televizije... Za programiranje i manipulaciju uređaja se koristi Xilinx ISE softver. Potrebe društva za ovakvom vrstom proizvoda su česte; košarka je najtrofejniji srpski sport, pa je korišćenje ovakvog proizvoda široko rasprostranjeno.

## 2. ARHITEKTURA

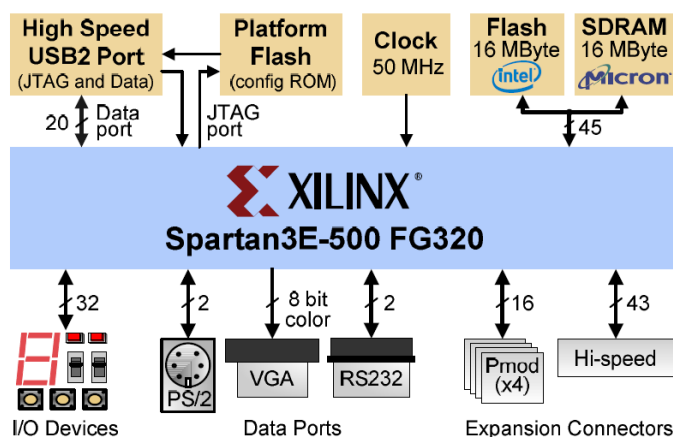
Ploča Nexus2 je razvojni sistem baziran na Xilinx Spartan 3E FPGA. Njen USB2 port, 16Mbajta RAM i ROM i nekoliko U/I uređaja i portova čine je idealnom platformom za različite digitalne sisteme. USB2 port obezbeđuje napajanje i interfejs za programiranje. Četiri 12-pinska konektora perifernih uređaja mogu da prime do osam uređaja, kao A/D i D/A konverzija, audio kola ili senzora. Svi konektori su zaštićeni od kratkog spoja, što obezbeđuje rad u bilo kom okruženju.



Slika 1. Razvojna ploča Nexus2

## 2.1. Osnovne karakteristike ploče Nexus2:

- Procesor 500K-gate Xilinx Spartan 3E FPGA
- USB2 za programiranje, preko besplatnog AdeptSuite softvera.
- Napajanje preko USB
- 16MB Micron RAM memorije i 16MB Intel StrataFlash ROM memorije
- Oscilator takta 50MHz i slot za dodatni oscilator
- 60 FPGA U/I konektora (brzi Hirose FX2 konektor sa 43 signala i 4 2x6 Pmod konektora, VGA konektor, RS232, PS/2 port)
- 8 LED dioda, 4 sedmosegmentna displeja, 4 tastera, 8 prekidača



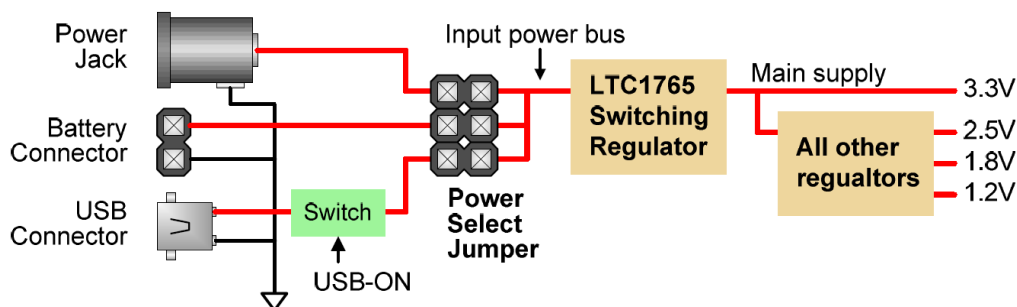
Slika 2. Blok šema ploče Nexus2

## 2.2. Izvori napajanja

Napajanje Nexus2 ploče može da se izvrši USB kablom, od 5VDC-15VDC, centralno pozitivnim, od 2.1mm; ili baterijom. Džamperom se selektuje izvor napajanja.

Ulazna magistrala napajanja provodi struju napona 3.3V. Neki uređaji zahtevaju dodatnih 2.5V, 1.8V i 1.2V, ove dodatke stvaraju regulatori iz glavnog izvora od 3.3V.

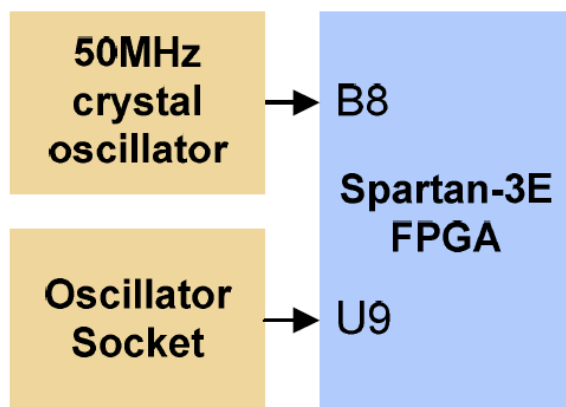
Ploča Nexus2 može da se poveže na napajanje ili da napaja druge uređaje preko Pmod konektora (perifernih modula) ili preko 100-pinskog konektora.



Slika 3. Blok šema sistema za napajanje ploče Nexus2

### 2.3. Klok

Ploča Nexus2 uključuje oscilator takta 50MHz i slot za dodatni oscilator. Signali kloka oscilatora su povezani sa globalnim ulaznim pinovima na FPGA, da bi mogli da upravljaju blokovima za sintetizovanje kloka dostupnim u FPGA, koji mogu da dupliraju ulaznu frekvenciju, dele ulaznu frekvenciju bilo kojim celim brojem i da definišu fazu ili kašnjenje između različitih signala kloka.



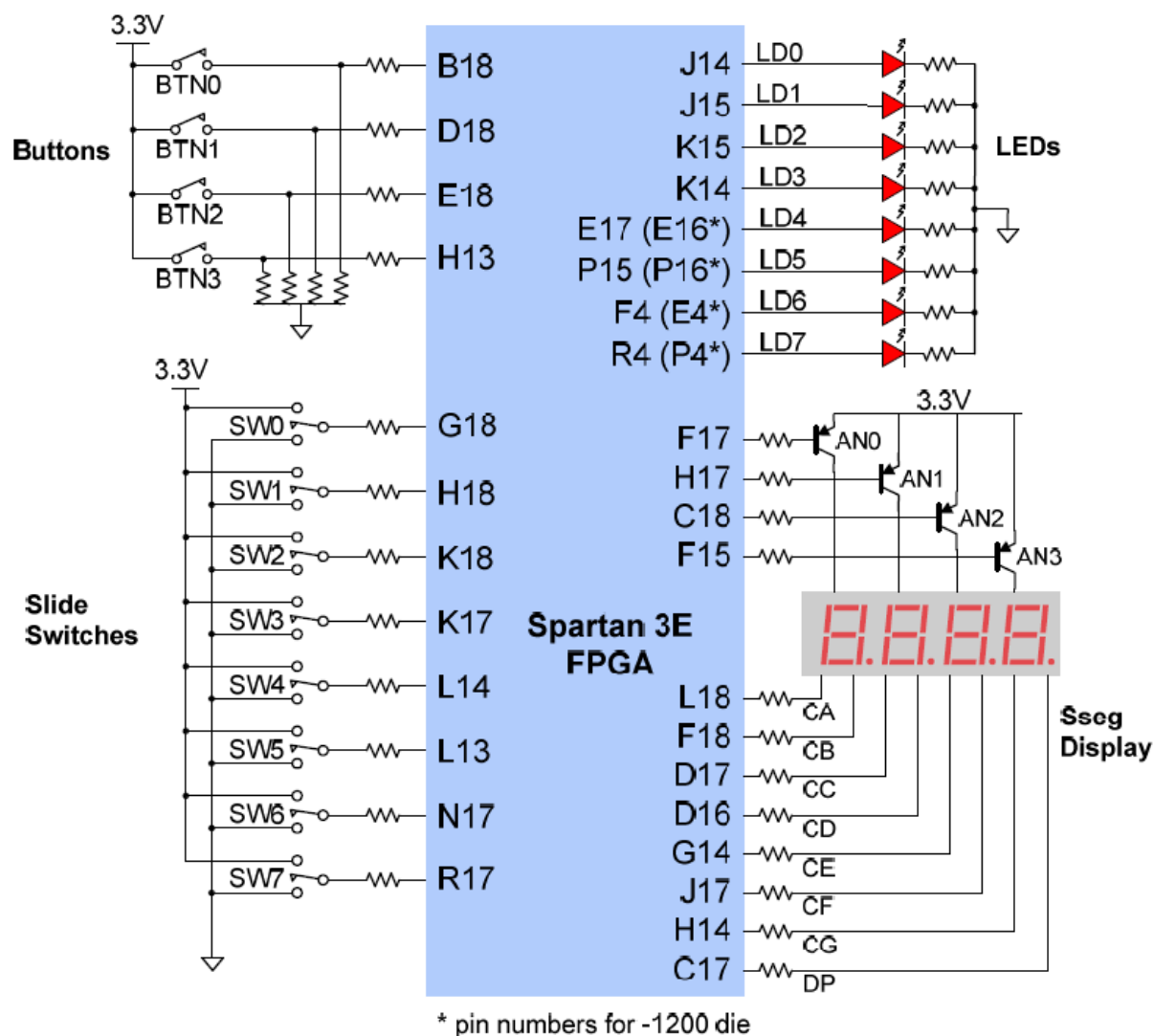
Slika 4. Povezivanje oscilatora na razvojnu ploču Nexus2

### 2.4. U/I portovi

Ploča uključuje nekoliko ulaznih uređaja, izlaznih uređaja i portove za podatke, obezbeđujući da realizaciju mnogih projekata i bez dodataka.

Postoje četiri tastera i osam prekidača kao ulazne komponente. Tasteri su na logičkoj nuli, osim ako nisu pritisnuti, kada prelaze u stanje logičke jedinice.





Slika 5. U/I jedinice na ploči Nexus2

Prekidači imaju konstantnu vrednost zavisno od trenutne pozicije. Tasteri i prekidači koriste otpornike kao zaštitu od kratkog spoja (kratak spoj bi se desio ako bi taster ili prekidač bili definisani kao izlazni).

Postoji osam LED dioda. LED anodama upravlja FPGA kroz otpornike od 390-ohm, tako da će logična jedinica da ih upali strujom od 3-4mA. Deveta dioda pokazuje da li je upaljena ploča, a deseta da li je isprogramirana. Ploča sadrži četiri sedmosegmentna LED displeja. Svaka od četiri cifre se sastoji od sedam segmenata, a svaki ima po diodu. Segmenti mogu biti pojedinačno paljeni, tako da bilo koja od 128 kombinacija može da bude prikazana. Anode sedam LED-a su vezane u jednu, zajedničku anodu, ali LED katode ostaju razdvojene. Zajednički signali anoda su dostupni kao četiri "enable" ulazna signala za četvorocifreni displej. Katode identičnih segmenata su povezani sa sedam čvorova u kolu označenim kao CA do CG.

Signali katoda su zajednički za sve cifre, ali mogu da osvetle samo segmente one cifre kojoj je dodeljena vrednost anode. Kako prikazati 4-cifreni broj? Smenjivati svaki displej i njemu odgovarajuću vrednost, tako da je dovoljno brzo da ljudsko oko ne može da detektuje promenu, a dovoljno sporo da se cifra zadrži dovoljno dugo na displeju-prihvatljiva frekvencija je 1KHz.

## 2.5. USB port

Ploča Nexus2 uključuje brzi USB2 port, baziran na USB kontroleru Cypress CY7C68013A.

USB port može biti korišćen za programiranje ploče, brzinom 38Mbajta/sec i da obezbedi napajanje.

Za programiranje se koristi besplatan *Digilent* softver *AdeptSuite*. Izvor napajanja preko USB-a se podešava pomoću džampera. Ploča obično vuče struju od oko 300mA iz USB-a, ali treba voditi računa da se, pri korišćenju spoljnih uređaja, ne poveća potrošnja preko 500mA.

Kada je ploča povezana, ona zahteva struju od 500mA nakon čega se preko tranzistora povezuje USB sa magistralom za napajanje.

## 2.6. PS/2 port

Šestopinski mini-Din konektor može da prihvati PS/2 miš ili tastaturu. Većina PS/2 uređaja može da radi sa 3.3V napajanja, ali stariji uređaji zahtevaju napajanje od 5VDC.

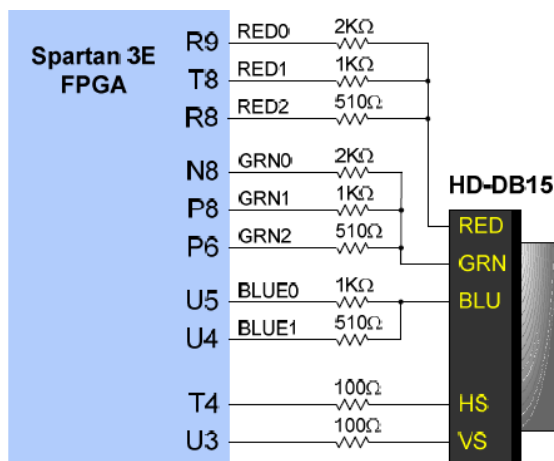
3-pinski džamper na Nexus2 ploči, odmah pored ovog porta selektuje da li regulisani napon od 3.3V ide na PS/2 port ili ide sa glavne, ulazne magistrale napajanja.

## 2.7. VGA port

Ploča koristi 10 FPGA signala za VGA port, 8-bitni signali boja i dva standardna signala.

Signali boja koriste kola razdelnika otpora koja rade vezano sa otpornikom od 75-ohm VGA displeja da bi stvorili osam nivoa signala na crvene i zelene VGA signale i četiri na plave (ljudsko oko je manje osetljivo na plave nivoe). Kolo sa slike stvara video signale boja u podjednakim rasponima između 0V (isključeno) i 0.7V (uključeno).





Slika 6. VGA port na ploči Nexus2

Koristeći ovo kolo, 256 različitih boja može biti prikazano, gde svaka boja ima jedinstvenu 8-bitnu kombinaciju. Mora biti napravljeno kolo video kontrolera u FPGA, da bi sinhronizovalo i upravljalo signalima boja sa tačnim tajmingom, jer u suprotnom ne bi radio sistem displeja.

## 2.8. Serijski port

Ploča sadrži dvožični serijski port baziran na naponskom konverteru ST *Microelectronics* ST3232.

ST3232 konvertuje signale koje koristi RS-232 komunikacija (-12V do -3V za logičku "1" i 12V do 3V za logičku "0") u signale od 3.3V koje koristi FPGA. Pošto su samo dva signala povezana (RXD i TXD), kontroler FPGA serijskog porta može samo da koristi softverske protokole (XON/XOF).

## 2.9. Memorija

Ploča ima spoljašnje RAM i ROM uređaje. Spoljašnji RAM je 128Mbitni Micron M45W8MW16 ćelijski RAM pseudo-static DRAM uređaj organizovan kao 8Mbajta x 16bitova. Može da funkcioniše kao uobičajeni asinhroni SRAM sa ciklusom čitanja i pisanja od 70ns ili kao sinhrona memorija sa 80MHz magistralom. Spoljašnji ROM je 128Mbitni Intel TE28F128J3D75-110 StrataFlash uređaj organizovan kao 8Mbajta x 16bitova. Unutar sebe sadrži 128 blokova koji mogu biti pojedinačno izbrisani i podržava 110ns cikluse čitanja, sa 25ns-im modom čitanja unutar blokova. Ima unutrašnji 32-bajtni bafer za pisanje, koji može biti ispisan 70ns-im ciklusima i 32-bitni bafer može biti prebačen u Flash niz za 218us (uobičajeno).

Oba uređaja dele 16-bitnu magistralu podataka i 24-bitnu adresnu magistralu. Ćelijski RAM adresiran bajtom, koristeći više-bajtne i niže-bajtne signale (MT-UB i MT-LB), dok je StrataFlash samo za 16bajtnje operacije.

Signali OE ("output enable") i WE ("write enable") dele oba uređaja, ali svaki uređaj ima posebne "chip enable" signal. Dodatno, ćelijski RAM ima klok (MT-CLK), čekanje (MT-WAIT), validnost adrese (MT-ADV) i uključivanje kontrolnog registra (MT\_CRE) i svi ovi signali su dostupni FPGA za korišćenje sa sinhronim transferima. StrataFlash ima signale Reset (RP#) i status (STS) unutar FPGA.

## **2.10. Periferni konektori**

Nexus2 ploča obezbeđuje četiri dvoredna 6-pinska Pmod konektora, koja zajedno mogu da prime do 8 perifernih modula.

Postoje četiri 12-pinska konektora i svaki ima 8 signala podataka, dva GND (zemlja) pina i dva Vdd pina.

Svi pinovi imaju otpornike kao zaštitu od kratkog spoja i ESD zaštitne diode. Blok džampera pored svakog konektora perifernog modula može da poveže Vdd Pmod-a na napajanje od 3.3V ili na ulaznu magistralu napajanja (VU).

Ako je džamper postavljen na VU, a USB napajanje ide kroz glavnu magistralu napajanja, treba obratiti pažnju da Pmod ne uzima više od 200mA. Dalje, ako je džamper podešen na VU, naponski izvor povezan na Pmod može da upravlja glavnom magistralom napajanja ploče, tako da bi trebalo da se izbegne sudaranje različitih izvora napajanja.

### 3. PROJEKTI ZADATAK

Projekat se zasniva na realizaciji košarkaškog semafora. Koriste se četiri tastera, tri prekidača i sva četiri sedmosegmentna displeja. Zavisno od stanja jednog od prekidača na displeju će biti prikazano trenutno vreme utakmice ili rezultat, tj. broj poena obe ekipe na po dva displeja.

Vreme je moguće zaustavljati pritiskom tastera, a rezultat se povećava za 1,2 ili 3 poena koristeći preostala tri tastera i ta promena, zavisno od stanja drugog prekidača, se odnosi na jednu ili drugu ekipu. Takođe, ako je došlo do greške pri dodeli poena, prebacivanjem trećeg prekidača u stanje logičke jedinice se mogu oduzeti dodeljeni poeni, ponovo koristeći pomenuta tri tastera.

### 4. REALIZACIJA PROJEKTOG ZADATKA

Projekat je realizovan korišćenjem tri prekidača "p0", "p1", i "p2", koji su povezani na pinove G18, H18 i K18, redom. Korišćena su četiri tastera "b0", "b1", "b2" i "b3", koji su povezani na pinove B18, D18, E18 i H13, redom. Četiri cifre displeja realizovane su četvorobitnom promenljivom "an", vezanom za anode F15, C18, H17 i F17, od viših ka nižim bitovima, redom. Za sedam segmenata displeja postoji sedam izlaza "a", "b", "c", "d", "e", "f" i "g" vezanih redom na L18, F18, D17, D16, G14, J17, H14. Izlazna promenljiva "dp" je vezana za pin C17, koji predstavlja decimalnu tačku, ali shodno tome da nije korišćena u projektu, dodeljena joj je konstantna vrednost 1 (invertovane su vrednosti, 0-uključen).

Od svakog ulaznog porta je potrebno napraviti "žicu", tj promenljivu tipa "wire", pa je na primer promenljivoj "p\_0", tipa "wire", konstantno dodeljena vrednost prekidača "p0", a analogno su pravljene i promenljive za ostale ulazne portove. Unutar programa figurira promenljiva za sedam segmenta "s\_seg", tipa "reg", čijih su sedam bitova, od višeg ka nižim, konstantno dodeljeni izlazima "g", "f", "e", ... Takođe, izlazu "an" je dodeljena vrednost četvorobitne promenljive "cifra", tipa "reg", koja je u programu zadužena, da svakim taktom menja cifru koja je upaljena, a tada se i dodeljuje adekvatna vrednost za sedam segmenata te cifre. Broj koji se trenutno ispisuje na cifru sedmosegmentnog je sadržan u četvorobitnoj promenljivoj "displej", tipa "reg", i kasnije se zavisno od vrednosti pale određeni segmenti displeja, tj. "s\_seg" uzima odedjenu vrednost. I na kraju, ulazna promenljiva "clk" je vezana za oscilator ploče na pinu B8.

U projektu postoji podmodul "bcd", to je u stvari konvertor binarnog broja u BCD oblik i koristi se za prikaz decimalnih brojeva na displeju. Radi metodom "shift and add 3", tj. šiftuje se ceo broj za jedan bit u levo, gde cifre iz kolone početnog broja prelaze u bcd kolone, tj. kolone jedinica, desetice itd., i ako je broj u jednoj od bcd kolona 5 ili veći dodaje mu se broj 3. Postupak šiftovanja se ponavlja onoliko puta, koliko binarni broj ima bitova, ne bi li svi bitovi prešli u bcd kolone, u našem slučaju šiftuje se osam puta, jer je promenljiva "broj" osmobitna. Modul kao ulazne promenljive ima "clk" i sam "broj" koji treba da se konvertuje, a izlazne su "jedinice" i "desetice", čije će vrednosti u glavnom modulu biti prosleđene na promenljivu "displej". Šesnaestobitna promenljiva "shift" predstavlja skup četvorobitnih kolona, 2x4bita za jedinice i desetice i još toliko za dve heksa cifre koliko ima "broj"; "shift" je tipa "reg" i pamti vrednosti kolona tokom ciklusa. Promenljiva "i" predstavlja broj izvršenih iteracija tokom ciklusa.

Operation	Hundreds	Tens	Units	Binary	
HEX				F	F
Start				1 1 1 1	1 1 1 1
Shift 1			1	1 1 1 1	1 1 1 1
Shift 2			1 1	1 1 1 1	1 1
Shift 3			1 1 1	1 1 1 1	1
Add 3			1 0 1 0	1 1 1 1	1
Shift 4		1	0 1 0 1	1 1 1 1	
Add 3		1	1 0 0 0	1 1 1 1	
Shift 5		1 1	0 0 0 1	1 1 1	
Shift 6		1 1 0	0 0 1 1	1 1	
Add 3		1 0 0 1	0 0 1 1	1 1	
Shift 7	1	0 0 1 0	0 1 1 1	1	
Add 3	1	0 0 1 0	1 0 1 0	1	
Shift 8	1 0	0 1 0 1	0 1 0 1		
BCD	2	5	5		

Slika 7. BCD kod

U glavnom modulu postoje i 8-bitne promenljive: "vremeA", predstavlja broj minuta utakmice, "vremeB" su sekunde i "rezula" je broj poena ekipe A, "rezulB" je broj poena ekipe B. Svaka od pomenutih je pozvana unutar "bcd" podmodula i ima odgovarajući izlaz iz njega, pa na primer "vremeA" kao izlaz iz bcd modula ima promenljive tipa "wire", "vremeAjedinice" i "vremeAdesetice", analogno važi i za ostale promenljive. Postoje još četiri promenljive tipa "reg", koje služe za tastere, tj. u njima se beleži da li je stisnuti taster pušten; svaka za određeni taster: stanjeB0, stanjeB1, stanjeB2 i stanjeB3, analogno.

Da bi se realizovao prikaz na displeju i tačno odbrojavanje, potrebno je precizno usporiti ulazni "clk" frekvencije 50MHz. Za prikazivanje na displeju koristi se "clkD", klok displeja, frekvencije 1KHz, a za njegovo usporavanje je korišćen "brojacD", brojač displeja, koji se svakim taktom povećava za jedan, dok ne stigne do 25000, kada se invertuje "clkD". Analogno, za odbrojavanje sekundi utakmice je potreban "clkV", klok vremena, logično frekvencije 1Hz, a za njegovo usporavanje je korišćen "brojacV", brojač vremena, koji se svakim taktom povećava za jedan, dok ne stigne do 25000000, kada se invertuje "clkV". U istoj "always" petlji, koja iterira svakom pozitivnom ivicom "clk"-a, se vrednost "displej"-a prevodi odgovarajućom kombinacijom bitova "s\_seg"-a, tj. paljenjem odredjenih segmenata displeja (logičke vr. segmenata su invertovane, 0-upaljen).

Jednostavno, u "always" petlji, koja iterira svakom pozitivnom ivicom usporenog "clkV"-a, se dekrementuju sekunde, "vremeB", a ako je prošlo 60 sekundi dekrementuju se i minuti, "vremeA", a sekunde kreću ponovo od 59.

U petlji, koja iterira svakom pozitivnom ivicom usporenog "clkD"-a, se rešavaju dva različita problema. Jedan je smenjivanje i dodeljivanje vrednosti jednoj od četiri cifre displeja i drugi rešavanje pritiska tastera. Prvi problem je rešen jednim "case" blokom, u kome se zavisno od vrednosti žice "p\_2", tj. stanja prekidača "p2", na displeju prikazuju bcd vrednosti vremena ili rezultata, i svakom pozitivnom ivicom "clkD"-a se pali naredna cifra displeja, na primer ako je "cifra" jednaka 0111, tj. upaljena je najviša cifra (invertovano, 0-upaljen), sledeća je prva s' desna, koja ima različitu vrednost, tj. menja se vrednost "displej"-a i ta vrednost se ispisuje na datu cifru promenom "cifra" vrednosti na 1110, analogno je za ostale slučajeve unutar "case" bloka. Drugi problem, koji se rešava unutar ove petlje, je pritiskanje tastera. "clkD" pravi dovoljno kašnjenje (50K puta sporiji od oscilatora na ploči) da se izvrši debaunsiranje tastera, ali bi i dalje jednim pritiskom bilo zabeleženo više promena nego što je namenjeno datom tasteru. Zamisao je da se promena namenjena datom tasteru izvrši, tek pošto je on pušten, tj. vraćen u stanje logičke nule. Posmatrajmo taster "b1", kada je vrednost žice "b\_1", koja vodi od ovog tastera, jednaka jedinici, tj. kada je taster pritisnut, promenljivoj "stanjeB1" dodeljujemo vrednost 1, u međuvremenu će proći određen broj ciklusa, kada korisnik pušta taster, odnosno "b\_1" je jednako 0, tada se ulazi u "case" blok i zavisno od vrednosti prekidača "p0" i "p1" izvršena je promena i "stanjeB1" je ponovo 0. Analogno se izvršava i pritiskanje preostalih tastera.

## 5. ZAKLJUČAK

Košarka je najtrofejniji srpski sport, u kome se beleže odlični rezultati kako na klupskom, tako i na reprezentativnom planu, pa je zainteresovanost dece u porastu, što iziskuje potražnju za ovakvom vrstom proizvoda. Ovaj uređaj je pogodan za kačenje dodatnih perifernih modula, pa bi lako moglo da se izvede beleženje i ispisivanje još mnogo bitnih parametara ove igre, kao lične greške igrača, broj pojedinačnih ili timskih faulova itd., ali ovako zamišljen koncept apsolutno zadovoljava potrebe dečijih utakmica, odnosno ciljne grupe koju pogađa ovakav proizvod, a uz uvođenje pomenutih izmena u projekat ovakav uređaj bi mogao da upravlja i ozbiljnijim, profesionalnim sistemima, pa je najveća prednost ovako osmišljenog projekta dobar "temelj" za raznorazne dodatne zahteve.

## 6. LITERATURA

- [1] Spartan-3 FPGA Family  
(<https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>)
- [2] Spartan-3E FPGA Family Data Sheet  
([http://www.xilinx.com/support/documentation/data\\_sheets/ds312.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf))
- [3] Nexys 2 Reference Manual  
(<https://reference.digilentinc.com/reference/programmable-logic/nexys-2/reference-manual>)

## 7. PRILOG KODOVI

```

module kosarka(
    input clk,
    input b0,b1,b2,b3,
    input p0,p1,p2,
    output a,b,c,d,e,f,g,dp,
    output [3:0] an
);

reg clkD=0,clkV=0;
reg [16:0] brojacD=0;
reg [28:0] brojacV=0;

reg [7:0] vremeA=40,vremeB=0,rezulA=0,rezulB=0;

reg [3:0] displej = 4'b0110;
reg [6:0] s_seg = 0;
reg [3:0] cifra = 4'b1110;

reg stop = 1'b1;
wire [3:0] vremeAdesetice, vremeAjedinice, vremeBdesetice, vremeBjedinice;
wire [3:0] rezulAdesetice, rezulAjedinice, rezulBdesetice, rezulBjedinice;

wire b_0,b_1,b_2,b_3;
wire p_0,p_1,p_2;

//buttoni
assign b_0 = b0;
assign b_1 = b1;
assign b_2 = b2;
assign b_3 = b3;
//prekidaci
assign p_0 = p0;
assign p_1 = p1;
assign p_2 = p2;

reg stanjeB0 = 1'b0;
reg stanjeB1 = 1'b0;
reg stanjeB2 = 1'b0;
reg stanjeB3 = 1'b0;

bcd bcd_vremeA(clk, vremeA, vremeAdesetice, vremeAjedinice);
bcd bcd_vremeB(clk, vremeB, vremeBdesetice, vremeBjedinice);
bcd bcd_rezulA(clk, rezulA, rezulAdesetice, rezulAjedinice);
bcd bcd_rezulB(clk, rezulB, rezulBdesetice, rezulBjedinice);

always @(posedge clk)
begin
    if(brojacD==25000)
    begin
        clkD = ~clkD;
        brojacD=0;
    end
    brojacD = brojacD+1;

    if(brojacV==25000000)
    begin
        clkV = ~clkV;
        brojacV=0;
    end
    brojacV = brojacV+1;

    case(displej)
        4'd0 : s_seg = 7'b1000000;
        4'd1 : s_seg = 7'b1111001;
        4'd2 : s_seg = 7'b0100100;
        4'd3 : s_seg = 7'b0110000;
        4'd4 : s_seg = 7'b0011001;
        4'd5 : s_seg = 7'b0010010;
        4'd6 : s_seg = 7'b0000010;
        4'd7 : s_seg = 7'b1111000;
        4'd8 : s_seg = 7'b0000000;
        4'd9 : s_seg = 7'b0010000;
        default : s_seg = 7'b1000000;
    endcase
end
always @(posedge clkV)
begin
    if(stop==0)
    begin
        if(vremeB==0)
        begin
            vremeB = 59;
            vremeA = vremeA-1;
        end
        else
            vremeB = vremeB-1;
        end
    end
end

```

// Klok  
// Cetiri tastera  
// Tri prekidaca  
// Sedam segmenata displeja i tacka  
// Cetvorocifreni displej

// Clock displeja, clock vremena  
// Brojac displeja  
// Brojac vremena

// Pocetak utakmice 40min, 0sec, rezultat ekipaA 0, ekipaB 0

// Broj koji se trenutno ispisuje na displeju  
// Sedmosegmentna kombinacija  
// Koja od cetiri cifre je trenutno upaljena

// Zaustavlja vreme  
// BCD oblik za vremeA i vremeB  
// BCD oblik za rezulA i rezulB

// Za svaki taster je definicana zica  
// Zice koje idu od prekidaca

// Dodeljivanje vrednosti zicama svih tastera

// Zica uzima vrednost prekidaca

// Pamte da li je stisnut taster

// Iz binarnog u BCD oblik

// Inkrementuje do 25000, jer je clk ploce 50MHZ  
// clkD 1kHz

// Inkrementuje do 25000000, hiljadu puta sporije od clkD  
// clkV 1Hz

// 0  
// 1  
// 2  
// 3  
// 4  
// 5  
// 6  
// 7  
// 8  
// 9  
// 0

// Ako nije zaustavljeno..  
// Ako izbrojao 60 sekundi..  
// ..nov minut  
// ..prosao je minut, dekrementuj

// Nastavlja da odborojava sekunde (ako nije prosao minut)



```

always @(posedge clkD)
begin
    if(vremeA==0 && vremeB==0)
        stop = 1;
        case({p_2,cifra})
            5'b00111: begin displej = vremeBjedinice; cifra = 4'b1110; end
            5'b01110: begin displej = vremeBdesetice; cifra = 4'b1101; end
            5'b01101: begin displej = vremeAjedinice; cifra = 4'b1011; end
            5'b01011: begin displej = vremeAdesetice; cifra = 4'b0111; end

            5'b10111: begin displej = rezulBjedinice; cifra = 4'b1110; end
            5'b11110: begin displej = rezulBdesetice; cifra = 4'b1101; end
            5'b11101: begin displej = rezulAjedinice; cifra = 4'b1011; end
            5'b11011: begin displej = rezulAdesetice; cifra = 4'b0111; end
        endcase

    if(b_0 == 1'b1)
        stanjeB0 = 1'b1;
    if(stanjeB0==1'b1 && b_0 == 1'b0)
        begin
            stop = ~stop;
            stanjeB0 = 1'b0;
        end

    if(b_1 == 1'b1)
        stanjeB1 = 1'b1;
    if(stanjeB1==1'b1 && b_1 == 1'b0)
        begin
            case({p_1,p_0})
                2'b00: rezulB = rezulB+3;
                2'b01: rezulB = rezulB-3;
                2'b10: rezulA = rezulA+3;
                2'b11: rezulA = rezulA-3;
            endcase
            stanjeB1 = 1'b0;
        end

    if(b_2 == 1'b1)
        stanjeB2 = 1'b1;
    if(stanjeB2==1'b1 && b_2 == 1'b0)
        begin
            case({p_1,p_0})
                2'b00: rezulB = rezulB+2;
                2'b01: rezulB = rezulB-2;
                2'b10: rezulA = rezulA+2;
                2'b11: rezulA = rezulA-2;
            endcase
            stanjeB2 = 1'b0;
        end

    if(b_3 == 1'b1)
        stanjeB3 = 1'b1;
    if(stanjeB3==1'b1 && b_3 == 1'b0)
        begin
            case({p_1,p_0})
                2'b00: rezulB = rezulB+1;
                2'b01: rezulB = rezulB-1;
                2'b10: rezulA = rezulA+1;
                2'b11: rezulA = rezulA-1;
            endcase
            stanjeB3 = 1'b0;
        end
    end

assign an = cifra;

assign {g, f, e, d, c, b, a} = s_seg;
assign dp = 1'b1;

endmodule

```

// Ako je istekla utakmica  
// ..zaustavi

// Ako je prekidač p2 = 0, prikazi vreme na displeju  
// Smenjuje se trenutna "cifra" koja se prikazuje

// Ako je prekidač p2 = 1, prikazi rezultat na displeju

// Stisnut je taster b0  
// ..ako taster nije više stisnut

// ..zaustavlja/kreće vreme  
// Izvršena je namena tastera

// Stisnut je taster b1  
// ..ako taster nije više stisnut

// Taster b1 = +/- 3  
// Prekidač p1 određuje kojoj se ekipi dodeljuju bodovi  
// Prekidač p0, da li dodeljuju ili oduzimaju bodovi?

// Izvršena je namena tastera

// Stisnut je taster b2  
// ..ako taster nije više stisnut

// Taster b2 = +/- 2  
// Prekidač p1 određuje kojoj se ekipi dodeljuju bodovi  
// Prekidač p0, da li dodeljuju ili oduzimaju bodovi?

// Izvršena je namena tastera

// Stisnut je taster b3  
// ..ako taster nije više stisnut

// Taster b3 = +/- 1  
// Prekidač p1 određuje kojoj se ekipi dodeljuju bodovi  
// Prekidač p0, da li dodeljuju ili oduzimaju bodovi?

// Izvršena je namena tastera

// Svaki segment uzima vrednost jednog bita "s\_seg"-a  
// Ne koristi se decimalna tačka na displeju

**BCD modul**

```

module bcd(clk, broj, desetice, jedinice);
    input clk;
    input [7:0] broj;
    output reg [3:0] desetice;
    output reg [3:0] jedinice;

    reg [15:0] shift;
    reg [3:0] i = 0;

    always @(posedge clk)
    begin
        if(i == 0)
        begin
            shift[15:8] = 0;
            shift[7:0] = broj;
        end

        if(i<8)
        begin
            if (shift[11:8] >= 5)
                shift[11:8] = shift[11:8] + 3;

            if (shift[15:12] >= 5)
                shift[15:12] = shift[15:12] + 3;

            shift = shift << 1;
            i = i+1;
        end
        else
        begin
            desetice = shift[15:12];
            jedinice = shift[11:8];
            i = 0;
        end
    end
endmodule

```

```

// Klok
// Broj koji treba da se konvertuje
// BCD desetice
// BCD jedinice

// Unutrasnja promenljiva za cuvanje bitova
// Broj iteracija u petlji

// Ocisti prethodni broj i cuvaj novi u shift registru

// Iteriraj osam puta

// Ako je kolona jedinica >=5..
// ..dodaj 3

// Ako je kolona desetica >=5..
// ..dodaj 3

// Shiftuj ceo registar levo jednom

// Decimalni brojevi na izlaz

```

**Implementacija**

```

// Dodeljivanje pinova za prekidače
NET "p0" LOC = "G18";
NET "p0" CLOCK_DEDICATED_ROUTE = FALSE;
NET "p1" LOC = "H18";
NET "p1" CLOCK_DEDICATED_ROUTE = FALSE;
NET "p2" LOC = "K18";
NET "p2" CLOCK_DEDICATED_ROUTE = FALSE;

//Dodeljivanje pinova za tastere
NET "b0" LOC = "B18";
NET "b0" CLOCK_DEDICATED_ROUTE = FALSE;
NET "b1" LOC = "D18";
NET "b1" CLOCK_DEDICATED_ROUTE = FALSE;
NET "b2" LOC = "E18";
NET "b2" CLOCK_DEDICATED_ROUTE = FALSE;
NET "b3" LOC = "H13";
NET "b3" CLOCK_DEDICATED_ROUTE = FALSE;

//Dodeljivanje pinova za 7-segmentni displej
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 "dp" LOC = "C17" ;
NET "an<3>" LOC = "F15";
NET "an<2>" LOC = "C18";
NET "an<1>" LOC = "H17";
NET "an<0>" LOC = "F17";

# Pin assignment for clock
NET "clk" LOC = "B8";

```