

 <p>UCG Univerzitet Crne Gore</p>	<p>UNIVERZITET CRNE GORE ELEKTROTEHNIČKI FAKULTET</p>	
STUDIJSKI PROGRAM:	<i>ELEKTRONIKA, TELEKOMUNIKACIJE I RAČUNARI</i>	
PREDMET:	<i>PROJEKTOVANJE DIGITALNIH SISTEMA</i>	
FOND ČASOVA:	<i>3 + 0 + 1</i>	

LABORATORIJSKA VJEŽBA

NAZIV:	<i>REALIZACIJA KOLA ZA DETEKCIJU BINARNIH TETRADA KOJE IMAJU DVIJE ILI VIŠE JEDINICA</i>
<p>CILJEVI VJEŽBE:</p> <ul style="list-style-type: none"> - praktičan rad sa SPARTAN 3E Starter kit razvojnom platformom i Xilinx ISE razvojnim okruženjem, - povezivanje teorijske osnove (predavanja) sa praktičnom realizacijom, - posmatranje promjena na izlazima kola u zavisnosti od ulaznih vrijednosti, - utvrđivanje načina rada kola za detekciju binarnih tetrađa. 	
<p>POTREBAN PRIBOR:</p> <ul style="list-style-type: none"> - pribor za pisanje. 	

IME I PREZIME: _____.

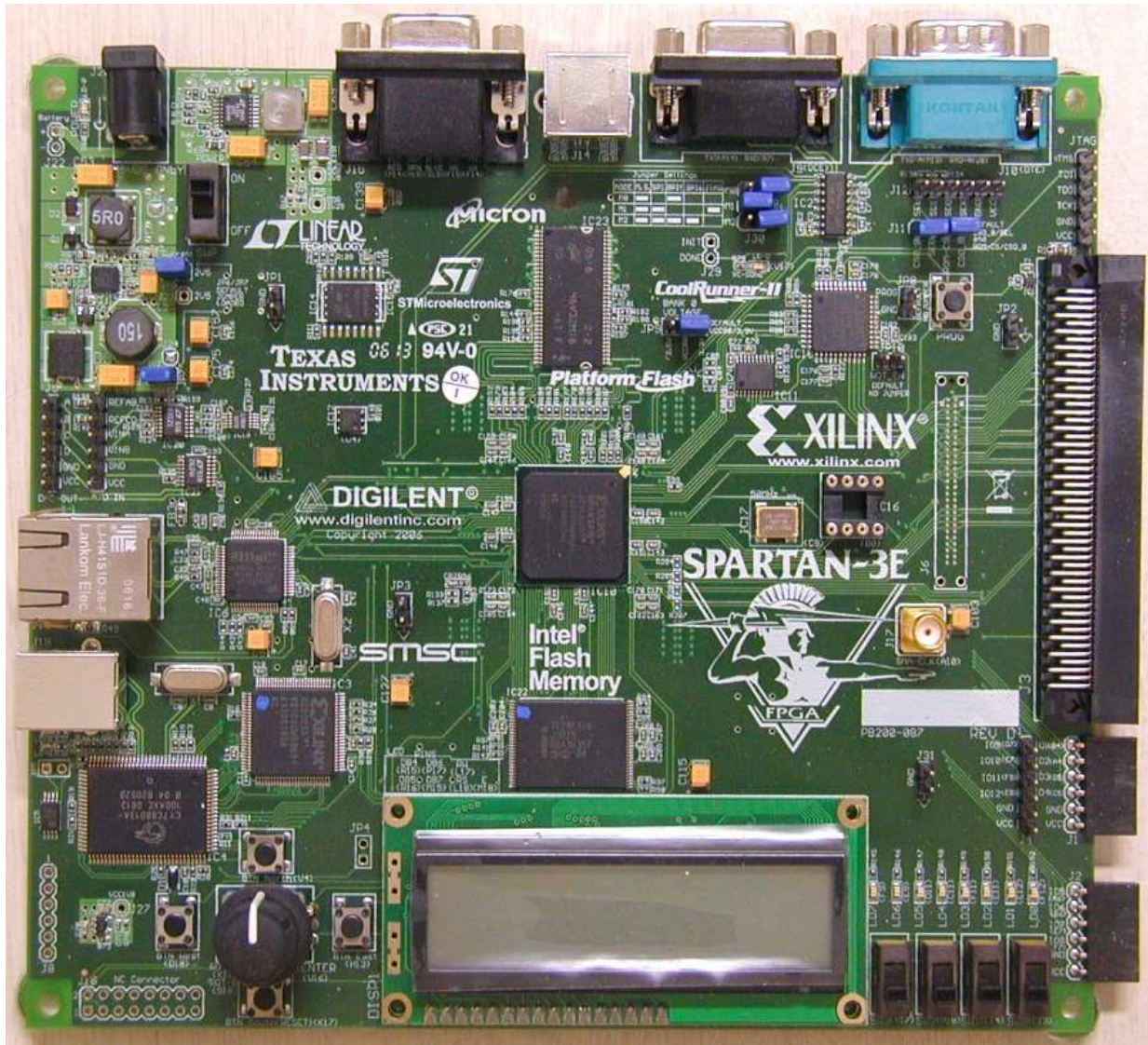
BROJ INDEKSA: _____.

BROJ POENA:	
OVJERAVA:	
DATUM:	

1. APARATURA

Na raspolaganju su sljedeći uređaji i oprema:

- SPARTAN 3E Starter Kit razvojna platforma,
- računar,
- napajanje,
- USB kabal,
- Xilinx ISE Design Suite v.14.7 razvojno okruženje.



Slika 1. Izgled SPARTAN 3E Starter Kit razvojne platforme

2. TEORIJSKA OSNOVA LABORATORIJSKE VJEŽBE

Binarnom tetradom se naziva grupa od četiri binarne cifre. Često se koristi za zapisivanje dekadnih brojeva u binarnom brojnom sistemu. Na primjer, binarno kodirani decimalni zapis (BCD) predstavlja binarne tetrade pomoću kojih se binarno predstavljaju dekadne cifre.

Prilikom realizacije kola za detektovanje binarnih tetrada koje imaju dvije ili više jedinica polazimo od Karnoove mape koja ima sledeći oblik:

CD \ AB	00	01	11	10
00			1	
01		1	1	1
11	1	1	1	1
10		1	1	1

Jedinice su upisane u ona polja kod kojih bar dvije promjenljive imaju vrijednost logičke jedinice. Zaokruživanjem odgovarajućih površina u mapi dolazi se do sljedeće funkcije:

$$F = AB + CD + BD + BC + AD + AC$$

Na ulaz kola treba dovesti četiri binarna signala čije vrijednosti možemo mijenjati i posmatrati izlaz. Prilikom implementacije, ulazima kola ćemo dodijeliti pinove koji su povezani sa prekidačima na razvojnoj platformi, a izlazu pin koji je povezan sa LED4 (slika 8). Promjenom položaja prekidača dovodimo logičku jedinicu ili logičku nulu na pinove, i time postavljamo vrijednosti ulaznih promjenljivih. Ukoliko kolo detektuje na ulazu dvije ili više jedinica izlaz će biti postavljen na nivo logičke jedinice, pa će LED svijetliti. U suprotnom LED će biti ugašena, što znači da je izlazni signal na nivou logičke nule.

2.1 Verilog kod za realizaciju kola za detekciju binarnih tetrada

Na osnovu gornje funkcije, lako se može napisati verilog kod za implementaciju traženog kola:

```

module detektor (out, a, b, c, d);
output out;
input a,b,c,d;

assign out = (a&b) | (c&d) | (b&d) | (b&c) | (a&d) | (a&c);

endmodule

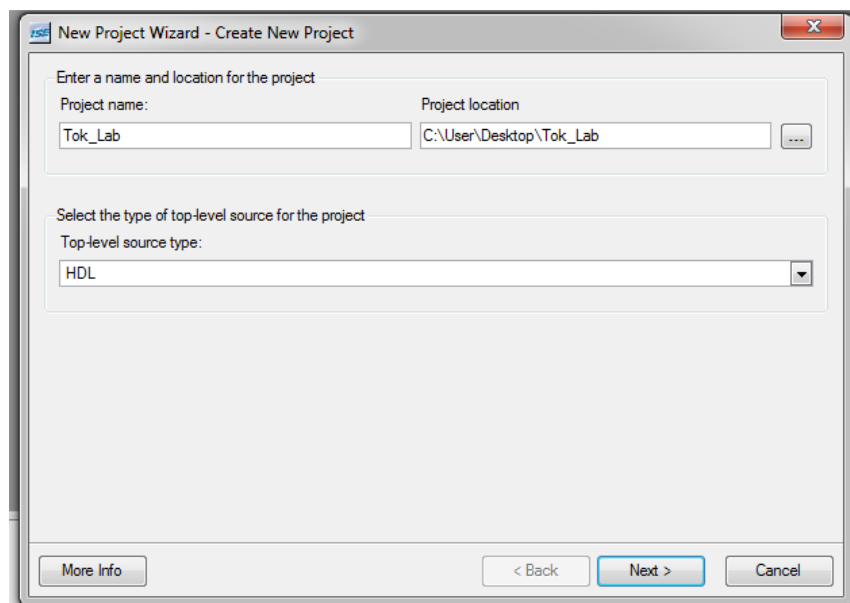
```

3. ZADACI LABORATORIJSKE VJEŽBE

Kreiranje dizajna

Pokrenuti *ISE Project Navigator* i napraviti novi projekat.

1. **Start**→**Programs**→**Xilinx Design Tools**→**ISE Design Suite 14.7**→**ISE Design Tools** →**Project Navigator**
2. U dobijenom prozoru treba izabrati **File**→**New Project**. Otvoriće se **New Project Wizard** (slika 1).

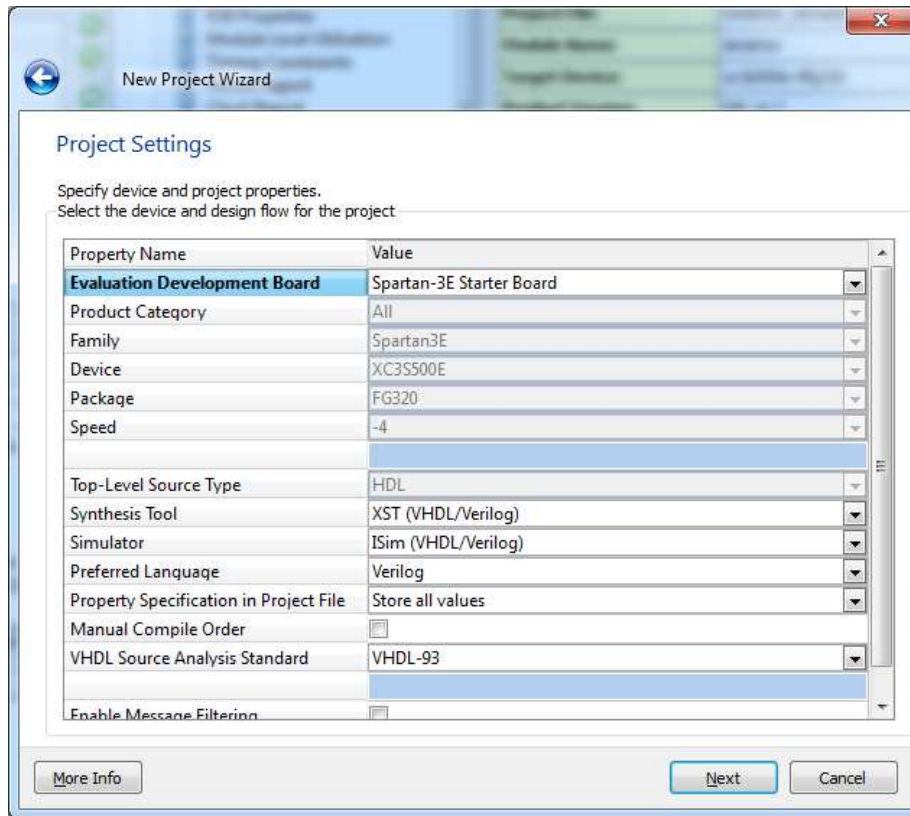


Slika 1: Prozor *New Project Wizard*

Dati ime novom projektu (npr. **detektor**), a zatim odrediti mjesto na hard disku gdje želite da vaš projekat i njegovi prateći fajlovi budu sačuvani (...)=**Browse**.

3. **<Next>**

Pojaviće se prozor prikazan na slici 2.

Slika 2: Prozor *Device Properties*

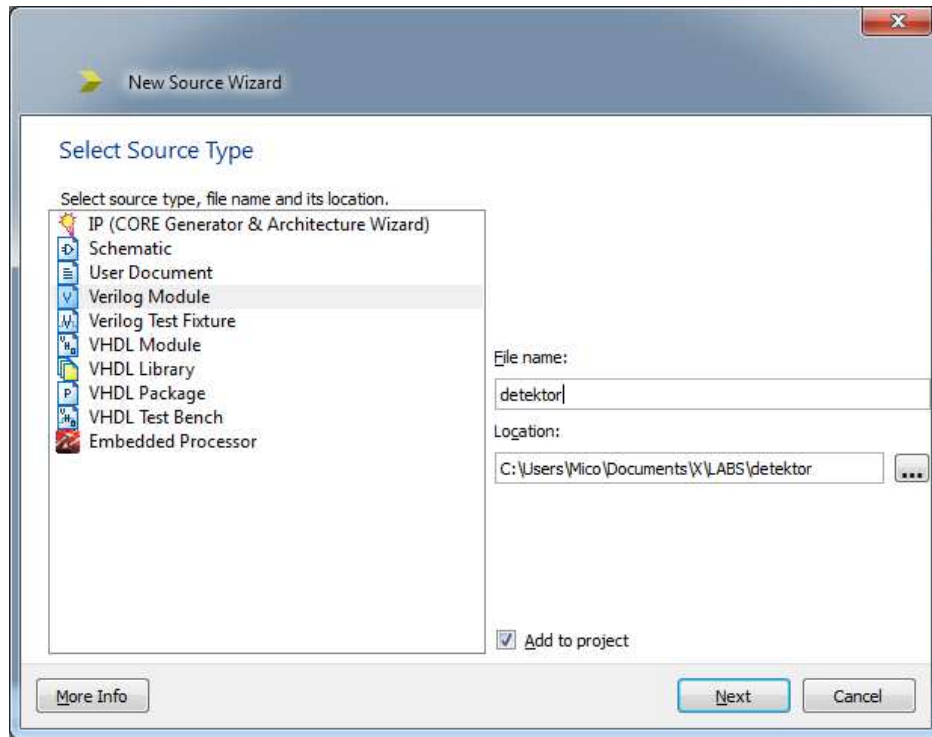
Pojaviće se dijalog za odabir uređaja i toka dizajna i treba odabrati sledeća podešavanja:

Device Family: **Spartan3E**
 Device: **xc3s500E**
 Package: **fg320**
 Speed Grade: **-4**
 Synthesis Tool: **XST (VHDL/Verilog)**
 Simulator: **ISim (VHDL/Verilog)**
 Preferred Language: **Verilog**

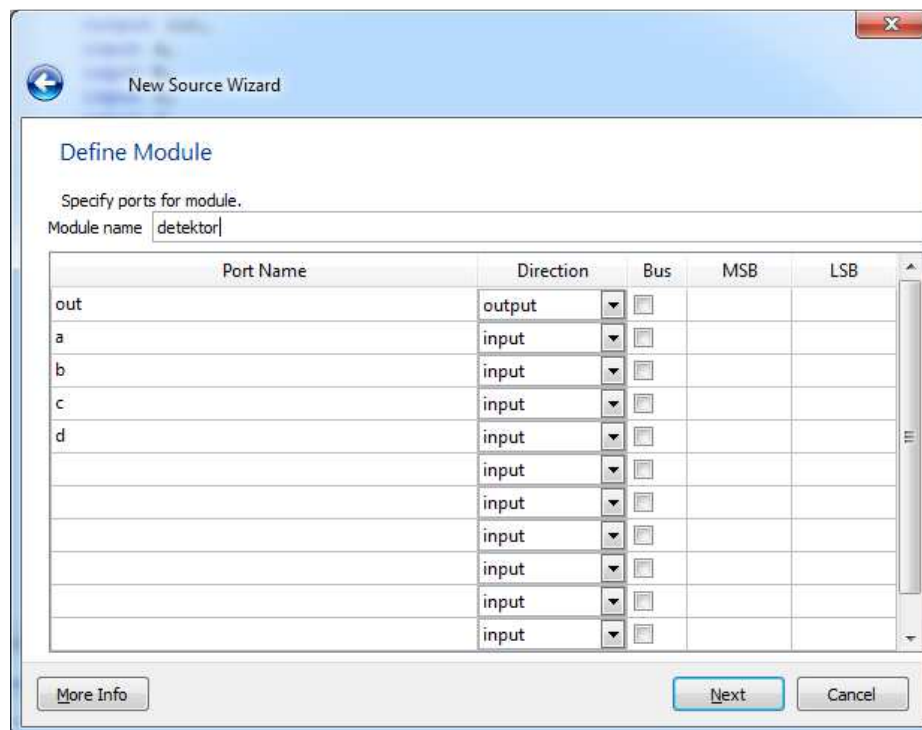
Napomena: Mogli smo pod **Evaluation Development Board** izabrati **Spartan-3E Starter Board** i tada bi se automatski podesile opcije za FPGA čip.

4. <Next>, pa <Finish>
5. Izabrati **Project**→**New Source**.

Pojaviće se sledeći prozor (**New Source Wizard** - slika 3) u kome treba izabrati opciju **Verilog Module**, dati ime fajlu (npr. **detektor**) i pritisnuti <Next>.

Slika 4: Prozor *New Source Wizard*

6. U sledećem prozoru unijeti nazive portova i njihove smjerove, kako je to prikazano na slici 5 i izabrati <Next> i <Finish>. Na pitanje da li želite da kreirate folder odgovoriti sa <Yes>.



Slika 5: Unos portova

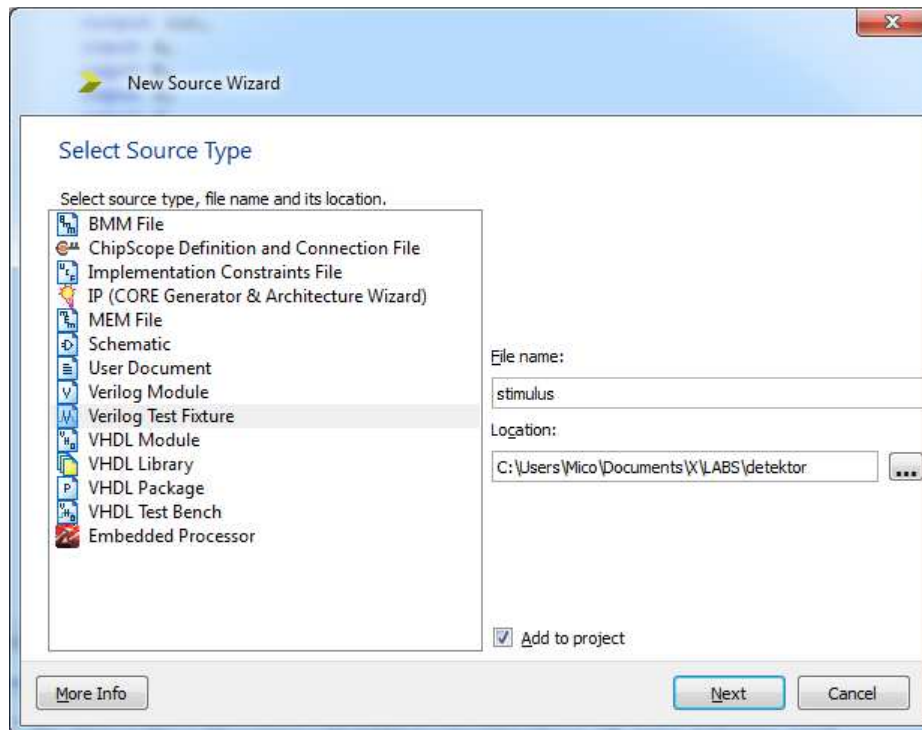
7. Dobijeni okvir za kreiranje modula dopuniti funkcionalnošću modula saglasno kodu prikazanom u poglavlju 2.1:

$$\text{assign out} = (\text{a\&b}) | (\text{c\&d}) | (\text{b\&d}) | (\text{b\&c}) | (\text{a\&d}) | (\text{a\&c});$$

8. Snimiti izmjene u fajlu: **File**→**Save**

Provjera funkcionalnosti

1. Da bi provjerili funkcionalnost dizajna, treba napraviti *stimulus* i izvršiti simulaciju. Izabrati **Project**→**New Source**, u novom prozoru selektovati **Verilog Test Fixture** i dati ime stimulusu (npr. **stimulus**) (slika 6). Potom izabrati **<Next>**, **<Next>**, **<Finish>**



Slika 6: Kreiranje stimulusa

2. Dobijeni fajl dopuniti kodom za testiranje detektora na sledeći način:

```
initial begin
```

```
  $monitor($time, "a=%b b=%b c=%b d= %b out=%b",a,b,c,d,out);
```

```
  // Initialize Inputs
```

```
  a = 0; b = 0; c = 0; d = 0;
```

```
  #10 a = 0; b = 0; c = 0; d = 1;
```

```
  #10 a = 0; b = 0; c = 1; d = 0;
```

```
  #10 a = 0; b = 0; c = 1; d = 1;
```

```
  #10 a = 0; b = 1; c = 0; d = 0;
```

```
  #10 a = 0; b = 1; c = 0; d = 1;
```

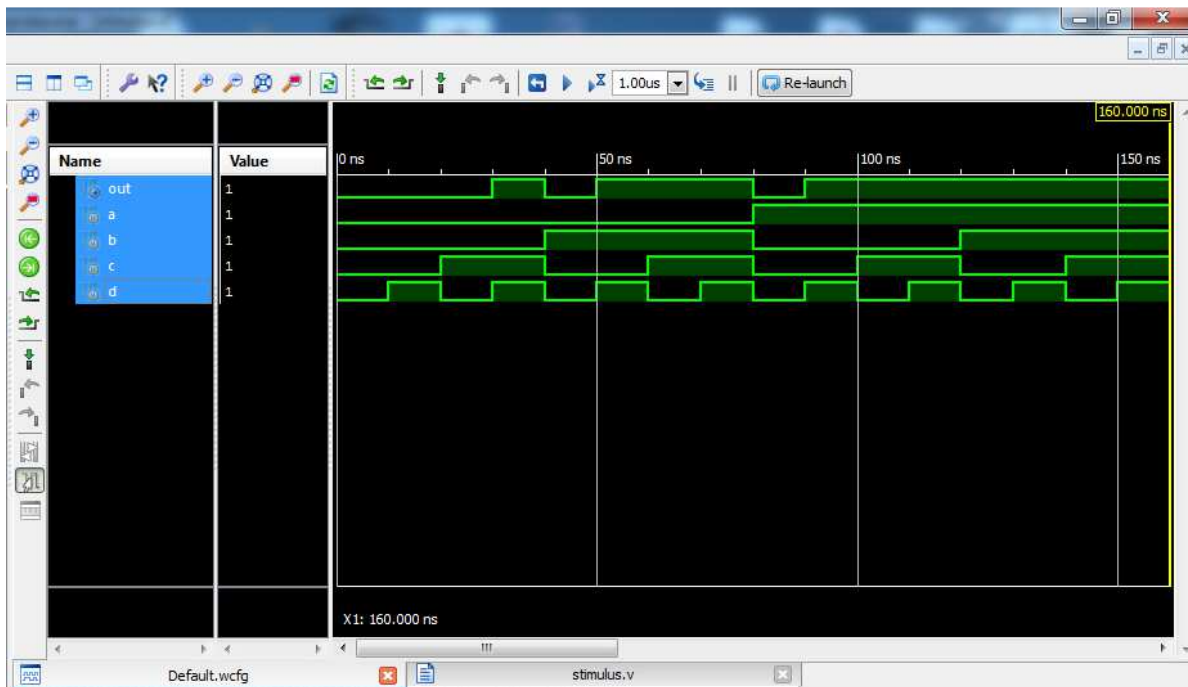
```
  #10 a = 0; b = 1; c = 1; d = 0;
```

```

#10 a = 0; b = 1; c = 1; d = 1;
#10 a = 1; b = 0; c = 0; d = 0;
#10 a = 1; b = 0; c = 0; d = 1;
#10 a = 1; b = 0; c = 1; d = 0;
#10 a = 1; b = 0; c = 1; d = 1;
#10 a = 1; b = 1; c = 0; d = 0;
#10 a = 1; b = 1; c = 0; d = 1;
#10 a = 1; b = 1; c = 1; d = 0;
#10 a = 1; b = 1; c = 1; d = 1;
#10 $finish;

```

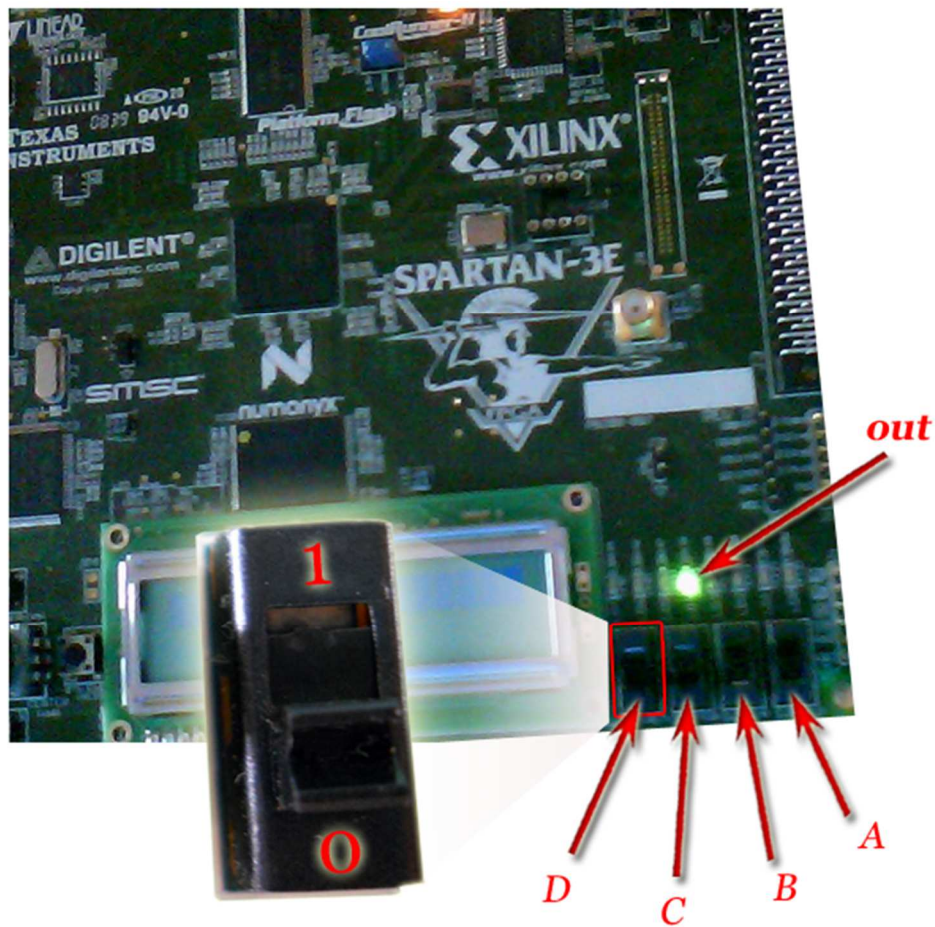
- U **Design View** prozoru selektovati opciju **Simulation**, a u **Design Hierarchy** prozoru selektovati **stimulus**.
- U **Processes** prozoru proširiti **Isim Simulator** i dva puta kliknuti na **Simulate Behavioral Model**. Otvorit će se ISim simulator i prikazat će se rezultati simulacije kao na slici 7.



Slika 7: Rezultati simulacije

Implementacija

- Nakon što se uvjerimo da kolo ispravno funkcioniše, prelazimo na implementaciju. Prvi korak je pridruživanje odgovarajućih pinova FPGA čipa signalima unutar našeg dizajna. Ulazne signale ćemo povezati na prekidače koji se nalaze na Spartan 3E Starter Kit razvojnoj platformi, a izlazni signal na LED 4 (slika 8). U tu svrhu možemo koristiti program **PlanAhead**, kao što je to prikazano u laboratorijskoj vježbi „Architecture Wizard i PACE Lab (PlanAhead)“. Međutim, ovdje ćemo isti zadatak obaviti na drugačiji način.



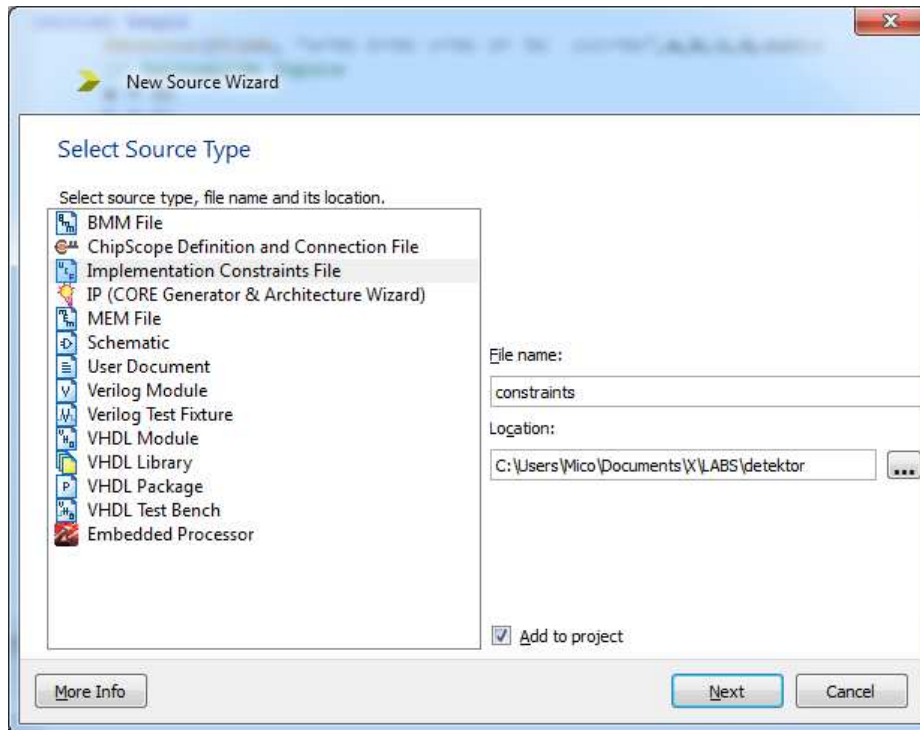
Slika 8: Položaj prekidača i LED koji se koriste u ovoj vježbi

2. U **Design View** prozoru selektovati opciju **Implementation**, a u **Design Hierarchy** prozoru selektovati **detektor**.
3. Izabrati **Project**→ **New Source**, u novom prozoru selektovati **Implementation Constraints File** i dati mu ime (npr. **constraints**) (slika 9). Potom izabrati <Next>, <Finish>
4. Upravo kreirani fajl, uz pomoć dokumenta *Spartan-3E Starter Kit Board User Guide*, unijeti kod kojim definišemo povezivanje signala unutar dizajna sa pinovima FPGA čipa:

```
NET "a" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP ;
NET "b" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP ;
NET "c" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP ;
NET "d" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP ;
```

```
NET "out" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
```

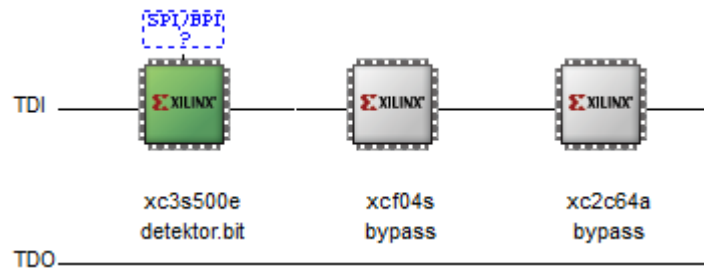
5. Snimiti unijete izmjene u *constraint* fajl: **File**→ **Save**
6. U **Processes** prozoru dva puta kliknuti na **Generate Programming File** (pri čemu su i dalje selektovane stavke iz koraka 2) i sačekati da se proces završi.
7. Uključiti Spartan-3E Starter Board razvojnu platformu i povezati je sa računarom pomoću USB kabla.

Slika 9: Kreiranje *constraints* fajla

8. U **Processes** prozoru raširiti **Configure Target Device** i dva puta kliknuti na **Manage Configuration Project (iMPACT)**.
9. Kada se pokrene program **iMPACT** u prozoru **iMPACT Flows** dva puta kliknuti na **Boundary Scan**. Nakon toga desnim tasterom miša kliknuti na praznu bijelu površinu na kojoj piše „*Right click to Add Device or Initialize JTAG chain*“ i izabrati opciju **Initialize Chain**. Na pitanje „*Do you want to continue and assign configuration file(s)*“ odgovoriti sa **<Yes>**.
10. Kada se pojavi dijalog **Assign New Configuration File**, izabere se *detektor.bit* fajl (odnosno fajl za ekstenzijom **.bit** i istim imenom kao i verilog fajl u projektu) za **xc3s500e** (prvi uređaj u JTAG ‘lancu’) i klikne se **<Open>**.

Napomena: ako se pojavi upozoravajuća poruka da je inicijalni *clock* promijenjen na JTAG-ov, samo treba kliknuti **<OK>**.

Napomena 2: ako se pojavi pitanje „*Do you want to attach an SPI or BPI PROM to this device*“ odgovoriti sa **<No>**.
11. Za ostale uređaje, kliknuti **Bypass**.
12. Kada se pojavi prozor **Device Programming Properties** samo kliknuti **<OK>**. Izgled radnog prozora prikazan je na slici 10.



Slika 10. JTAG lanac sa dodijeljenim konfiguracionim fajlom

13. Kliknuti desnim klikom na ikonicu čipa **xc3s500e** koji je označen zelenom bojom (ako nije označen prvo ga selektovati lijevim tasterom miša) u radnom prozoru i odabrati **Program**.
14. Mijenjati položaje prekidača na razvojnoj platformi kako bi se promjenljivim A, B, C i D dodjeljivale vrijednosti iz tabele 1 i promjene pratiti na LED. Na osnovu dobijenih rezultata popuniti tabelu 1.

A	B	C	D	OUT
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Tabela 1

4. ZAKLJUČAK