SIMULACIJA FUNKCIONALNOSTI DIZAJNA

Startovati **ISE Project Navigator** i kreirati novi projekat: **File->New Project**. Nakon unosa naziva projekta even_detector, potrebno je izvršiti određena podešavanja u okviru **Project Settings**, slika 1.

Project Settings		
Specify device and project properties.		
elect the device and design flow for the pr	oject	
Property Name	Value	
Evaluation Development Board	None Specified	-
Product Category	All	-
Family	Spartan3E	-
Device	XC3S500E	-
Package	FG320	-
Speed	-4	-
Top-Level Source Type	HDL	-
Synthesis Tool	XST (VHDL/Verilog)	-
Simulator	ISim (VHDL/Verilog)	-
Preferred Language	VHDL	-
Property Specification in Project File	Store all values	-
Manual Compile Order		
VHDL Source Analysis Standard	VHDL-93	-
Enable Message Filtering		

Slika 1. New Project Wizard

Izabrati <Next>, a potom <Finish>.

U okviry **Hierarchy**, izabrati even_detector, desni klik miša, a potpom **New Source**. U okviru **Select Source Type**, izabrati **VHDL Module**. Dodijeliti naziv even_detector, **<Next>**. Uz pomoć prikazanog **New Source Wizard**-a definisati modul kako je prikazano na slici 2.

Izabrati <Next>, a potom <Finish>.

Unijeti kod dat u listingu 1.

New So	urce Wizard					
Define Modu	ıle					
Specify ports for r	module.					
Entity name	even_detector					
Architecture name	arc_even_detector					
	Port Name	Direction			MSB	LSB
a		in	-	V	2	0
even		out	-			
		in	-			
		in	-			
		in	-			
		in	-			
		in	-			
		in	-			
		in	-			
		in	-			

Slika 2. New Source Wizard

Listing 1

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity even_detector is
    Port ( a : in STD_LOGIC_VECTOR (2 downto 0);
        even : out STD_LOGIC);
end even_detector;
architecture arc_even_detector of even_detector is
        signal odd: std_logic;
begin
        even <= not odd;
        odd <= a(2) xor a(1) xor a(0);
end arc_even_detector;
```

Radno okruženje je prikazano na slici 3. Izabrati opciju **Check Syntax**, kako bi izvršili provjeru sintakse koda.



Slika 3. Radno okruženje. Provejra sintakse koda.

U okviru **Hierarchy**, izabrati <u>even_detector.vhd</u>, desni klik miša, a potpom **New Source**. Izabrati **VHDL Test Bench**, dodijeliti naziv test_even_detector, <**Next**>, kao **Associate Source** izabrati even_detector, <**Next**>, <**Finish**>.

Unijeti kod dat u listingu 2. Nakon što je unijet odgovarajući kod, potrebno je u **Design** prozoru odabrati opciju **Simulaton** i pokrenuti simulaciju dvoklikom na **Simulate Behavioral Model**, slika 4.

Nakon nekog vremena, prikazaće se prozor sa rezultatima simulacije, slika 5. U desnom dijelu, nalazi se dijagram sa talasnim oblicima signala. Kako je prikazano stanje signala na samom kraju simulacije, potrebno je vratiti se na njen početak pomjeranjem klizača koji se nalazi ispod dijagrama. Međutim, moguće je da razmjera na vremenskoj osi nije pogodna za pregled ovih signala pa je najkraći metod da se pritiskom na desni taster miša (dok se kursor nalazi na prostoru dijagrama) izabere opcija **To Full View**. Lijevo od dijagrama nalaze se dvije kolone: **Name** i **Value**. U koloni **Name** nalaze se imena signala čiji su dijagrami prikazani. Ovi signali se mogu rasporediti u proizvoljan redoslijed prostim prevlačenjem sa jedne pozicije na drugu pritiskom tastera miša. Moguće je i obrisati signale koji nisu od interesa.

U kolni **Value** se nalaze vrijednosti signala na poziciji na kojoj se trenutno nalazi kursor (žuta linija). Kursor se može pomjerati uz pritisak tastera miša. Listing 2

```
entity even detector testbench is
end even detector testbench;
architecture tb arch of even detector testbench is
       component even detector
              port
               (
                      a: in std_logic_vector(2 downto 0);
                      even: out std logic
              );
       end component;
       signal test_in: std_logic_vector(2 downto 0);
       signal test_out: std_logic;
begin
--- instanciranje kola koje se testira ---
       uut: even detector
              port map( a => test in, even => test out );
--- test vector generator ---
       process
       begin
              test in <= "000";
              wait for 200 ns;
              test in <= "001";
              wait for 200 ns;
              test_in <= "010";</pre>
              wait for 200 ns;
              test in <= "011";
              wait for 200 ns;
              test in <= "100";
              wait for 200 ns;
              test_in <= "101";
wait for 200 ns;
              test_in <= "110";</pre>
              wait for 200 ns;
              test_in <= "111";</pre>
              wait for 200 ns;
       end process;
--- verifier ---
process
       variable error status: boolean;
begin
       wait on test in;
       wait for 100 ns;
       if ((test in = "000" and test out = '1') or
              (test_in = "001" and test_out = '0') or
(test_in = "010" and test_out = '0') or
(test_in = "011" and test_out = '1') or
               (test in = "100" and test out = '0') or
               (test_in = "101" and test_out = '1') or
              (test_in = "110" and test_out = '1') or
              (test in = "111" and test out = '0'))
       then
              error_status := false;
       else
              error_status := true;
       end if;
----- error reporting -----
       assert not error status
```

report "test failed."
severity note;
end process;
end tb_arch;

🍃 ISE Project Navigator (P.20131013) - C:\Users\Milena\Tehnologija mikrosistema\Primjeri\even_detector\even_detector.xise - [test_even 💻 💻 🗴									
	File Edit View Project Source Process Tools	Wi	ndow La	yout Help – 🗗 🗙					
	× ∽ ≈ X 🗊 🛱 😽 😽	Æ	P 👂 🖉) / 2 🕞 🗄 🗉 🖻 🥬 / Y 🕨 🗴 📌 💡					
Desig	jn ↔□₽×	4	69	entity even detector testbench is					
-*	View: 🔘 🌆 Implementation 💿 🎆 Simulation	ÞE	70	end even_detector_testbench;					
E.	Behavioral 🗸	-	71						
0El	Hierarchy	=	72	architecture the arch of even_detector_testbench is					
81		2	74	component even_detector					
	\Rightarrow $(1 + 1)$		75						
	even detector testbench - tb arch (test ever	10	76	a: in std logic vector(2 downto 0);					
•	uut - even_detector - arc_even_detector	_	77	even: out std logic					
6 7		1	78);					
		%	79	end component;					
×		*26	80	signal test_in: std_logic_vector(2 downto 0);					
Ť		X	81	signal test_out: std_logic;					
	No Processes Running	2	83	begin -					
B)	Processes: uut - even detector - arc even detector	\bigcirc	84	instanciranje kola koje se testira					
++ +		0	85						
×	Behavioral Check Syntax	_	86	uut: even_detector					
90	Simulate Behavioral Model		87	<pre>port map(a => test_in, even => test_out);</pre>					
	Uetrai		88						
			89	test vector generator					
			90	DYOGRAM					
			91	begin					
			< <u>Ⅲ</u>	•					
	Start 🔍 Design 🖺 Files 🚺 Libraries	Σ	Design Su	mmary 🗵 📄 even_detector.vhd 🗵 📄 test_even_detector.vhd 🗵					
Warr	nings			↔□₽×					
•				Þ					
	Console 😢 Errors 🔬 Warnings 🕅 Find in Files I	Resul	ts	Ln 123 Col 47 VHDL					
		_							

Slika 4. Radno okruženje. Simulacija funkcionalnog modela.



Slika 5. ISim Simulator. Simulacioni dijagrami.

Provjera ispravnosti rada modula se može izvršiti vizuelnim pregledom dijagrama, tako što se za svaku kombinaciju ulaznih signala provjeri vrijednost na izlazima.

Nakon završetka simulacije treba zatvoriti **ISim**. Pokretanje dvije instance simulatora nije dozvoljeno i često može da izazove probleme u radu.

Ukoliko se promijeni neki dio koda u **ISE** razvojnom okruženju, najlakše je restartovati simulaciju tako što će se zatvoriti **ISim** simulator, pa ponovo pokrenuti proces **Simulate Behavioral Model** unutar **ISE** razvojnog okruženja.

Umjesto vizuelnog pregleda dijagrama nakon svake simulacije, moguće je napisati testni kod koji će automatski provjeriti ispravnost izlaznih signala. Za svaki izlazni signal može se napisati task koji prihvata ulazni parametar (koji sadrži očekivanu vrijednost izlaza) i provjerava da li se njegova vrijednost poklapa sa trenutnim stanjem na izlazu. Ako tokom simulacije dođe do grešaka, na konzoli će se ispisati vremenski trenutak kada se greška dogodila, trenutna vrijednost izlaznog signala i vrijednost koja se očekivala. Ako nije bilo grešaka, na konzoli se neće ništa ispisivati.

VREMENSKA SIMULACIJA DIZAJNA

Osim funkcionalne ispravnosti dizajna, neophodno je voditi računa i o ostalim aspektima koji utiču na ispravno funkcionisanje uređaja koji dizajniramo. Jedan od važnih faktora je i vrijeme koje je potrebno za propagaciju pojedinih signala kroz FPGA kolo. Stoga je potrebno izvršiti i tzv. vremensku simulaciju dizajna kako bi se procijenilo kojom se brzinom prostiru signali i na taj način izbjeglo nepredviđeno ponašanje nakon upisa dizajna u FPGA čip.

Da bi se izvršila vremenska simulacija i analiza, **ISE** mora izvršiti procese synthesize, translate, map i place & route. U tu svrhu je potrebno izabrati opciju **Implementation**, slika 6. Navedeni procesi se nalaze unutar grupa **Synthesize – XST** i **Implement Design**.

Dvostrukim klikom na **Design Summary/Reports** dobija se detaljan izvještaj o rezultatu procesa dizajna, slika 7.

Unutar sekcije **Detailed Reports** može se izabrati **Post PAR Static Timing Report**, kako bi se vidjelo kašnjenje koje nastaje zbog propagacije signala, slika 8.

Unutar sekcije **Synthesis Report** se može naći informacija o maksimalnom kašnjenju kombinacionih kola, koje u datom primjeru iznosi 6,209 ns, slika 9. Ukoliko bi se ulazni signali mijenjali brže od ove vrijednosti, moglo bi doći do neočekivanog ponašanja našeg dizajna.

ISE Project Navigator (P.20131013) - C:\Users\Milena\Tehnologija	mikrosistema\Primjeri\even_detector\even_detector.xise - [even_detector.vhd]						
File Edit View Project Source Process Tools Wind	w Layout Help	_ 8 ×					
🗋 ờ 🗟 🕼 🐇 🖉 🙆 🗶 🖄 👘 🗟 🐔	፼፼≠፪ 🔽 ጜେ = = = ₽ 🛠 ▶ Σ 🛠 💡						
Design ↔ G X Wex: @ ∰ Implementation @ Smudation Hierarchy G U Constraints Hierarchy G U Constraints Hierarchy	<pre>10 Tool versions: 11 Description: 12 13 Dependencies: 14 15 Revision: 16 Revision 0.01 - File Created 17 Additional Comments: 18 19</pre>						
No Processes Running Processes even, detector - arc_even, detector Design Summary/Reports Design Villes User Constraints User Constraints User Constraints User Constraints User Constraints User Constraints	<pre>28 any Xilinx primitives in this code. 29library UNISH: 30use UNISH: VComponents.all; 31 32 entity even_detector is 33 Fort (a : in STD_LOSIC_VECTOR (2 downto 0); 34 even : out_STD_LOSIC_VECTOR (2 downto 0); 35 end even_detector; 36 37 architecture arc_even_detector of even_detector is 38 signal odd: std_logic; 39 begin 40 even <= not odd; 41 odd <= a(2) xor a(1) xor a(0); 42 end arc_even_detector; 43 44 44 44 44 44 44 44 44 44 44 44 44</pre>	E •					
Warnings ++ □ 8 ×							

Slika 6. Radno okruženje. Implementacija.

≽ ISE Project Navigator (P.20131013) - CAUsers/Milena/Tehnologija mikrosistema/Primjeri/even_detector/even_detector/xize - [Design Summary]											
∑ File Edit View Project Source Process Tool	s Wi	indow Layout Help								E ×	
○20日間	Æ	/ / / / / / / / / / / / / / / / / / / /	-	× № ≥ 1	۴ 💡						
Design ↔ □ ♂ ×	۹ 🎧	Design Overview	*	1	even_detect	or Proje	ect Status				
View: Vi	- 6	IOB Properties		Project File:	even_detector.xise	Pars	er Errors:		No Errors		
Hierarchy	00	Module Level Utilization Timing Constraints	ш	Module Name:	even_detector	Imp	lementatio	n State:	Placed and Routed		
com detector and detector (non		Pinout Report Clock Report		Target Device:	xc3s500e-4fg320		• Errors:		No Errors		
		Static Timing		Product Version:	ISE 14.7		• Warning	15:	No Warnings		
	30	Errors and Warnings Parser Messages Synthesis Messages		Design Goal:	Balanced		Routing Results:		All Signals Completely Routed		
	100	Translation Messages Map Messages		Design Strategy:	Xiinx Default (unlocked)		• Timing Constra	ints:			
		Place and Route Messages Bitgen Messages		Environment:	System Settings		• Final Tin Score:	ning	0 <u>(Timing</u> Report)		
		All Implementation Messages									
		Synthesis Report			Device Utilization Summary				Ŀ	1	
No Processes Running	ocesses Running	Translation Report		Logic Utilization		Used	Available	Utilizatio	Note(s)		
Processes: even detector - arc even detector	7	Place and Route Report	E	Number of 4 input LU	Лīs	1	9,312	19		-	
	ŧ.	Post-PAR Static Timing Report Dever Report		Number of occupied Slices		1	4,656	19	2	-	
Design Utilities		D Disco Durant		logic	containing only related	1	1	1009	2		
Synthesize - XST View RTL Schematic		Design Properties		Number of Slices o logic	containing unrelated	0	1	09			
View Technology Schematic		Show Clock Report Show Failing Constraints Show Warnings Show Errors		Total Number of 4 in	put LUTs	1	9,312	19	5		
Generate Post-Synthesis Simulation				Number of bonded I	OBs	4	232	19	•	-	
Contraction				Average Fanout of N	ut of Non-Clock Nets 1.00						
B C Map											
Place & Route					Performance	Summary			E	1	
Configure Target Device				Final Timing Score	e: 0 (Setup: 0, Hold	: 0)	Pino	it Data:	Pinout Report	<u>t</u> _	
Start 🔍 Design 🟠 Files 🏠 Libraries	E	Design Summary 🔯 📄 even_	dete	ctor.vhd	test_even_detecto	or.vhd		, Data:	Clark Donart		
Warnings									++ 🗆	ð×	
										*	
📱 Console 🔕 Errors 🔔 Warnings 🐹 Find in File	s Resu	its	-			_				_	
· · · · · · · · · · · · · · · · · · ·											
	-									-	

Slika 7. Radno okruženje. Pregled karakteristika dizajna.



Slika 8. Radno okruženje. Propagaciono kašnjenje.



Slika 9. Radno okruženje. Maksimalno kašnjenje kombinacionih kola.