



Radionica digitalnog dizajna I Dan II

dr. sc. Jurica Kandrata

Centar za svemirsku i inovativnu tehnologiju

6. - 7. svibnja 2023.

Raspored

6. svibnja 2023., subota	
9.00 - 12.00	Uvod u Verilog jezik, testne postavke i simulaciju digitalnih sklopova
12.00 - 12.45	Pauza za ručak
12.45 - 15.00	Dizajn kombinacijskih i sekvencijalnih logičkih sklopova
7. svibnja 2023., nedjelja	
9.00 -12.00	Radni tijek s FPGA platformom
12.00 - 12.45	Pauza za ručak
12.45 - 15.00	Implementacija projekta na FPGA razvojnoj ploči

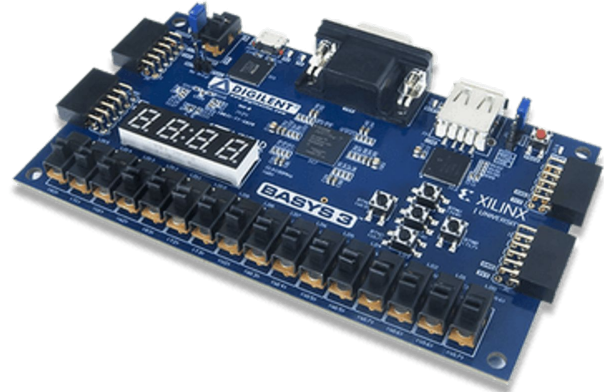
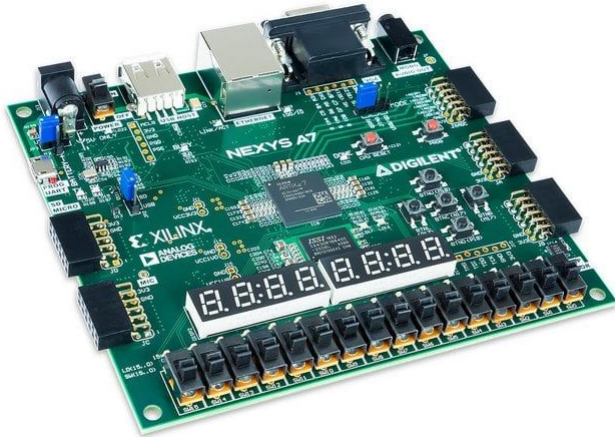
Dan II

Sadržaj

- Tok rada u Vivado-u
- FPGA projekt - Sat

Vivado

- Vivado je sveobuhvatan alat za dizajn FPGA tvrtke Xilinx.
- Nudi širok raspon značajki i mogućnosti za projektiranje, provjeru i programiranje Xilinx FPGA uređaja.



Tok rada u Vivado-u (I)

1. Napravite novi projekt:

- Odaberite odgovarajući FPGA uređaj (broj dijela) za svoj dizajn.
- Odredite direktorij projekta i naziv projekta.
- Odaberite izvore dizajna (Verilog, VHDL ili mješoviti) i datoteke ograničenja (XDC).

Tok rada u Vivado-u (II)

2. Unos dizajna:

- Napišite RTL (Register Transfer Level) kod za svoj dizajn koristeći HDL (Verilog, VHDL ili mješoviti).
- Napravite modul najviše razine koji povezuje sve podmodule.

3. Dodajte ograničenja:

- Napišite XDC datoteku da navedete dodjele I/O pinova, razine napona i vremenska ograničenja.

Tok rada u Vivado-u (III)

4. RTL sinteza:

- Pokrenite sintezu za generiranje spojne liste na razini vrata iz RTL koda.
- Provjerite upozorenja sinteze, pogreške i izvješća o vremenu.

5. RTL simulacija (izborna):

- Napišite testove za svoje module dizajna i pokrenite simulacije ponašanja kako biste provjerili njihovu funkcionalnost.
- Popravite sve probleme pronađene tijekom simulacije.

Tok rada u Vivado-u (IV)

6. Implementacija:

- Pokrenite proces implementacije, koji uključuje prijevod, mapiranje, postavljanje i spajanje.
- Provjerite izvješća o implementaciji za vrijeme i korištenje resursa.

7. Simulacija nakon implementacije (izborno):

- Pokrenite simulacije nakon implementacije kako biste provjerili funkcionalnost dizajna nakon procesa implementacije.
- Popravite sve probleme pronađene tijekom simulacije.

Tok rada u Vivado-u (IV)

8. Generiranje bitstreama:

- Generirajte bitstream datoteku (.bit) za programiranje FPGA uređaja.

9. Programirajte FPGA:

- Spojite FPGA razvojnu ploču na svoje računalo.
- Koristite Vivado Hardware Manager za programiranje FPGA s generiranom bitstream datotekom.

Tok rada u Vivado-u (V)

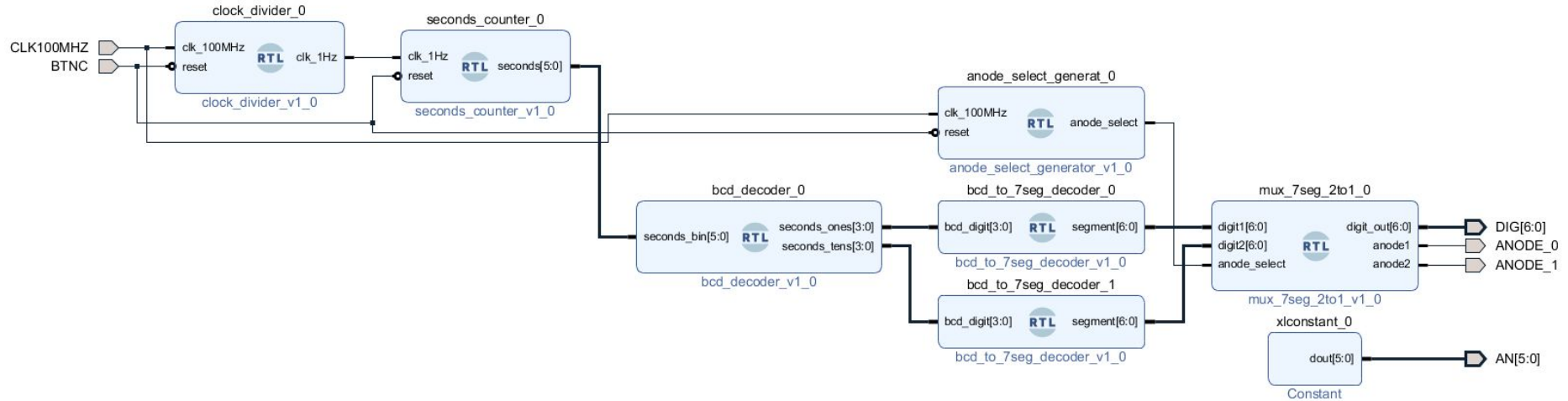
10. Provjerite dizajn na FPGA:

- Testirajte funkcionalnost svog dizajna na FPGA razvojnoj ploči.
- Otklonite pogreške i napravite promjene u dizajnu prema potrebi.

FPGA projekt - Sat

- Jednostavni digitalni sat koji prikazuje sekunde na zaslonu od sedam segmenata
- Moduli digitalnog sustava:
 - Dijelilo takta 100 MHz na 1 Hz
 - Brojilo sekundi
 - BCD enkoder
 - BCD u 7 segmentni dekoder
 - Multiplekser za znamenke
 - Generator selektora anode za 7 segmentni prikaz

FPGA projekt - Sat - Blok diagram

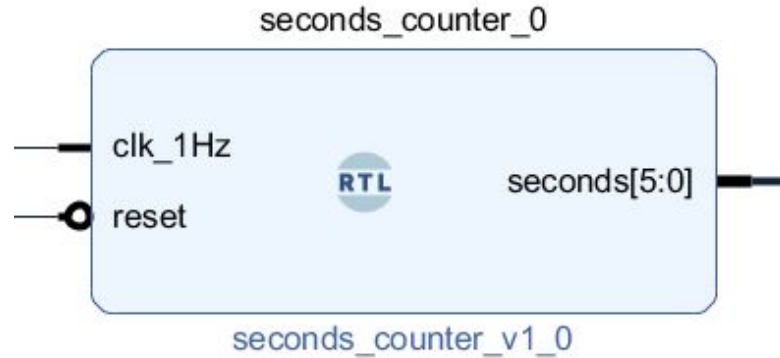


FPGA projekt - Sat - Dijelilo takta 100 MHz na 1 Hz



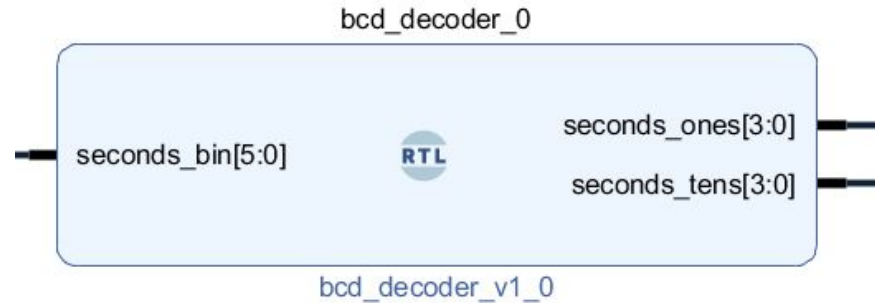
- Dijeli ulazni signal takta od 100 MHz i na izlazu daje podijeljeni signal takta od 1 Hz
- Podržava resetiranje s visokom aktivnom razinom

FPGA projekt - Sat - Brojač sekundi



- Broji sekunde od 00 do 59 i nakon toga se premota
- Podržava reset visoke aktivne razine
- Na izlazu daje 6-bitni binarni broj koji predstavlja broj sekundi

FPGA projekt - Sat - BCD enkoder



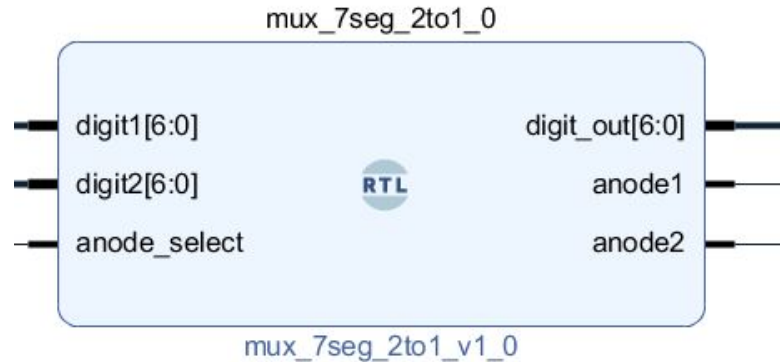
- Broj sekundi u binarnom obliku enkodira u BCD format, binarno kodirani decimalni format
- Na izlazu daje dvije BCD znamenke

FPGA projekt - Sat - BCD u 7 segmentni dekodier



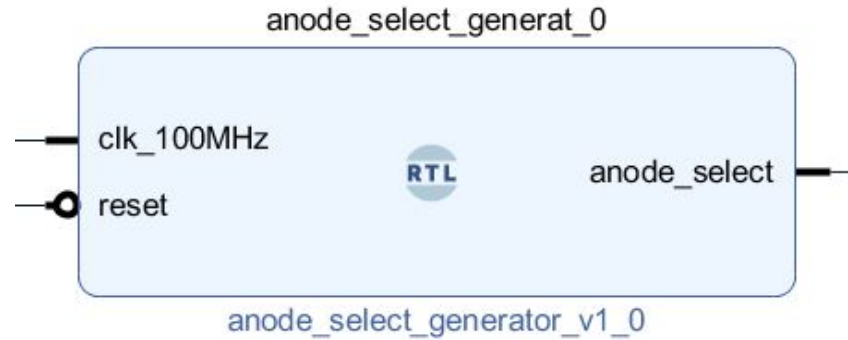
- Znamenku kodiranu u BCD formatu dekodira u format za prikaz na 7 segmentnom zaslonu
- Ulaz je 4-bitni, a izlaz je 7-bitni podatak

FPGA projekt - Sat - multiplekser za 7 segmentni prikaz



- Temeljem izbora znamenke **anode_select** prosljeđuje znamenku na izlaz i aktivira odgovarajuću anodu znamenke

FPGA projekt - Sat - multiplexer za 7 segmentni prikaz



- Dijelilo frekvencije sa 100 MHz na 1 kHz
- Izlazni takt predstavlja izbor znamenke:
 - 0 → znamenka jedinica
 - 1 → znamenka desetica

Literatura

- <https://www.chipverify.com/verilog/verilog-tutorial>
- <https://www.doulos.com/knowhow/verilog/>
- <https://fpgacademy.org/courses.html>
- https://hdlbits.01xz.net/wiki/Main_Page

Ekstra

FPGA projekt I - Treptajući LED s upravljanjem

- Treptajuća LED dioda s podesivom brzinom treptanja
- **Razdjelnik takta:** Vaša FPGA razvojna ploča imaće ugrađeni izvor takta (npr. 50 MHz ili 100 MHz). Morat ćete izraditi modul razdjelnika sata za generiranje sporijeg signala sata za kontrolu brzine treptanja LED-a.
- **Kontrola brzine treptanja:** Dizajnirajte modul za korištenje ugrađenih prekidača za odabir željene brzine treptanja (npr. sporo, srednje, brzo). Ovaj modul će poslati signal odabira na temelju položaja prekidača.
- **LED blinker:** Napravite modul LED blinkera koji kao ulaz uzima podijeljeni sat i odabir brzine treptanja. Ovaj modul će uključiti LED na željenu brzinu treptanja na temelju odabira.
- **Gornji modul:** Stvorite modul najviše razine koji instancira i povezuje sve komponente (razdjelnik sata, kontrola brzine treptanja i LED blinker).

FPGA projekt I - Treptajući LED s upravljanjem

```
module blinking_led (  
    input clk,                // Onboard clock source  
    input reset,              // Reset signal  
    input [1:0] blink_rate_sw, // Switches for blink rate selection  
    output led                // LED output  
);  
    // Instantiate clock divider, blink rate control, and LED  
    blinker  
    // Connect the components as required  
  
endmodule
```