

Digitalno načrtovanje

Generični VHDL moduli in povezovanje
Simulator

Izvajalca: prof. dr. Patricio Bulić in

doc. dr. Nejc Ilc

Asistent: Ratko Pilipović

Generic

- generic spremenljivke uporabljamo za gradnjo splošnih gradnikov, npr. n-bitni števec
- ob deklaraciji navedemo privzeto vrednost.

entity counter is

Port (

data : in STD_LOGIC_VECTOR (3 downto 0);

clock : in STD_LOGIC;

reset : in STD_LOGIC

);

end counter ;

Generic

- generic spremenljivke uporabljamo za gradnjo splošnih gradnikov, npr. n-bitni števec
- ob deklaraciji navedemo privzeto vrednost.

entity counter is

Generic (data_width : integer := 8);

Port (

data : in STD_LOGIC_VECTOR (data_width -1 downto 0);

clock : in STD_LOGIC;

reset : in STD_LOGIC

);

end counter ;

Povezovanje gradnikov

- Želimo bolj pregledno strukturo gradnikov in večkratno uporabo že zgrajenih gradnikov
 - npr. "kodo" za 4-bitni števec lahko uporabimo tudi za 5-bitni števec
- Primer: Števec iz zadnje vaje razdelimo v 2 manjša gradnika
 - prescaler
 - vhod: clock, reset, limit
 - izhod: enable
 - števec
 - vhod: clock, reset, enable
 - izhod: data

Prescaler

entity prescaler is

Generic (

max_width : integer := 8;

);

Port (

clock: in STD_LOGIC;

reset: in STD_LOGIC;

enable: out STD_LOGIC;

limit: in STD_LOGIC_VECTOR(max_width-1 downto 0);

);

end prescaler ;

Counter

entity counter is

Generic (width : integer := 8);

Port (

clock: in STD_LOGIC;

reset: in STD_LOGIC;

enable: in STD_LOGIC;

data: out STD_LOGIC_VECTOR(width-1 downto 0)

);

end counter ;

Zunanji modul

- zunanji modul ne more biti generični

entity top is

Port (

clock: in STD_LOGIC;

reset: in STD_LOGIC;

data: out STD_LOGIC_VECTOR(3 downto 0);

);

end top ;

Povezovanje

- Gradnik, ki ga želimo uporabiti znotraj drugega gradnika, imenujemo komponenta

Deklaracija:

```
component ime_komponente  
  port ( ime: smer tip_signala ...);  
end component;
```

Povezovanje:

oznaka: ime_komponente

```
port map (  
  ime_signala_kom => ime_signala_vezje,  
  ime_signala_kom2 => ime_signala_vezje2,  
  ...  
);
```


Povezovanje

architecture Behavioral **of** top **is**

component prescaler

Generic (max_width : integer := 8;);

Port (

clock: in STD_LOGIC;

reset: in STD_LOGIC;

enable: out STD_LOGIC;

);

end component;

component counter

Generic (width : integer := 8);

Port (

clock: in STD_LOGIC;

reset: in STD_LOGIC;

enable: in STD_LOGIC;

data: out STD_LOGIC_VECTOR (width-1 downto 0);

);

end component;

--notranji signal, ki ga uporabimo za povezovanje obeh komponent

signal enable: STD_LOGIC;

begin

Povezovanje

pr : prescaler

generic map (max_width => 28)

port map (

clock => clock,

reset => reset,

enable => enable,

limit => X"5F5E0FF"

);

- signali na desni strani so del top modula, signali na levi pa del modula, ki ga povezujemo

Povezovanje

cnt : counter

generic map

(
width => 4
)

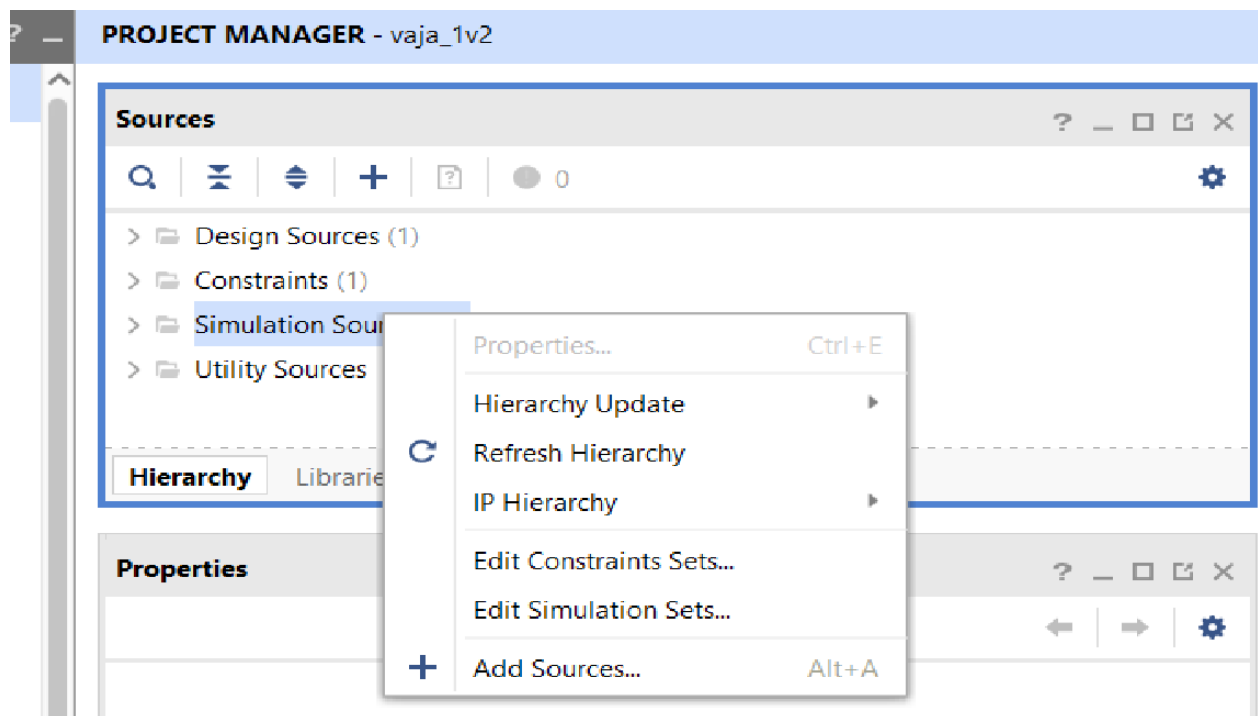
port map

(
clock => clock,
reset => reset,
enable => enable,
data => data
);

end Behavioral;

Simulator

- Dodamo fajl za simulacijo ali test bench



Simulator

- Izberemo modul, ki ga želimo testirati
- Napišemo test bench modul, ki vsebuje izbrani modul
- Definiramo vrednosti na vhodu in spremembe skozi čas

Test bench

- Ločen proces za generiranje ure

```
-- Instantiate the Unit Under Test (UUT)
 uut: counter PORT MAP (
     clk => clk,
     rst => rst,
     count_out => count_out
 );

-- Clock process definitions
clk_process :process
begin
    clk <= '0';
    wait for clk_period/2;
    clk <= '1';
    wait for clk_period/2;
end process;
```

Test bench

- Ostale vhodne vrednosti definiramo v ločenem procesu

- prireditveni stavki in wait ukazi

- `wait for 100 ns;`

- `wait for clk_period*10;`

- clk_period je preddefiniran konstanta

- `wait;`

- čaka nedoločeno časa

Test bench

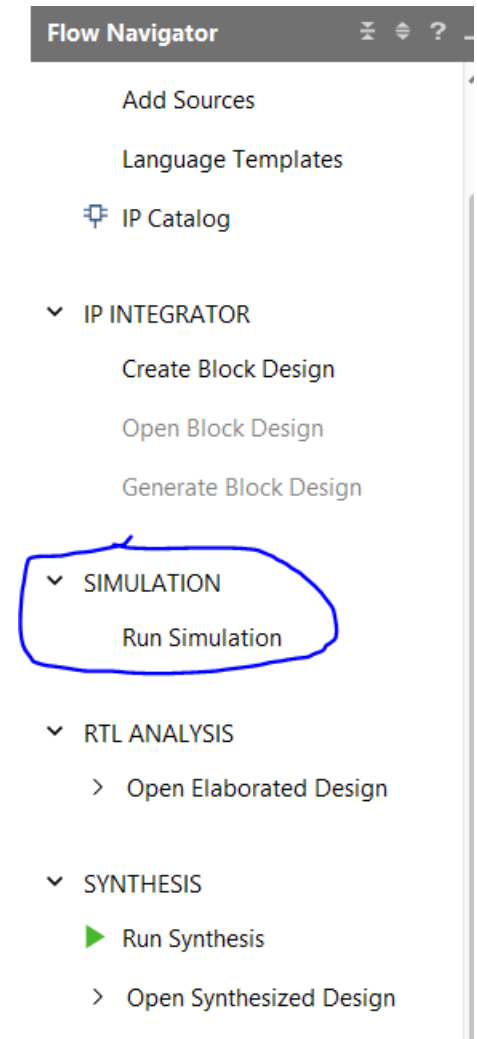
```
-- Stimulus process
stim_proc: process
begin
    rst <= '1';
    wait for 100 ns;
    -- hold reset state for 100 ns.
    rst <= '0';
    wait for clk_period*10;

    -- insert stimulus here

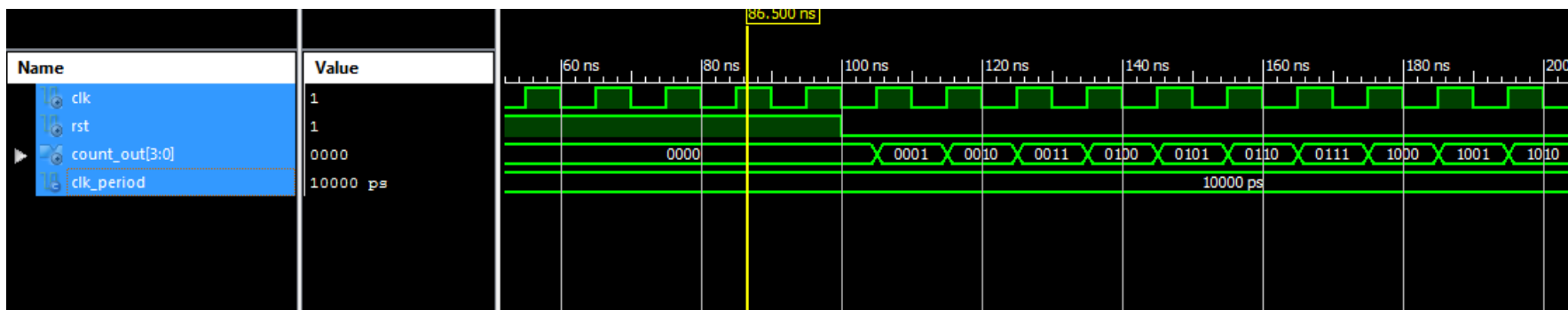
    wait;
end process;
```


Uporaba simulatorja

- Izberi modul za test
- Poženi



Simulacija



Naloga

- Dokončajte projekt iz vaje:
 - dodajte modul za prescaler in dokončajte sekundni števec
 - zamenjajte module „counter“ s „shifter“ modulom, ki bo upravljal LED diode kot v nalogi prejšnjega tedna