

Digitalno načrtovanje

2015/16

Izvajalec: prof. dr. Patricio Bulić

Asistent: Rok Češnovar

Generic

Generic spremenljivke uporabljamo za gradnjo bolj splošnih gradnikov.

Ob deklaraciji navedemo privzeto vrednost.

```
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity stevec is
```

```
  Generic ( data_width : integer := 8 );
```

```
  Port (
```

```
    enable_i : in STD_LOGIC; clk_i : in STD_LOGIC;
```

```
    rst_i : in STD_LOGIC;
```

```
    data_o : out STD_LOGIC_VECTOR ( data_width -1 downto 0)
```

```
  );
```

```
end stevec ;
```

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: clk_i, reset_i
 - izhod: enable_o
 - števec
 - vhod: clk_i, reset_i, enable_i
 - izhod: data_o

Prescaler

entity prescaler is

Generic (

width : integer := 8;

value : integer := 255

);

Port (

clk_i: in STD_LOGIC; reset_i: in STD_LOGIC;

enable_o: out STD_LOGIC

);

end prescaler ;

Counter

entity counter is

Generic (

width : integer := 8

);

Port (

clk_i: in **STD_LOGIC**; reset_i: in **STD_LOGIC**;

enable_i: in **STD_LOGIC**;

data_o: in **STD_LOGIC_VECTOR**(width-1 downto 0)

);

end counter ;

Zunanji modul

```
entity TopModul is
```

```
  Port (
```

```
    clk_i: in STD_LOGIC; reset_i: in STD_LOGIC;
```

```
    data_o: out STD_LOGIC_VECTOR(3 downto 0);
```

```
  );
```

```
end TopModul ;
```

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 TopModul is

component prescaler

```
Generic ( width : integer := 8; value : integer :=255 );  
Port ( clk_i: in STD_LOGIC; reset_i: in STD_LOGIC;  
enable_o: out STD_LOGIC;  
);
```

end component;

component counter

```
Generic ( width : integer := 8 );  
Port ( clk_i: in STD_LOGIC; reset_i: in STD_LOGIC;  
enable_i: in STD_LOGIC;  
data_o: 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 (

width => 12,
value => 4000

)

port map (

clk_i => clk_i,
reset_i => reset_i,
enable_o => enable

);

Povezovanje

cnt : counter

generic map

(

width => 4

)

port map

(

clk_i => clk_i,

reset_i => reset_i,

enable_i => enable,

data_o => data_o

);

end Behavioral;