Univerza v Ljubljani
Fakulteta
za računalništvo
in informatiko

Metode logičnega snovanja

Modularno opisovanje vezij v jeziku VHDL

Miha Moškon

Opisovanje delovanja vezij

V arhitekturi VHDL programa lahko opišemo delovanje vezja na tri možne načine:

- vzporedno (glej 1. in 2. vajo),
- zaporedno (glej 3. in 4. vajo),
- modularno.

Modularno opisovanje vezij

Uporablja se, kadar isti modul nastopa večkrat.

Omogoča nam uporabo že napisanih modulov (MUX, seštevalnik, register, števec, pomnilnik ...) in bolj pregledno programiranje pri večjih projektih (razvoj procesorja lahko razbijemo na ALE, podatkovno enoto, kontrolno enoto in pomnilnik).

V posameznem modulu lahko uporabimo spremenljivke **`generic`**, ki nam omogočajo večjo prilagodljivost (npr. določanje velikosti registra šele ob uporabi).

Modularno opisovanje vezij (definicija modulov)

Posamezen modul definiramo na enak način, kot smo to delali do sedaj (**entiteta + arhitektura**).

Če želimo izbrane spremenljivke v modulu določiti šele ob uporabi, v eniteto vključimo **`generic`** del:

```
entity register is
  generic(
     size : INTEGER := 8); -- privzeta velikost je 8

  port(
     data_in : IN STD_LOGIC_VECTOR(size-1 downto 0);
     data_out : OUT STD_LOGIC_VECTOR(size-1 downto 0);
     .... );
```

Modularno opisovanje vezij (uporaba modulov)

V arhitekturo dodamo **component** , **port map** in **generic map** del:

```
architecture behavioral of primer is
component komponenta
    -- prekopiramo iz entitete modula
    generic
    (size : INTEGER := 8);
    port
    (
    CLK : IN STD_LOGIC;
    data_in : IN STD_LOGIC_VECTOR(size-1 downto 0);
    data_out : OUT STD_LOGIC_VECTOR(size-1 downto 0);
    ...
    );
end component;
```

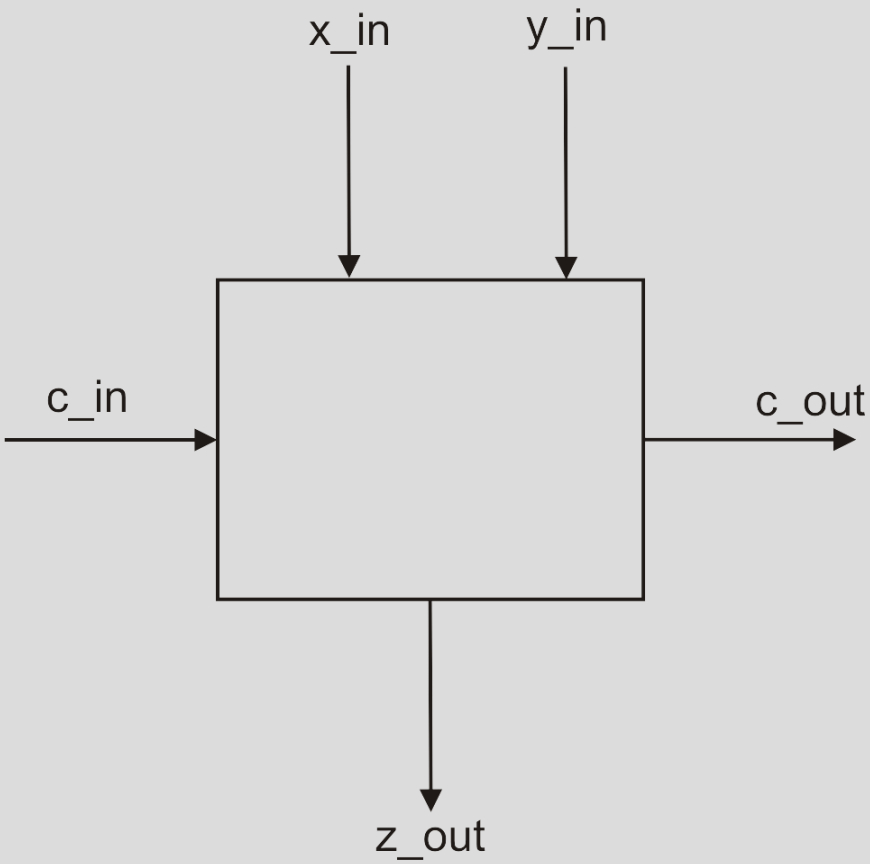
Modularno opisovanje vezij (uporaba modulov) (2)

```
begin
        komp1:komponenta
        generic map
        (
                size => 6 -- velikost registra
        )
        port map
        (
                CLK => CLK,
                data_in => lokalni_data_in,
                data_out => lokalni_data_out,
                ...
        );
        komp2:komponenta
        port map
        (
                ...
        );
end behavioral;
```

Naloga

1. Napišite program za 4-bitni seštevalnik. Pri tem implementirajte modul1-bitnega seštevalnika (ločena vhd datoteka) in ga uporabite pri gradnji 4-bitnega.
2. Seštevalnik dopolnite tako, da bo rezultate seštevanja shranjeval v register. Pri tem implementirajte modul `register`.

1-bitni seštevalnik



Register