Univerza v Ljubljani
Fakulteta
za računalništvo
in informatiko

Metode logičnega snovanja
VHDL, vzporedno opisovanje (2. del)

Miha Moškon

Vporedno opisovanje vezij

Vsi stavki se izvajajo istočasno – vzporedno.

Trije različni načini:

- z logičnimi enačbami,
- s stavkom `when-else`,
- s stavkom `with-select`.

with-select

```
with ime_signala1 select
        ime_signala2 <= izrazS1 when izrazP1,
                        izrazS2 when izraz P2,
                        ...
                        izrazSN when others;
```

- primer:

```
with vhod1 select
  izhod2      <=  vhod2 when “00”,
                  vhod3 when “11” | “10”,
                  ‘1’ when others;
```

Primer programa MUX 2/1

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity MUX is
   port (I: in std_logic_vector(1 downto 0); -- podat. vhoda
         A: in std_logic; -- adresni vhod
         O: out std_logic); -- izhod MUX-a
end MUX;

architecture behavioral of MUX is
begin
   -- jedro programa (glej naslednjo prosojnico)
end behavioral;
```

Primer programa MUX 2/1 (2)

- s stavkom `with-select`:

```
with A select
        O <= I(0) when '0',
             I(1) when others;
```

Sinteza (končno vezje) je enaka kot pri opisu z logično enačbo oziroma stavkom `when-else`.

Signali

- Z njimi definiramo notranje povezave med elementi v čipu.
- Če je način delovanja priključka `out`, ga ne moremo uporabiti na desni strani prireditvenega stavka. V takem primeru uporabljamo signale, ki jih priredimo izhodnemu signalu.
- Definiramo jih v arhitekturi pred `begin`.

Signali (primer)

```
architecture behavioral of primer is
    -- definicije signalov
    signal primer1: std_logic_vector(2 downto 0);
    signal primer2: std_logic;
begin
```

Lastni podatkovni tipi

- Deklaracija tipa:

```
type ime_tipa is (vrednost1, vrednost2,...);
```

- Uporaba tipa:

```
signal ime_signala: ime_tipa;
```

Konstante

- Deklariramo jih lahko kjerkoli.

```
constant ime_konstante: tip := vrednost;
```

- Primer

```
constant ena: std_logic := '1';
```

Delo s števili

- Vključimo knjižnici `unsigned` in `arith`.

```
library IEEE;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
ali
use IEEE.STD_LOGIC_SIGNED.ALL;
```

- Glede na uporabljeno knjižnico (`SIGNED` ali `UNSIGNED`) se intepretira vrednost vektorjev.

- Ne moremo uporabiti obeh hkrati!

Operator združevanja

- Operator združevanja ali konkatenacija:

  &

- Primer

```
   signal vektor1: std_logic_vector(3 downto 0);
   signal vektor2: std_logic_vector(2 downto 0);
   signal signal1: std_logic;
...
   vektor1 <= vektor2 & signal1;
   vektor2 <= '0' & "11";
   signal1 <= '0';
```

Lastni podatkovni tipi

- Deklaracija tipa:

```
type ime_tipa is (vrednost1, vrednost2,...);
```

- Uporaba tipa:

```
signal ime_signala: ime_tipa;
```

Simuliranje (ročno)

- iz tbw datoteke se ustvari vhd datoteka, ki jo uporablja simulator
- ogled zgenerirane vhd kode:
  - Sources for (Behavioral Simulation) → x.tbw
  - Processes → View Generated Test Bench As HDL
- neskončna zanka, v kateri s pomočjo stavka `wait for x ns` definiramo vrednosti posameznega signala

Simuliranje (ročno)

Ročno generiranje testnih signalov (z vhd datoteko):

- Project → New Source → VHDL Test Bench
- določanje poteka signalov s pomočjo stavka `wait for`
- izpis poteka simulacije – izberemo:
  - Sources for: Behavioral simulation (levo zgoraj), kjer mora biti izbrana naša datoteka s testnimi signali,
  - Processes (levo spodaj),
  - Simulate Behavioral Model (levo na sredini).

Naloge

1) Realizirajte 4/1 podatkovni izbiralnik (multipleksor).

   Nalogo rešite na dva načina, in sicer z uporabo logičnih enačb in z uporabo konstrukta `with-select`.

   Pravilnost delovanje rešitve preverite s simulacijo, pri čemer poskusite ročno napisati tesno vhd datoteko.

Naloge

2) Realizirajte 2-bitni seštevalnik, ki kot vhod sprejme dve 2-bitni števili ter vrne njuno vsoto v obliki 2-bitnega števila in prenosa.

   Nalogo rešite na dva načina, in sicer z uporabo logičnih enačb in z uporabo knjižnice `numeric_std`. Vhodi in izhodi iz vezja naj bodo v obeh primerih tipa `std_logic` oziroma `std_logic_vector`.