Prosojnice

Monday, October 25, 2021 9:58 AM

02_Uvod - 2.del

Digitalno načrtovanje

Izvajalca: prof. dr. Patricio Bulić
in doc. dr. Nejc Ilc
Asistent: Ratko Pilipović

Proces

!?!?

```
<ime_procesa>: process (<sensitivity_list>)
begin
--koda procesa
end
```

- v <sensitivity_list> navedemo vse signale, ki bodo povzročili spremembe na izhodih iz procesa

if stavek

- Uporaben samo znotraj procesa

```
if pogoj then
        --prireditve
else
        --prireditve
end if;
```

```
if pogoj then
        --prireditve
elsif pogoj then
        --prireditve
else
        --prireditve
end if;
```

case stavek

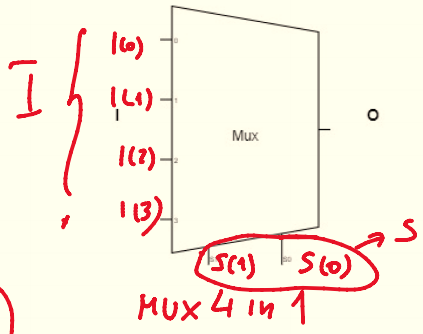
```
case s is
when vred1 => izhod <= izraz1;
when vred2 => izhod <= izraz2;
when vred3 => izhod <= izraz3;
when others => izhod <= izraz4;
end case;
```

- zadnji (default) člen je obvezen

Primer

- 4/1 mux

```
process(i,s)
begin
        case s is
            when "00" => o <= i(0);
            when "01" => o <= i(1);
            when "10" => o <= i(2);
            when "11" => o <= i(3);
            when others => o <= i(0);
        end case;
end process;
```

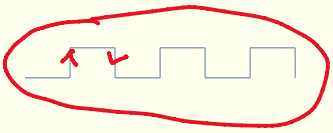


Sekvenčna vezja

- V sekvenčnih vezjih se spremembe dogajajo ob dogodkih ure (prvi/zadnji fronti)
- Dogodek na signalu ure zaznamo z clk'event
  - clk je signal za uro
  - 'event je true, ko je prišlo do spremembe signala
- Prva fronta

  clk'event and clk = '1'
- Zadnja fronta

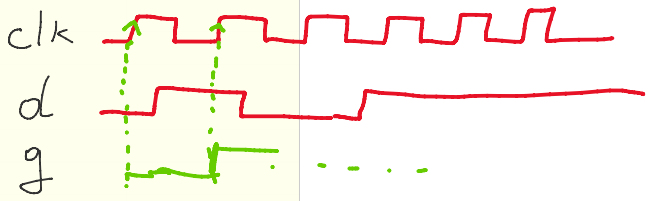
  clk'event and clk = '0'



Primer – D celica

```
process(clk)
begin
if clk'event and clk = '1' then
        q <= d ;
end if ;
end process ;
```

CLK ENOTRAJ. SENS. LIST



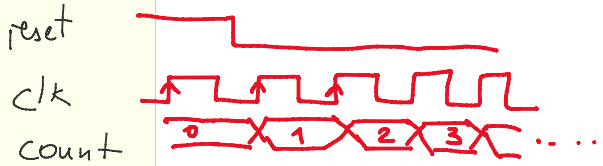
Primer - števec

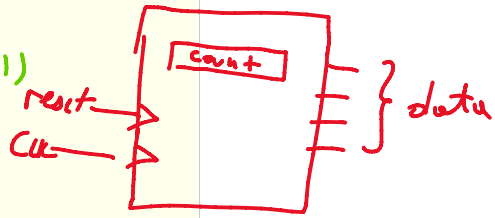
```
process(clk)
begin
if clk'event and clk = '1' then
        if(rst= '1 ') then
                --(others => '0') resetira vrednost na 0
                count <= (others => '0');
        else
                count <= count + 1;
        end if;
end if ;
end process ;
```

COUNT UP COUNT DOWN

(others => '1')

count - 1





VHDL knjižnice

- Če ste bili pozorni ste v VHDL kodi opazili:

library IEEE;
use IEEE. STD_LOGIC_1164. ALL;

- knjižnica IEEE. STD_LOGIC_1164 definira tipa STD_LOGIC in STD_LOGIC_VECTOR ter nekaj uporabnih funkcij, naprimer:
  - rising_edge (namesto clk'event and clk = '1')
  - falling_edge (namesto clk'event and clk = '0')

IEEE.NUMERIC_STD

use IEEE.NUMERIC_STD.ALL

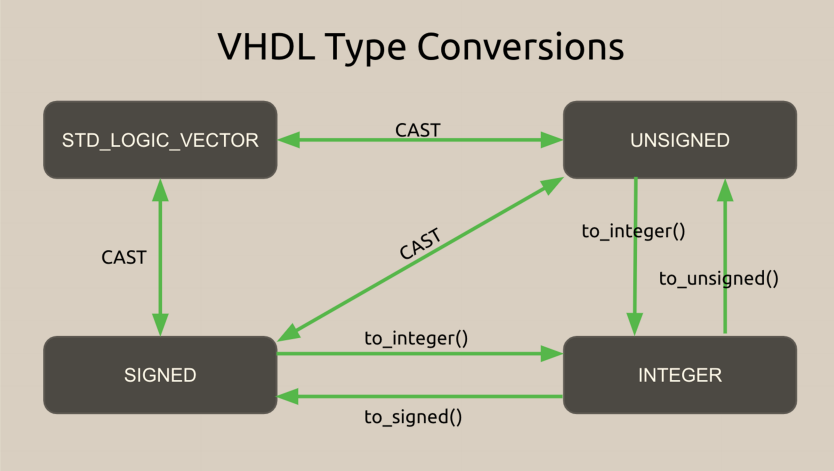
- Definira aritmetične operacije in tipove unsigned in signed!!!
- definira naslednje operacije nad unsigned in signed!!!!:
  - +,-,*
  - <, <=, >, >=, =, /=
  - shift_left(OP1,OP2), shift_right(OP1,OP2)
    - (OP1 << OP2), OP1 (un)signed, OP2 integer

IEEE.NUMERIC_STD

use IEEE.NUMERIC_STD.ALL

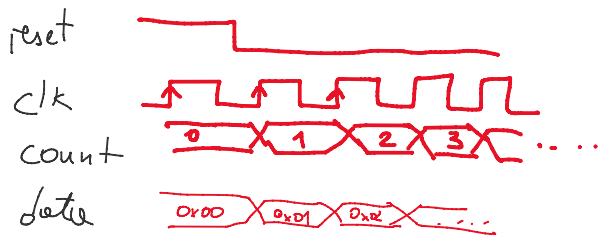
- Pri uporabi aritmetičnih operacij rezultat ali operande kastovati v/iz signed ali unsigned tipa
- F-je za kastovanje
  - (un)signed(ime_vektora)
  - std_logic_vector(ime_vektora)
  - to_integer(), to_signed(), to_unsigned()

Konverzija tipov



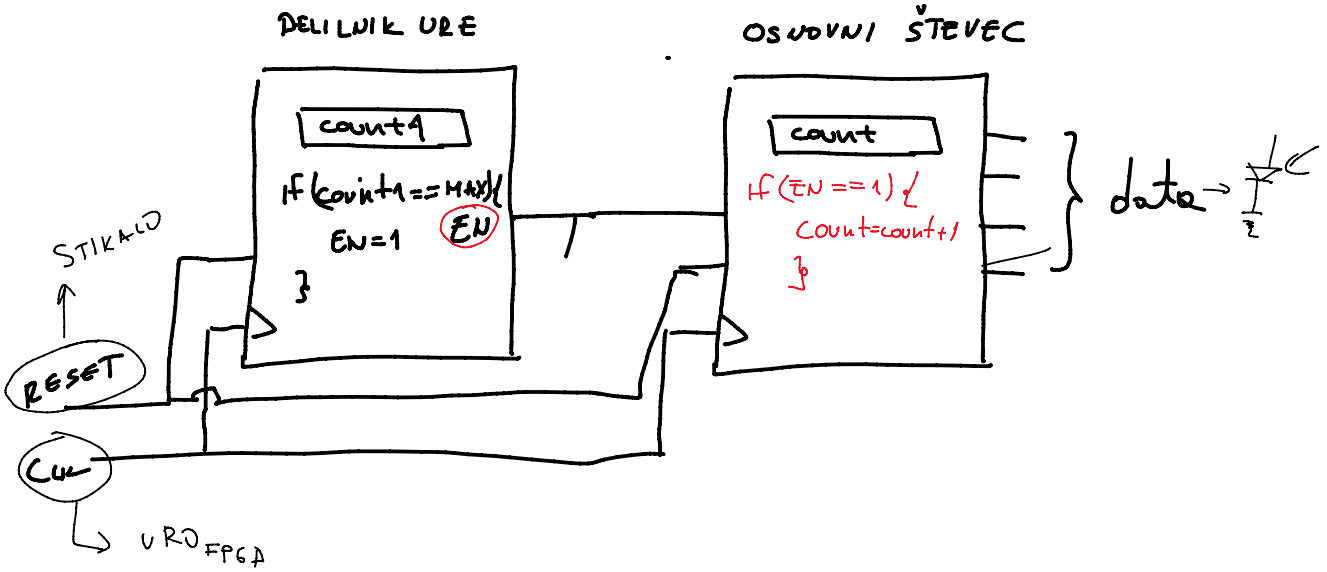
Naloga

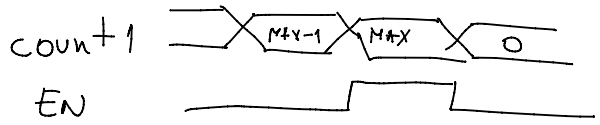
- Napišite VHDL modul, ki bo ustvaril vzorec „pomikanja“ na LED (pol-sekundni interval)



$f = 100\ MHz$

$T = \frac{1}{100 \cdot 10^6}$




count+1
MAX-1
MAX
0
EN