Digitalno načrtovanje

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

Potek vaj

- opisovanje strojne opreme z jezikom VHDL
- izgradnja sistema na čipu
  - osnovno delo s stikali, gumbi in LED
  - izgradnja krmilnikov VGA, PS2, UART, …
  - vključitev in povezovanje PicoBlaze CPE
  - …
- Pogoj za opravljene vaje
  - seminar

VHDL

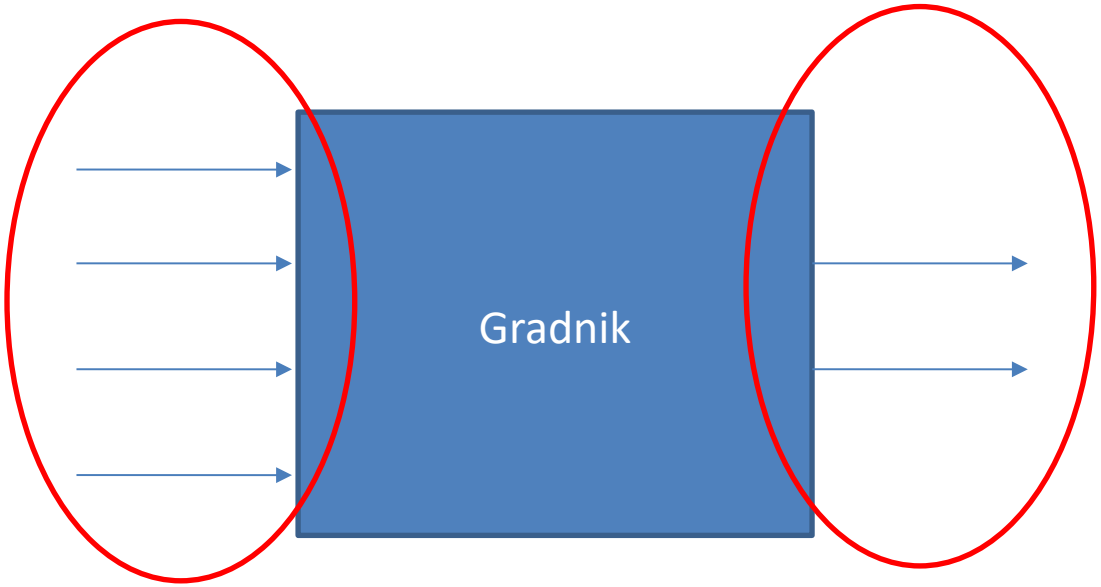
- VHSIC Hardware Description Language
  - VHSIC = very-high-speed integrated circuits
- Jezik za opisovanje strojne opreme/digitalnega vezja ter modeliranje/simulacijo vezij
- VHDL opis bomo sintetizirali in programirali programabilno logično vezje (FPGA)

Oprema

- Programska oprema
  - Xillinx Vivado (Povezava na e-učilnici)
- Strojna oprema
  - Razvojna plošča Digilent Nexys4
  - Razovjna plošča Digilent Nexys4DDR
  - Razovjna plošča Digilent Nexys A7
    - 50T
    - 100T

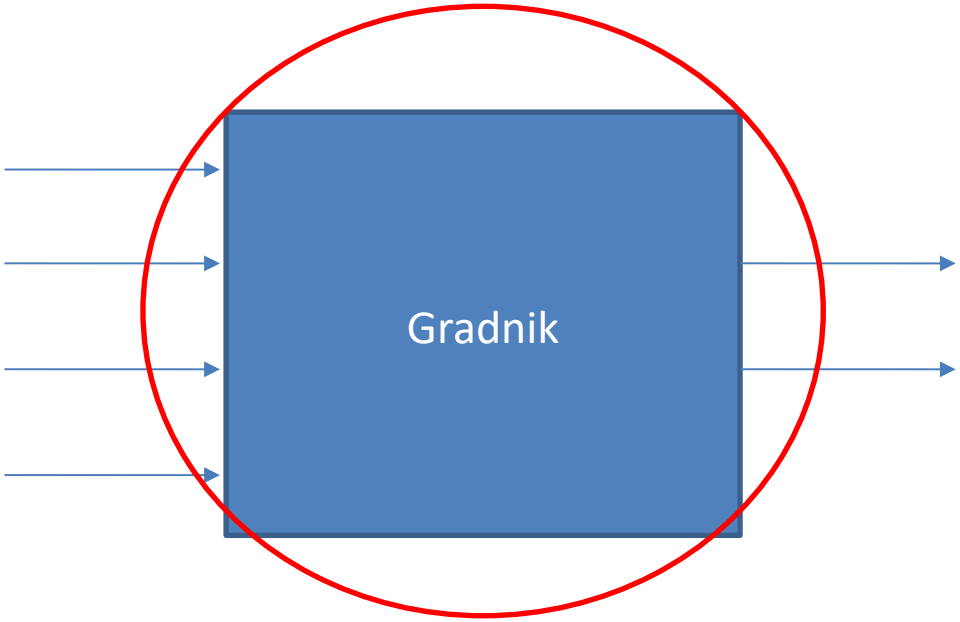
Opis osnovnega gradnika

Opis osnovnega gradnika



Definiramo zunanje signale gradnika

Opis osnovnega gradnika



Definiramo delovanje gradnika – kaj "počne" s signali

Opis vhoda/izhoda - primer


```
entity ime_gradnika is
        port(
                a: in std_logic;
                b: out std_logic
               );
end ime_gradnika;
```

Opis zunanjih signalov

```
entity ime_vezja is
port (
      ime_signala : smer tip_signala;
      ime_signala_2 : smer tip_signala;
      …
      ime_signala_n: smer tip_signala
      );
end ime_vezja;
```

- Smer: in, out, inout
- Tip signala: std_logic, std_logic_vector()

Opis vhoda/izhoda – primer 2

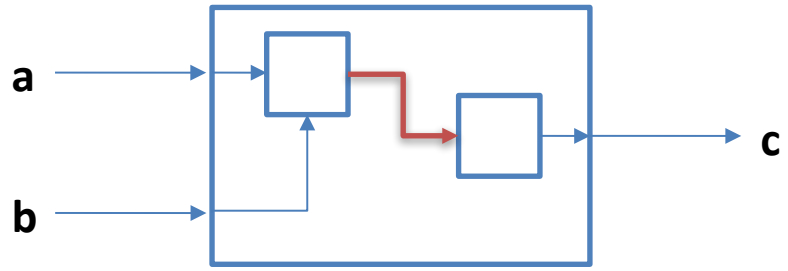

```
entity ime_gradnika is
        port(
                a: in std_logic_vector(7 downto 0);
                b: in std_logic_vector(0 to 3);
                c: in std_logic_vector(7 downto 0);
               );
end ime_gradnika;
```

Opis delovanja (logike) gradnika

```
architecture opis_vezja of ime_gradnika is
//deklaracija notranjih signalov
begin
    //stavki za opis vezja
end opis_vezja;
```

Deklaracija notranjih signalov

```
architecture Behavioral of ime_vezja is
    signal ime_signala: tip_signala;
begin …
```



Prireditveni stavek

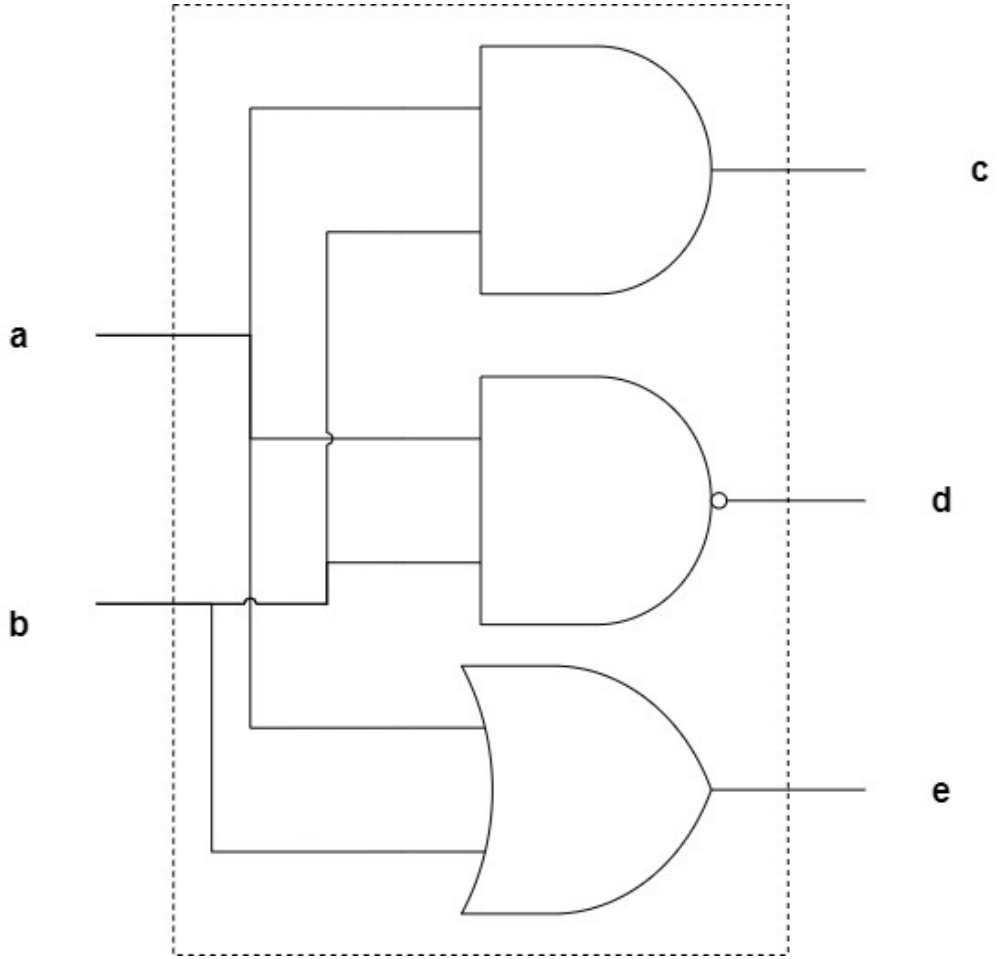
Sintaksa:

signal <= izraz;

Primeri:

```
a <= '0'; //prireditev konstante
b <= "01001"; //prireditev konstante  za vektor
c(3 downto 0) <= "0111"; //prireditev delu vektorja
```

Primer



Primer – dvovhodna logična vrata

```
entity logicna_vrata is
    port(
    a: in std_logic;
    b: in std_logic;
    c: out std_logic;
    d: out std_logic;
    e: out std_logic
    );
end logicna_vrata ;
```

Opis parametrov in signalov

- Uporabe osnovni logičnih operatorjev
  - and, nand, or, nor, xor, xnor, not

```
architecture Behavioral of logicna_vrata is
begin
//stavki za opis vezja
c <= a and b;
d <= a nand b;
e <= a or b;
end Behavioral ;
```

Pogojni prireditveni stavek

```
signal <= izraz1 when pogoj1 else izraz2;
```

```
signal <= izraz1 when pogoj1 else
          izraz2 when pogoj2 else
          izraz3;
```

Pogojni operatorji:

- enako, ni enako =, /=
- večje, manjše,… >,<,>=,<=

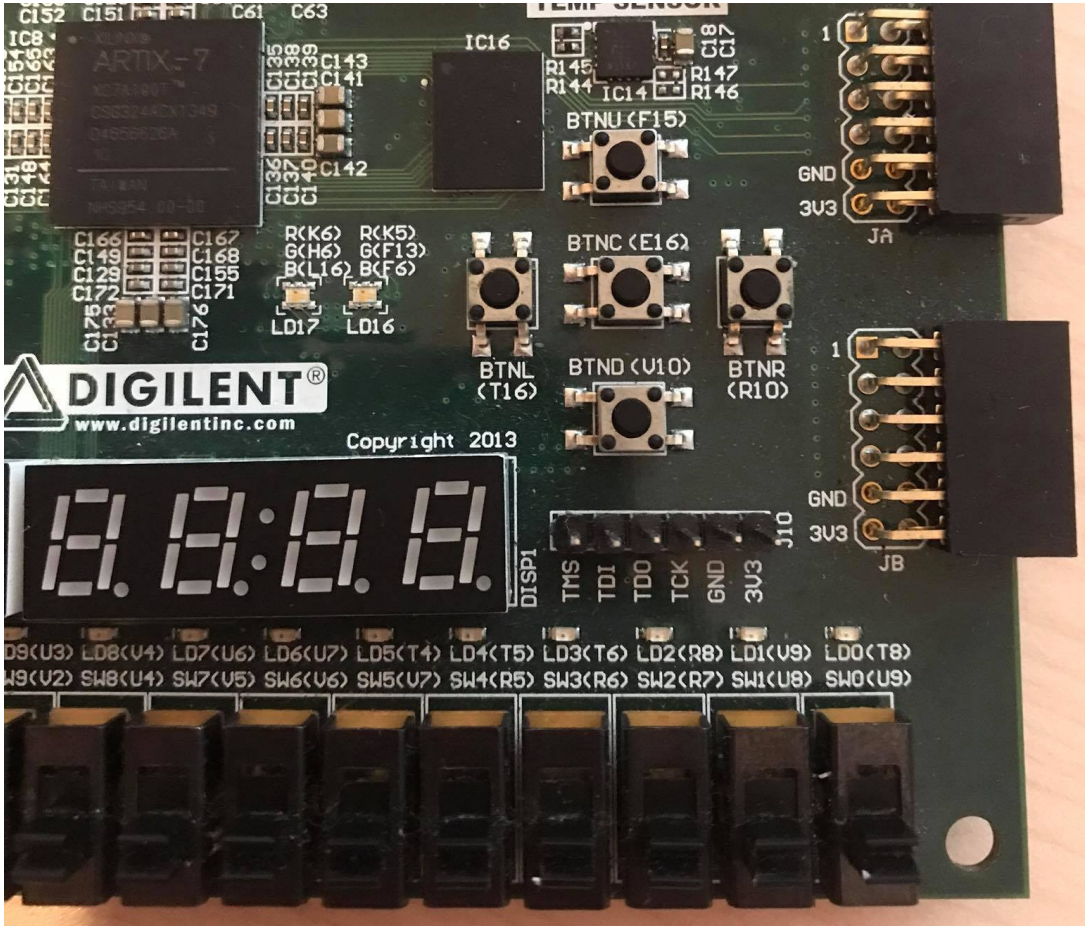
Signali gradnika in FPGA

- Pred sintezo je za glavni (zunanji) gradnik potrebno določiti kam naj bodo povezani zunanji signali
- npr.: a in b sta stikala, c & d led diode
- To počnemo v t.i. XDC datoteki (Xilinx Design Constraints)
- Sintaksa

```
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { a }];
# vektor
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { b[0] }];
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { b[1] }];
```

- Oznako fizičnega vhoda/izhoda najdete v reference manual-u razvojne plošče ali neposredno na razvojni plošči

Oznake pinov - razvojna plošča

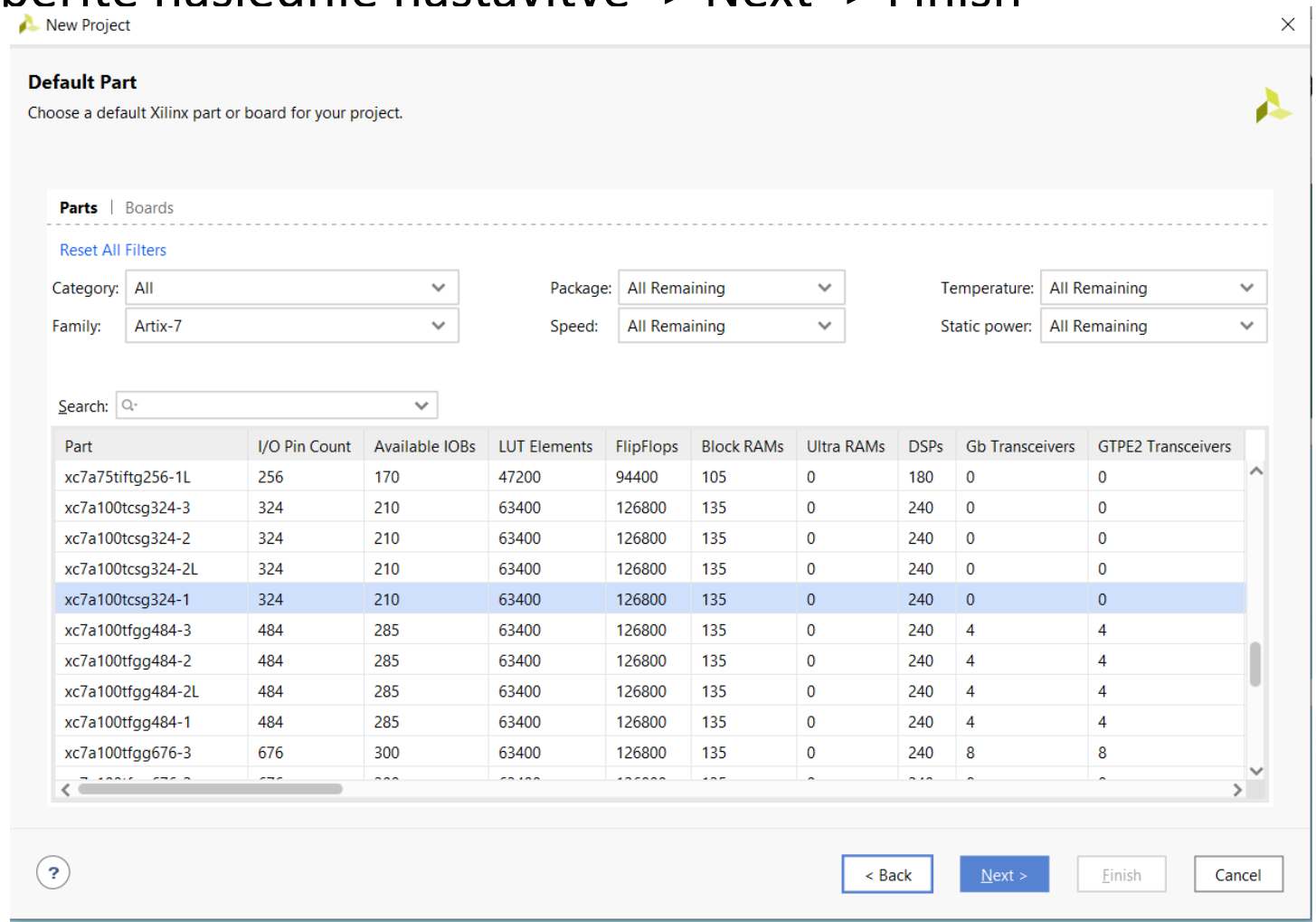


Projekt v Xillinx Vivado – Nexys4(DDR) in Nexys7

- Odprite Project Navigator
- File -> Project -> New ->
  - Izberite ime in direktorij
  - RTL project -> izberite source in XDC fajle

Projekt v Xillinx Vivado – Nexys4(DDR) in Nexys7

- Izberite naslednie nastavitve -> Next -> Finish



Projekt v Xillinx Vivado – Nexys4(DDR) in Nexys7

- Oznaka plošče:
  - xc7a50tcsg324-1 – Nexys A7 50T
  - xc7a100tcsg324-1 – Nexys A7 100T, Nexys A4 DDR, Nexys A4

Naloge

- Vzpostavite prvi projekt in napišite VHDL modul s katerim boste prižigali/ugašali LED diodo
- Realizirajte primerjalnik dveh štiri-bitnih števil (vsako število = štiri stikala)
  - izhod=2, ko je prvo število večje
  - izhod=1, ko je prvo število manjše
  - izhod=0, ko sta števili enaki