

vaja 01

Uvod: jezik VHDL in orodje Xilinx Vivado

Digitalno načrtovanje – laboratorijske vaje
asistent: Nejc Ilc

Potek vaj

- opisovanje digitalnega sistema z jezikom VHDL
- osnovno delo s perifernimi napravami (stikala, gumbi, LED, 7-segmentni prikazovalnik, ...)
- vključitev in povezovanje CPE PicoBlaze
- izdelava krmilnika za UART
- delo na seminarški nalogi

- Tedenski izzivi
 - na vajah razložimo navodila
 - oddaja rešitev na spletno učilnico

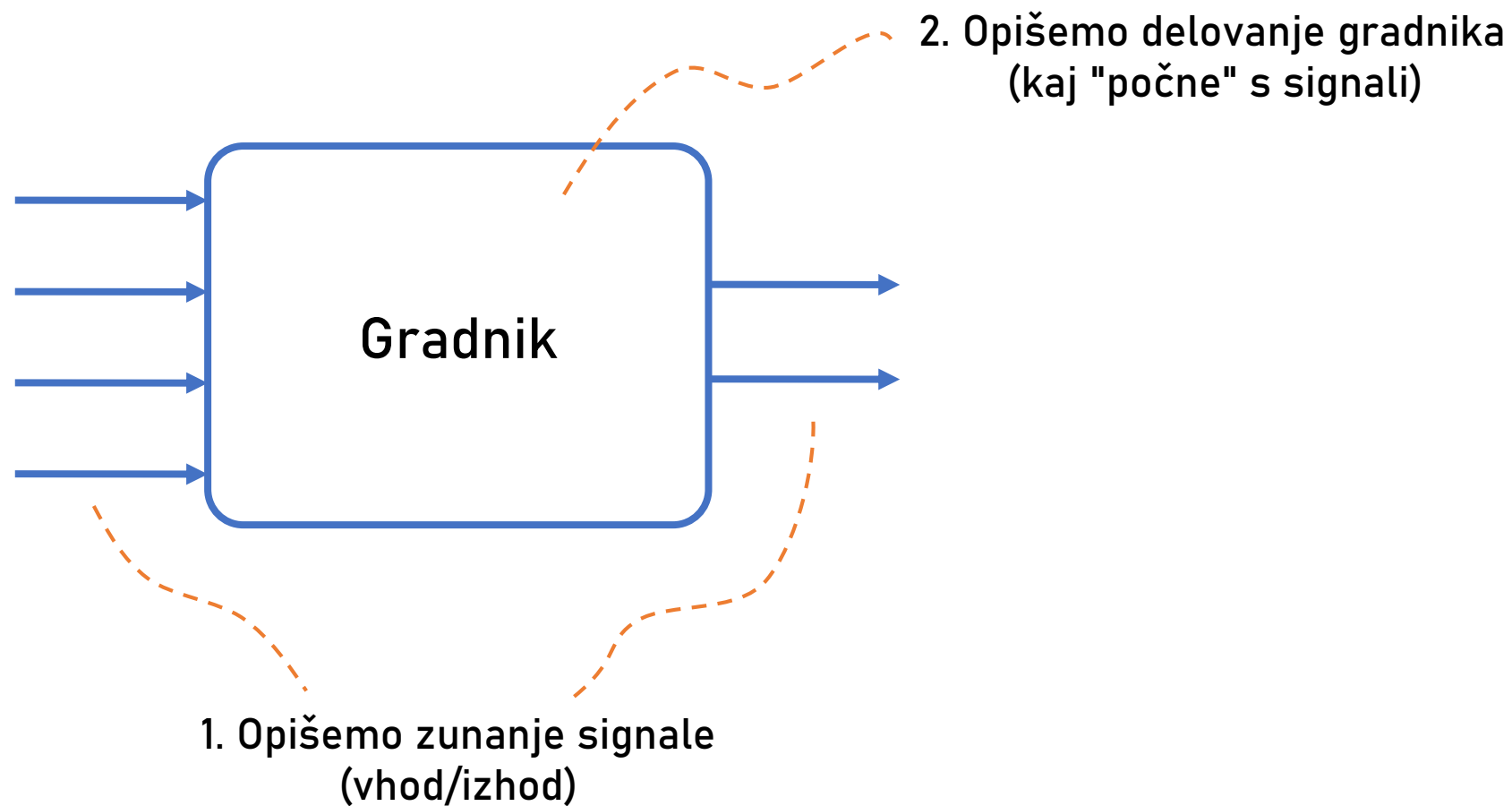
Jezik VHDL

- VHDL = VHSIC Hardware Description Language
- VHSIC = very-high-speed integrated circuits
- Visokonivojski jezik za opisovanje in simulacijo digitalnega vezja
- Opis v VHDL bomo sintetizirali in programirali "programirljivo polje vrat" (FPGA)

Orodja

- Razvojno okolje Xilinx Vivado ML Standard Edition (brezplačno)
- Razvojna plošča Digilent Nexys
 - Nexys4
 - Nexys4 DDR
 - Nexys A7 50T in 100T

Opis osnovnega gradnika



Opis zunanjih signalov



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

```
entity ime_gradnika is
  port(
    ime_signala_1 : smer tip_signala;
    ime_signala_2 : smer tip_signala;
    ...
    ime_signala_n : smer tip_signala
  );
end ime_gradnika;
```

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

Opis zunanjih signalov: primer



```
entity Gradnik is
  port(
    a: in  std_logic_vector(1 downto 0);
    b: in  std_logic_vector(0 to 3);
    c: out std_logic_vector(7 downto 0)
  );
end Gradnik;
```

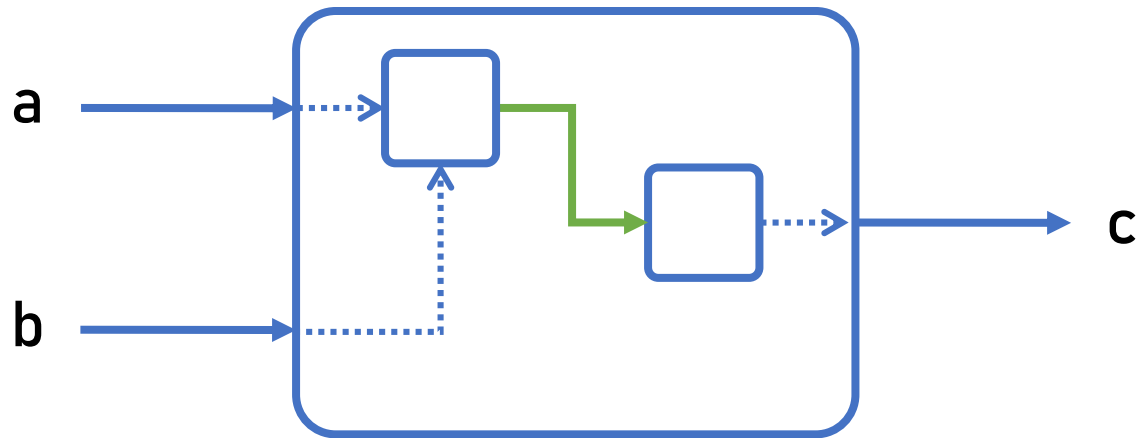
Opis delovanja gradnika

```
architecture ime_opisa of ime_gradnika is
    -- deklaracija notranjih signalov

begin
    -- stavki za opis vezja

end ime_opisa;
```


Deklaracija notranjih signalov



```
architecture ime_opisa of ime_gradnika is
```

```
    signal ime_signala: tip_signala;
```

```
begin
```

```
...
```

Prireditveni stavek

- Sintaksa

```
signal <= izraz;
```

- Primeri

Prireditev konstante:

```
a <= '0';
```

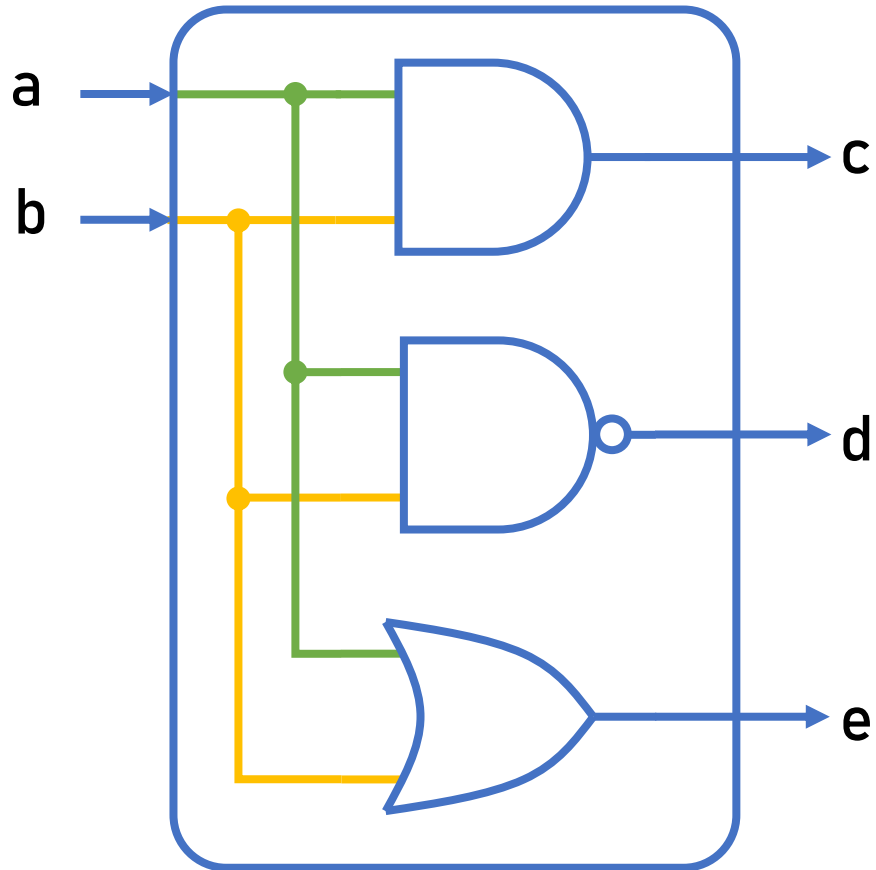
Prireditev konstante za vektor:

```
b <= "01001";
```

Prireditev delu vektorja:

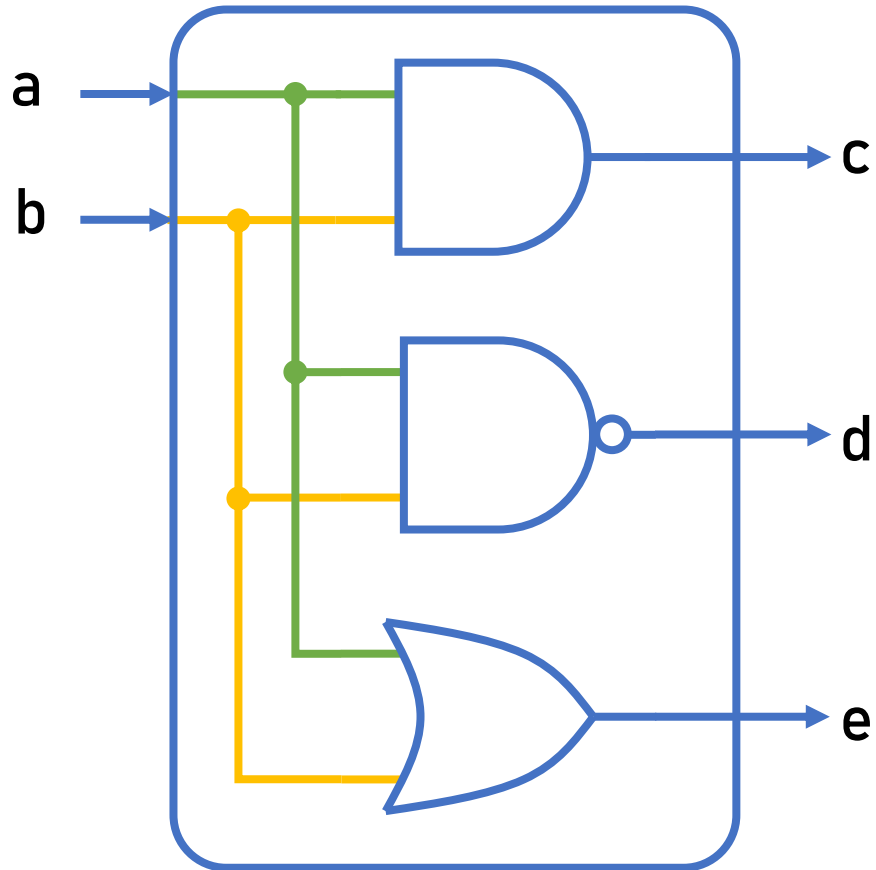
```
c(3 downto 0) <= "0111";
```

Prireditveni stavek: primer



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

Prireditveni stavek: primer



```
architecture arch of vrata is
begin
  -- osnovni logični operatorji:
  -- and, nand, or, nor, xor,
  -- xnor, not
  c <= a and b;
  d <= a nand b;
  e <= a or b;
end arch;
```

Pogojni prireditveni stavek

- Sintaksa

```
signal <= izraz_1 when pogoj_1 else izraz_2;
```

```
signal <= izraz_1 when pogoj_1 else  
        izraz_2 when pogoj_2 else  
        izraz_3;
```

- Pogojni operatorji:

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

Stavek "select"

- Sintaksa

```
with izbira select
```

```
    izhod <= izraz_1 when vred_1,
```

```
        izraz_2 when vred_2,
```

```
        ...
```

```
    izraz_df1;
```

- Nas to na kaj spominja?

Povezovanje zunanjih signalov na fizične priključke (pine)

- Pred sintezo je za glavni (top-level) gradnik potrebno določiti, kam naj bodo povezani zunanji signali
- Primer: a in b sta stikali, c, d in e pa diode LED
- To določimo v datoteki XDC (Xilinx Design Constraints)

- Sintaksa

```
set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports {a}];
```

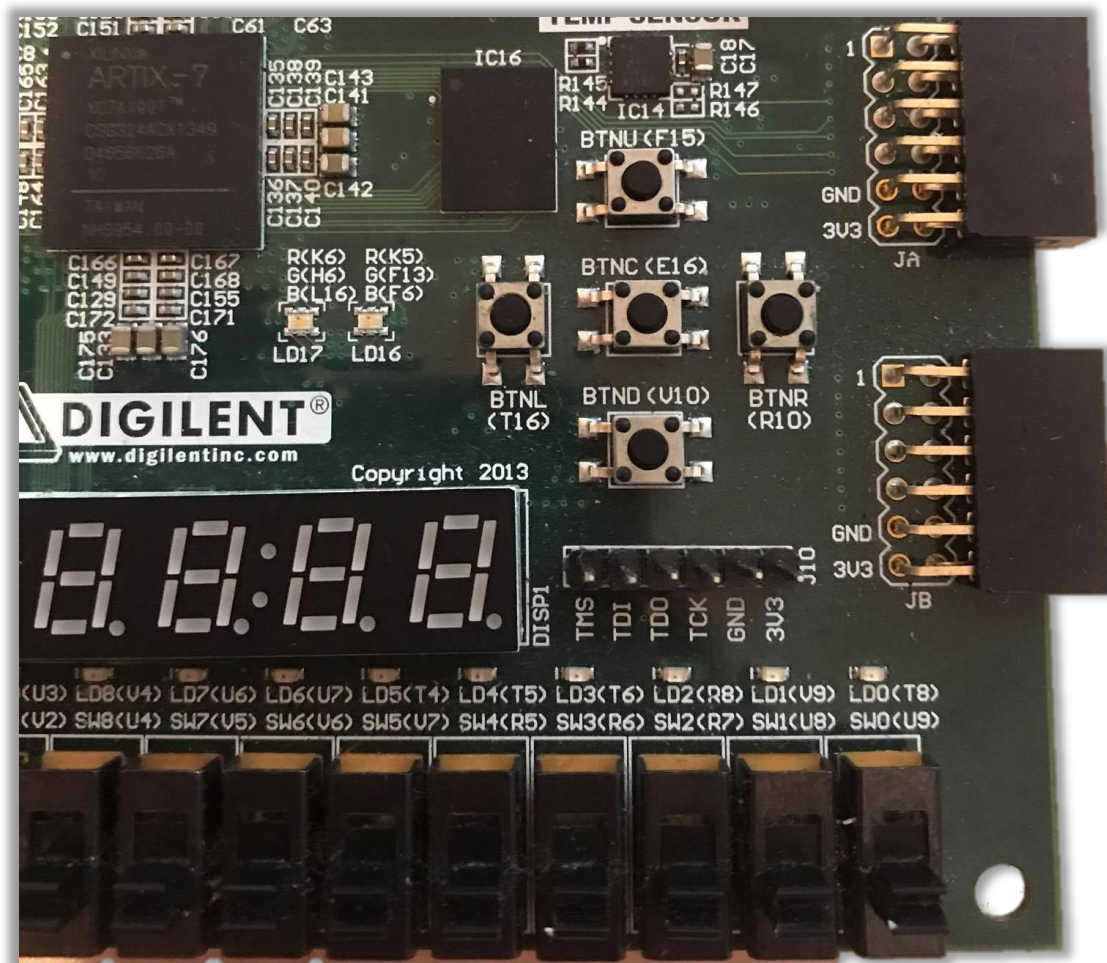
```
# vektor
```

```
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33} [get_ports {c[0]}];
```

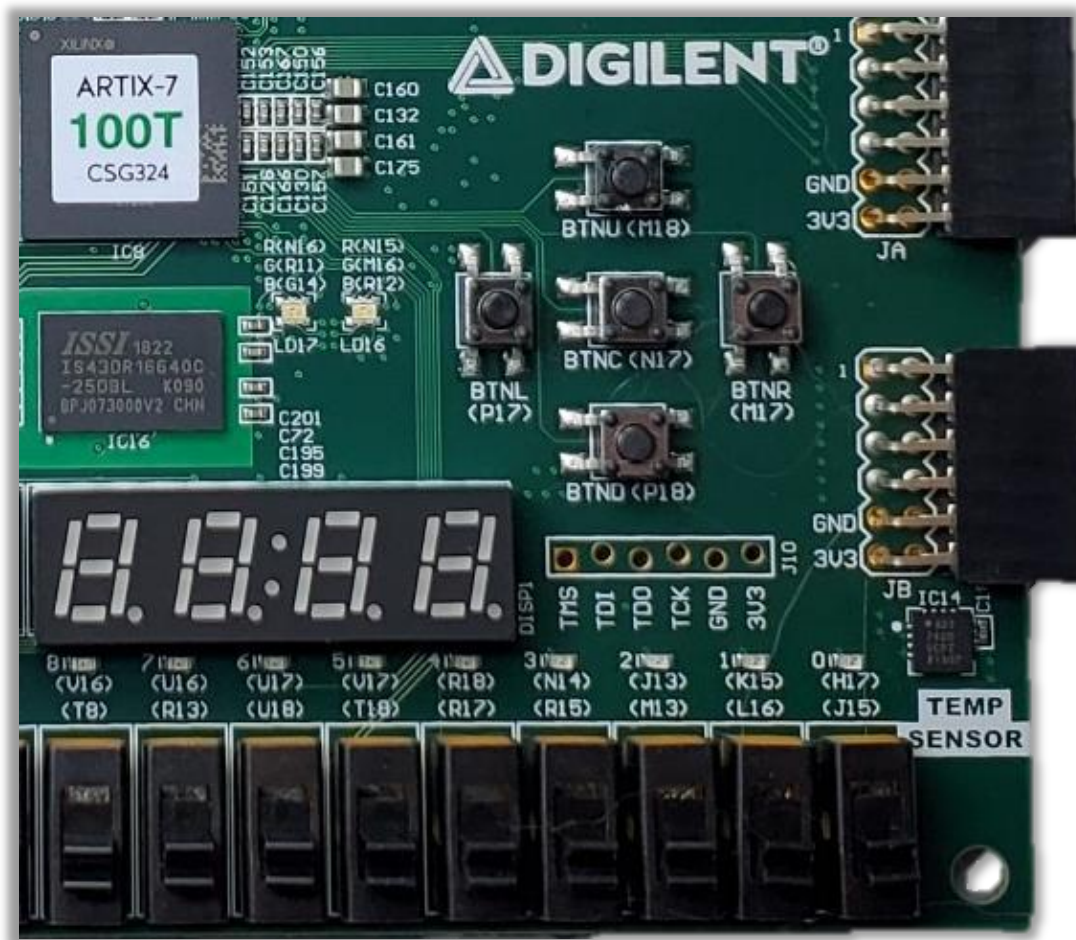
```
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33} [get_ports {c[1]}];
```

- Oznako pina najdete v dokumentaciji razvojne plošče ali neposredno na plošči.

Oznake pinov na razvojni plošči



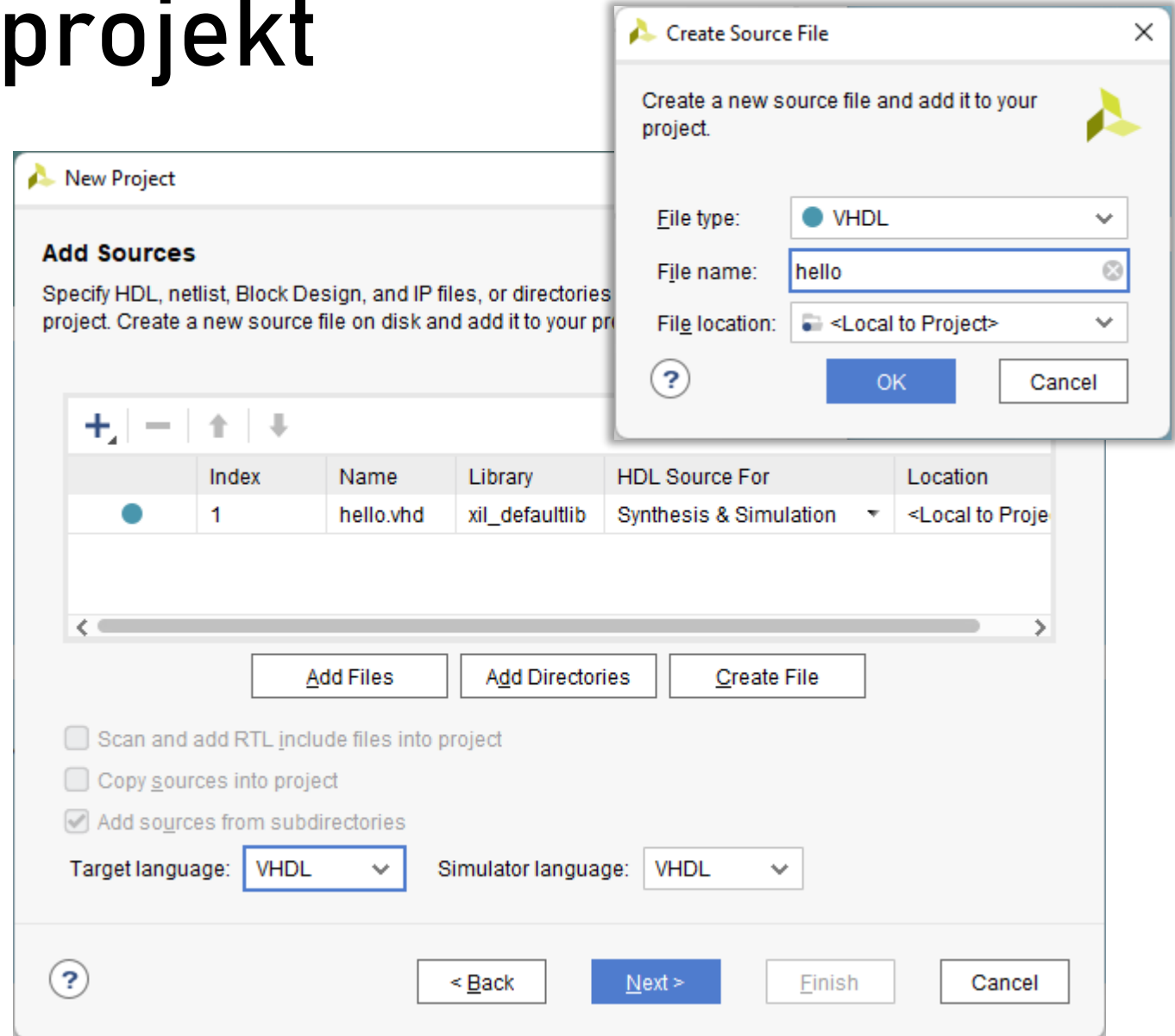
Nexys4



Nexys4 DDR / Nexys A7

Xilinx Vivado: nov projekt

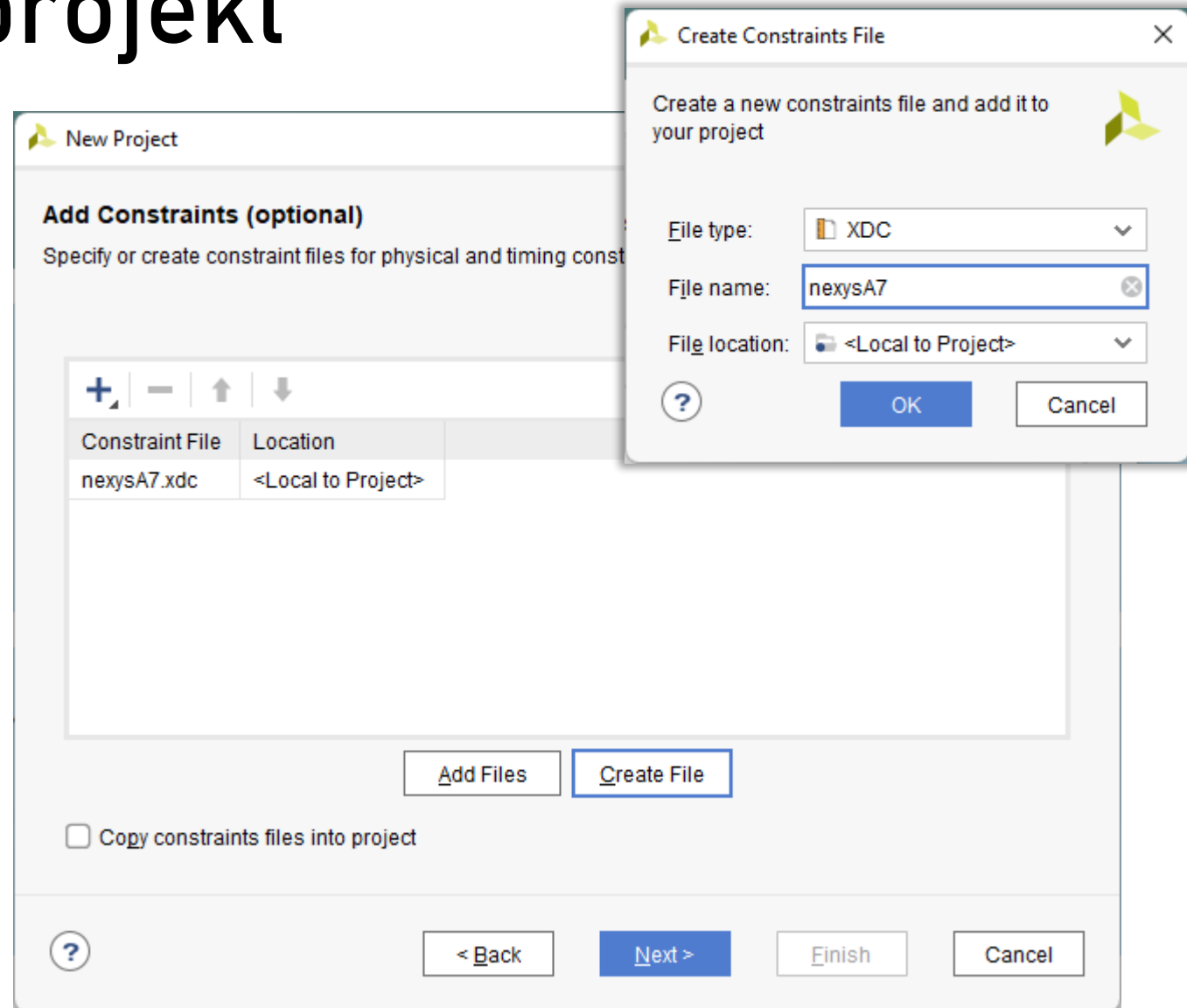
1. File → Project → New...
2. Določite ime projekta (Project name) in mesto na disku (Project location)
3. Izberite RTL Project
4. Določite/ustvarite datoteke z izvorno kodo (izberite jezik VHDL)



Xilinx Vivado: nov projekt

5. Določite/ustvarite datoteko XDC za povezovanje pinov

- na učilnici so povezave do že pripravljenih predlog



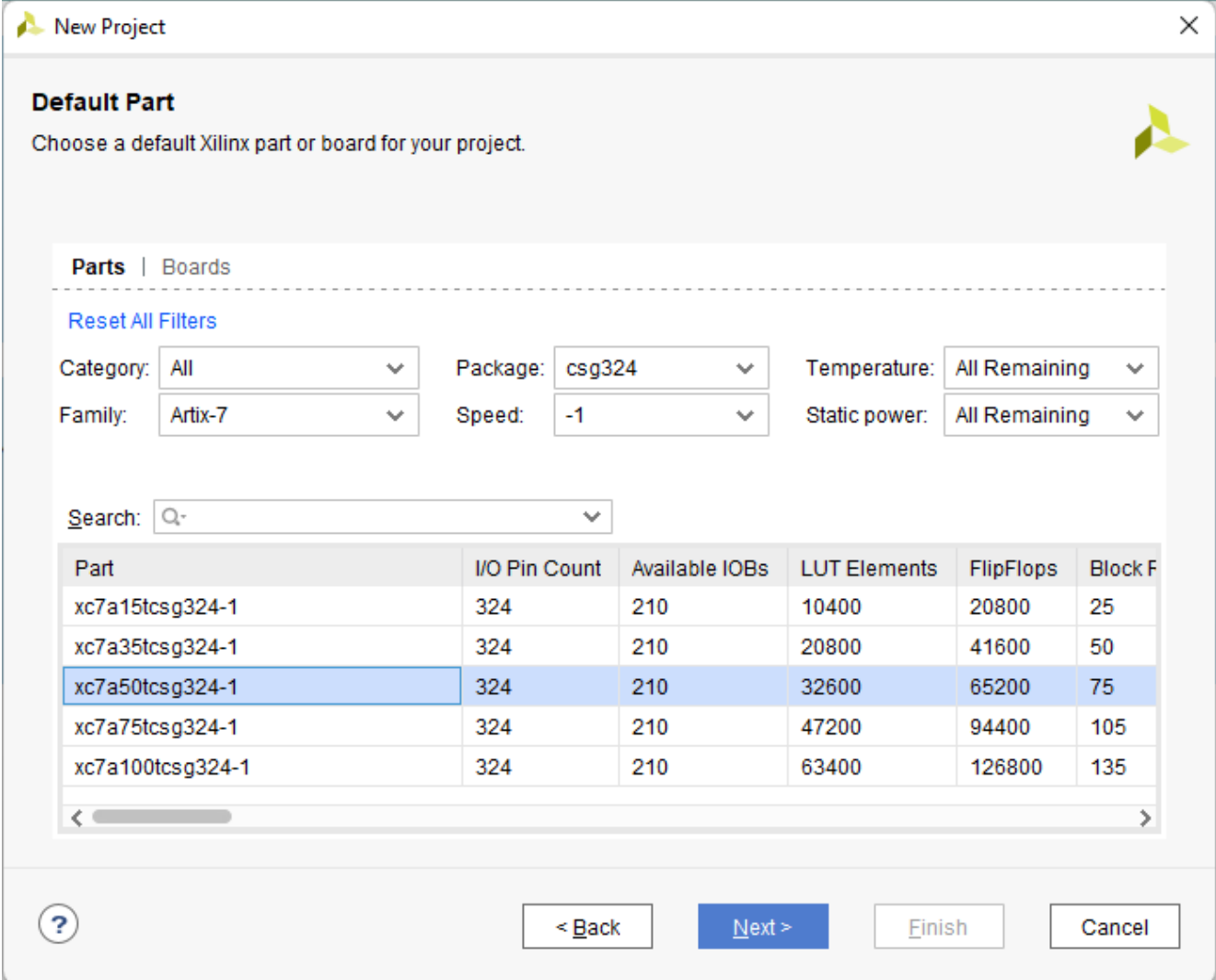
Xilinx Vivado: nov projekt

6. Izberite čip FPGA

- Nexys4,
Nexys4 DDR in
Nexys A7 100T:
xc7a100tcsg324-1
- Nexys A7 50T:
xc7a50tcsg324-1

7. Next

8. Finish



New Project

Default Part
Choose a default Xilinx part or board for your project.

Parts | Boards

[Reset All Filters](#)

Category: All Package: csg324 Temperature: All Remaining
Family: Artix-7 Speed: -1 Static power: All Remaining

Search: Q-

| Part | I/O Pin Count | Available IOBs | LUT Elements | FlipFlops | Block F |
|------------------------|---------------|----------------|--------------|--------------|-----------|
| xc7a15tcsg324-1 | 324 | 210 | 10400 | 20800 | 25 |
| xc7a35tcsg324-1 | 324 | 210 | 20800 | 41600 | 50 |
| xc7a50tcsg324-1 | 324 | 210 | 32600 | 65200 | 75 |
| xc7a75tcsg324-1 | 324 | 210 | 47200 | 94400 | 105 |
| xc7a100tcsg324-1 | 324 | 210 | 63400 | 126800 | 135 |

< ? > Back Next > Finish Cancel

Izziv

1. Ustvarite prvi projekt in opišite gradnik v VHDL, ki bo prižigal oziroma ugašal diodo LED glede na stanje stikala.
2. Opišite primerjalnik dveh štiri-bitnih števil
 - vsako število naj bo predstavljeno s štirimi stikali
 - rezultat primerjanja prikažite z diodami LED
 - izhod = 2 = 10_2 , ko je prvo število večje
 - izhod = 1 = 01_2 , ko je prvo število manjše
 - izhod = 0 = 00_2 , ko sta števili enaki