

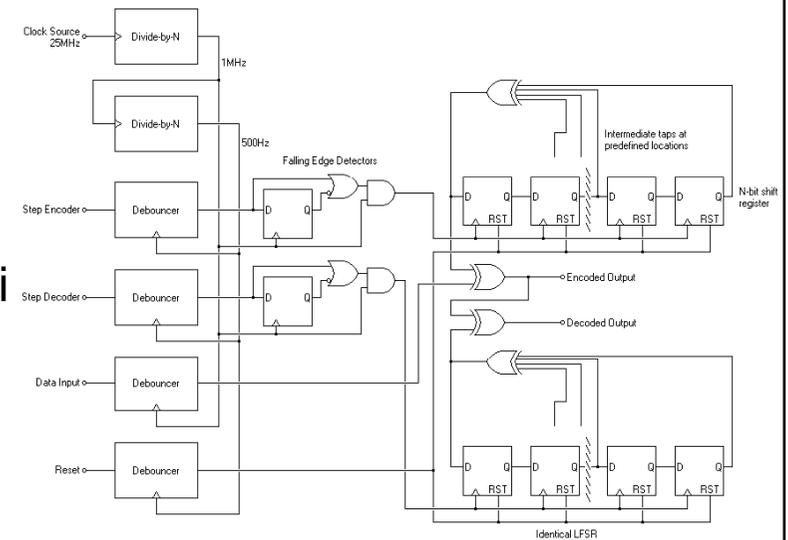
Digitalne strukture

4. Vaja

Opis vezij z VHDL

Načrtovanje digitalnih vezij:

- ponavadi shematsko (nivo logičnih vrat, tranzistorjev, registrov)
- pri kompleksnih vezjih zaradi večje preglednosti uporabimo **strojno opisne jezike**



Načrtovanje digitalnih vezij:

- ponavadi shematsko (nivo logičnih vrat, tranzistorjev, registrov)
- pri kompleksnih vezjih zaradi večje preglednosti uporabimo **strojno opisne jezike**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity prim is
    Port ( A : in STD_LOGIC_VECTOR (1 downto 0);
          B : in STD_LOGIC_VECTOR (1 downto 0);
          f : out STD_LOGIC);
end prim;

architecture Behavioral of prim is
    signal f1, f2, f3: std_logic;
begin
    f1<= A(1) or (not B(1));
    f2<= A(1) or A(0) or (not B(0));
    f3<= A(0) or (not B(1)) or (not B(0));
    f<= f1 and f2 and f3;
end Behavioral;
```

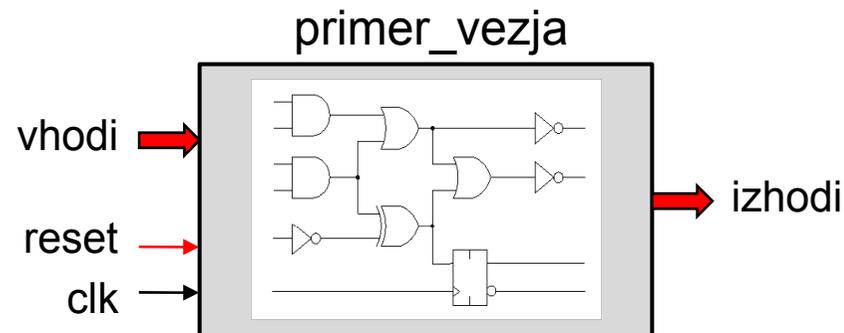
Načrtovanje digitalnih vezij:

- ponavadi shematsko (nivo logičnih vrat, tranzistorjev, registrov)
- pri kompleksnih vezjih zaradi večje preglednosti uporabimo **strojno opisne jezike**

Strojno opisni jeziki: jeziki za simulacijo in opis digitalnih vezij (VHDL, Verilog, Abel,...)

VHDL: **V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage
(jezik za opis zelo hitrih digitalnih vezij)

Model vezja v VHDL:



Model vezja v VHDL:

➤ vmesnik (*entity*)

entity primer_vezja **is**

port (vhod: **in** std_logic_vector (3 **downto** 0));

reset, clk: **in** std_logic;

izhod: **out** std_logic_vector (3 **downto** 0));

end primer_vezja;

➤ zgradba (*architecture*)

architecture opis **of** primer_vezja **is**

signal x, y, z: std_logic;

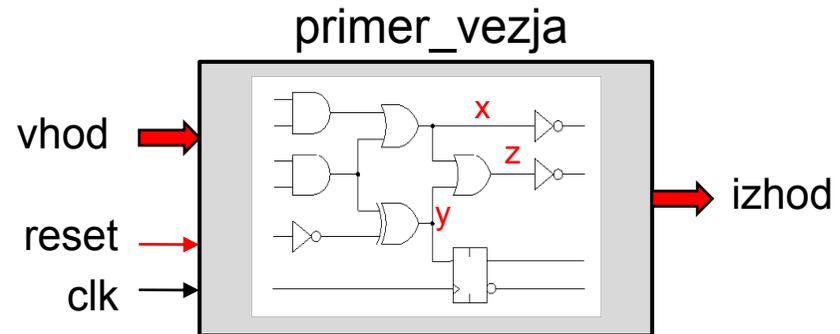
begin

x<= (vhod(0) **and** vhod(1)) **or** (vhod(2) **and** vhod(3));

izhod (0)<= not(x);

:

end opis;



4a) Enobitni primerjalnik

VHDL:

```
f<= x1 and not (x2);
```

4b) Dvobitni primerjalnik

Izhajajte iz oblike MDNO, vpeljite nove spremenljivke:

npr: $f(x_1, x_2, x_3) = x_1x_2 + x_2\bar{x}_3 = w+y$

VHDL:

```
signal w, y: std_logic;
```

```
f<= w or y;
```

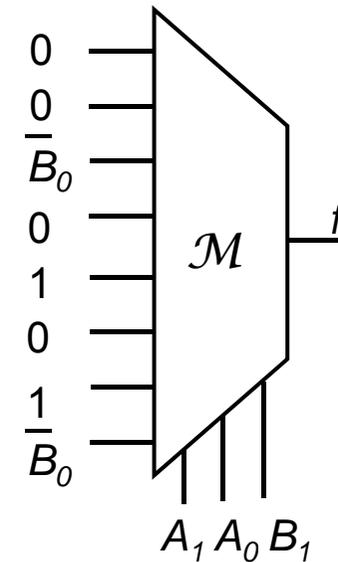
```
w<= x1 and x2;
```

```
y<= x2 and not (x3);
```

4c) Primerjalnik z multipleksorjem

Uporabite stavek *when...else*

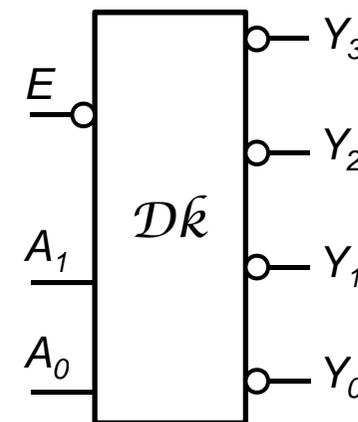
```
ime_spremenljivke <= vrednost1 when izraz1 else
                    vrednost2 when izraz2 else
                    ...
                    vrednost n;
```



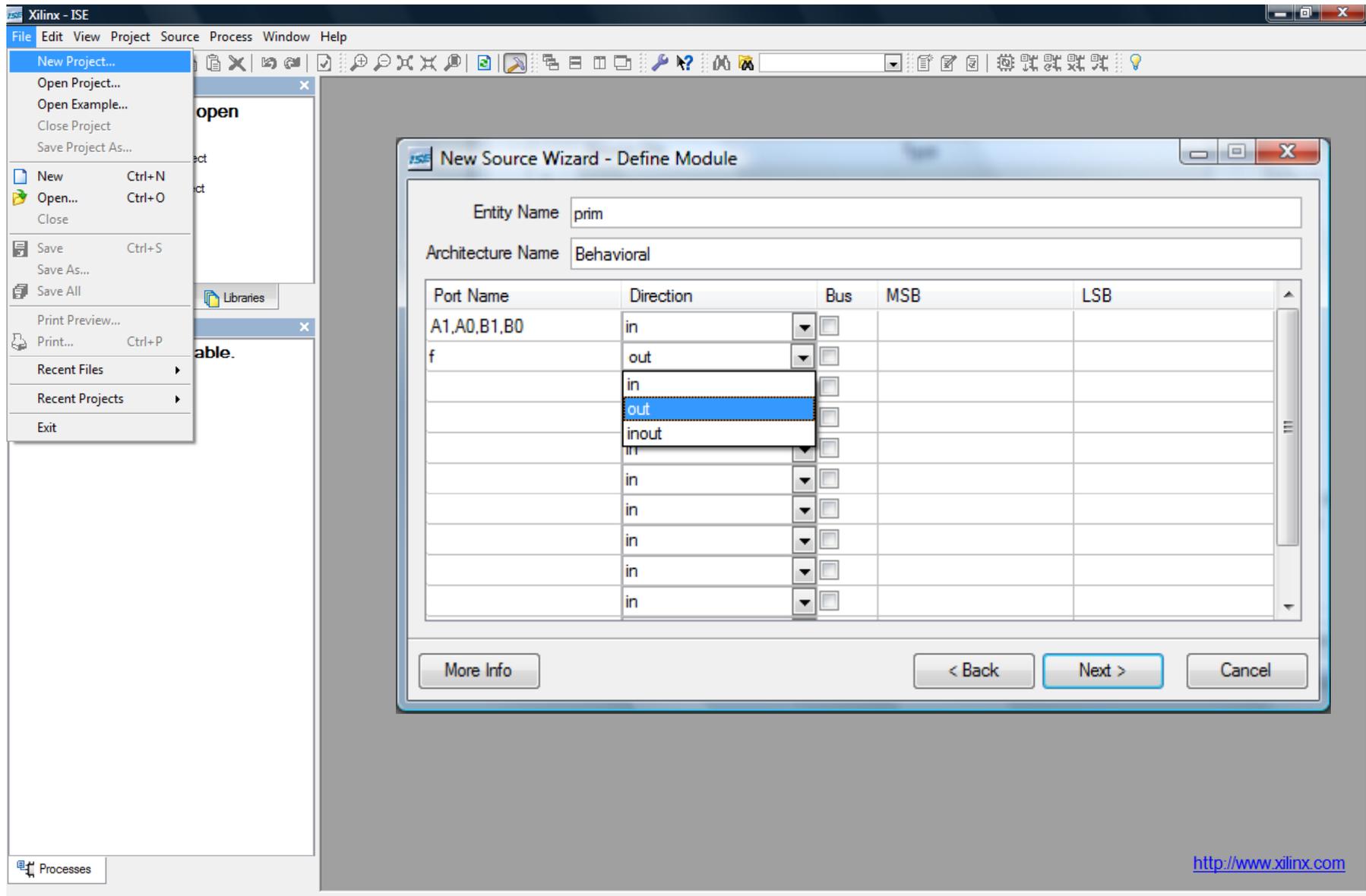
4d) Dekodirnik 2-4

Uporabite stavek *with...select...when*

```
with ime_izbirnega_signala select
ime_spremenljivke <= vrednost 1 when izbor 1,
                    vrednost 2 when izbor 2,
                    ...
                    vrednost n when others;
```



Xilinx ISE – Programiranje FPGA



Xilinx ISE – Programiranje FPGA

The screenshot displays the Xilinx ISE IDE interface. The main window shows a VHDL code editor for a file named 'prim.vhd'. The code is as follows:

```
1 -----  
2 -- Company:  
3 -- Engineer:  
4 --  
5 -- Create Date:    17:59:28 12/14/2010  
6 -- Design Name:  
7 -- Module Name:   prim - Behavioral  
8 -- Project Name:  
9 -- Target Devices:  
10 -- Tool versions:  
11 -- Description:  
12 --  
13 -- Dependencies:  
14 --  
15 -- Revision:  
16 -- Revision 0.01 - File Created  
17 -- Additional Comments:  
18 --  
19 -----  
20 library IEEE;  
21 use IEEE.STD_LOGIC_1164.ALL;  
22 use IEEE.STD_LOGIC_ARITH.ALL;  
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;  
24  
25 ---- Uncomment the following library declaration if instantiating  
26 ---- any Xilinx primitives in this code.  
27 --library UNISIM;  
28 --use UNISIM.VComponents.all;  
29  
30 entity prim is  
31     Port ( A1,A0,B1,B0 : in  STD_LOGIC;  
32           f : out  STD_LOGIC);  
33 end prim;  
34  
35 architecture Behavioral of prim is  
36  
37 begin  
38  
39  
40 end Behavioral;  
41  
42
```

Two red arrows point from the word 'komentarji' to the header section of the code (lines 2-17). The 'Sources' pane on the left shows the project structure with 'prim - Behavioral (prim.vhd)' selected. The 'Processes' pane shows various design steps like 'Synthesize - XST' and 'Implement Design'.

Xilinx ISE – Programiranje FPGA

The screenshot shows the Xilinx ISE IDE interface. The main window displays a VHDL file named 'prim.vhd' with the following code:

```
1 -----  
2 -- Company:  
3 -- Engineer:  
4 --  
5 -- Create Date:    17:59:28 12/14/2010  
6 -- Design Name:  
7 -- Module Name:   prim - Behavioral  
8 -- Project Name:  
9 -- Target Devices:  
10 -- Tool versions:  
11 -- Description:  
12 --  
13 -- Dependencies:  
14 --  
15 -- Revision:  
16 -- Revision 0.01 - File Created  
17 -- Additional Comments:  
18 --  
19 -----  
20 library IEEE;  
21 use IEEE.STD_LOGIC_1164.ALL;  
22 use IEEE.STD_LOGIC_ARITH.ALL;  
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;  
24  
25 ---- Uncomment the following library declaration if instantiating  
26 ---- any Xilinx primitives in this code.  
27 --library UNISIM;  
28 --use UNISIM.vcomponents.all;  
29  
30 entity prim is  
31     Port ( A1,A0,B1,B0 : in  STD_LOGIC;  
32           f : out  STD_LOGIC);  
33 end prim;  
34  
35 architecture Behavioral of prim is  
36  
37 begin  
38  
39  
40 end Behavioral;  
41  
42
```

The 'Processes' window on the left shows the following list of actions:

- Add Existing Source
- Create New Source
- View Design Summary
- Design Utilities
- User Constraints
- Synthesize - XST
 - View Synthesis Report
 - View RTL Schematic
 - View Technology Schematic
 - Check Syntax**
 - Generate Post-Synthesis Simulation Model
- Implement Design
- Generate Programming File

A red box highlights the 'Check Syntax' option in the 'Synthesize - XST' sub-menu. A red arrow points from this box to a larger red box containing the text 'Check Syntax'.

Xilinx ISE – Programiranje FPGA

The screenshot shows the Xilinx ISE interface with the following components:

- Sources:** Shows the project structure for 'prim.vhd'.
- Design Browser:** Displays the hierarchy: I/O Pins, Global Logic, and Logic.
- Design Object List - I/O Pins:** A table listing the I/O pins and their configurations.
- Processes:** A list of design steps, with 'Assign Package Pins' highlighted in a red box.
- Device Architecture:** Shows the physical layout of the xc3s200-4-ft256 device.

I/O Name	I/O Direction	Loc	Bank	I/O Std.
A0	Input	k14	BANK	
A1	Input	k13	BANK	
B0	Input	j14	BANK	
B1	Input	j13	BANK	
f	Output	p13		

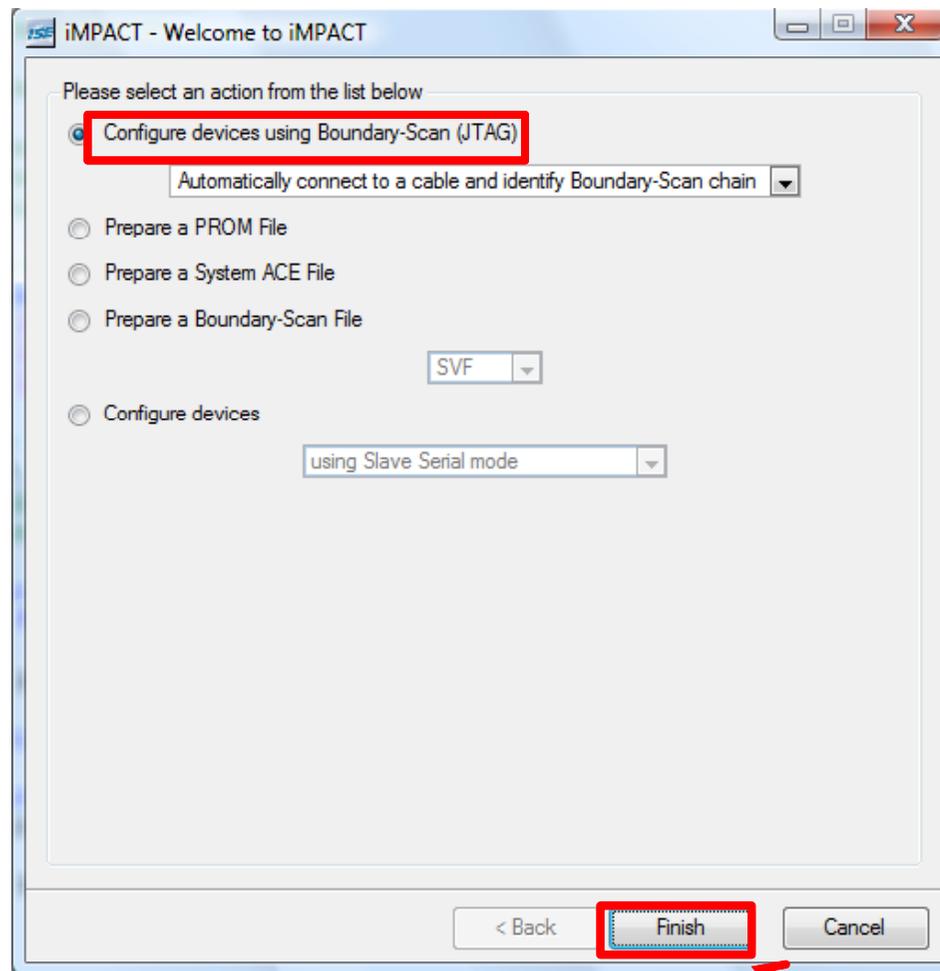
Assign Package Pins

Xilinx ISE – Programiranje FPGA

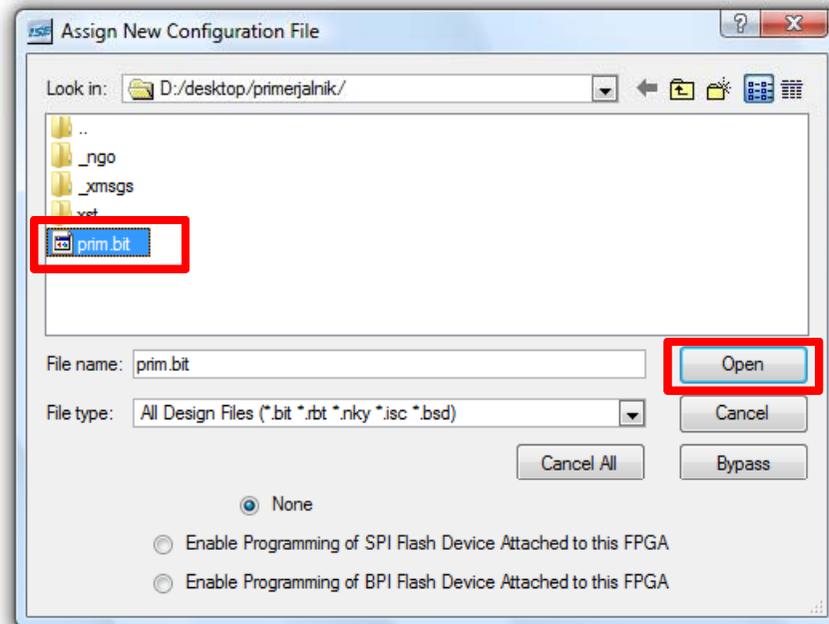
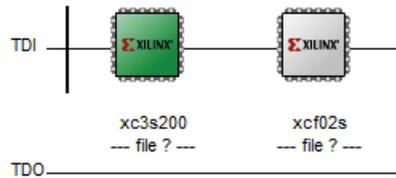
The screenshot shows the Xilinx ISE IDE interface. The 'Sources' window on the left shows the project structure for 'prim.vhd'. The 'Processes' window is open, displaying a list of tasks for the 'prim - Behavioral' project. The task 'Generate PROM, ACE, or JTAG File' is highlighted with a red box. A red arrow points from this box to another red box on the right containing the text 'Generate PROM, ACE or JTAG File'. The main editor window displays the VHDL code for the 'prim' module, including library declarations for IEEE and UNISIM, and the module's architecture.

```
1 -----
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date:    17:59:28 12/14/2010
6 -- Design Name:
7 -- Module Name:   prim - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity prim is
31     Port ( A1,A0,B1,B0 : in  STD_LOGIC;
32           f : out  STD_LOGIC);
33 end prim;
34
35 architecture Behavioral of prim is
36
37 begin
38     f<=(A1 and A0)or(B1 and B0);
39
40 end Behavioral;
41
42
```

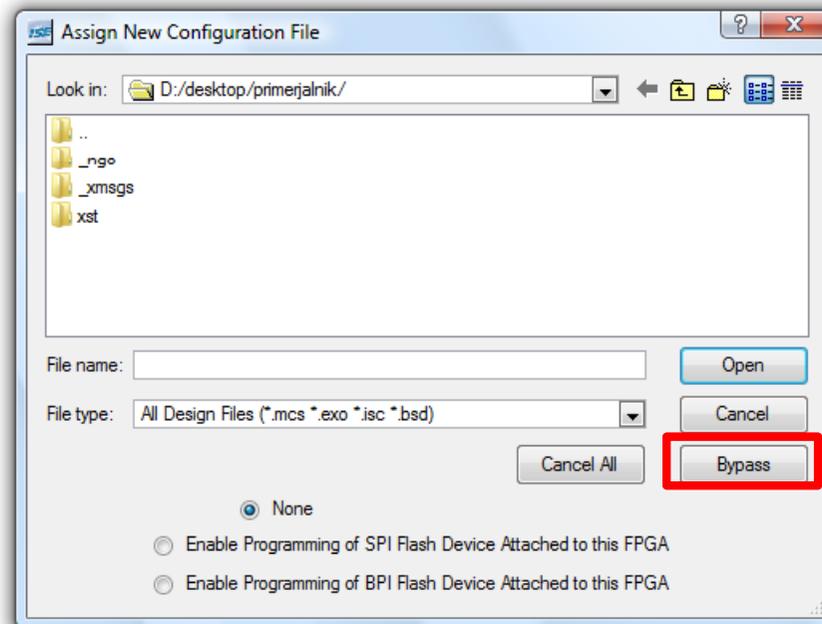
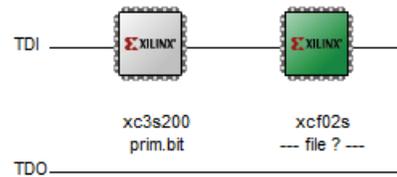
Xilinx ISE – Programiranje FPGA



Xilinx ISE – Programiranje FPGA

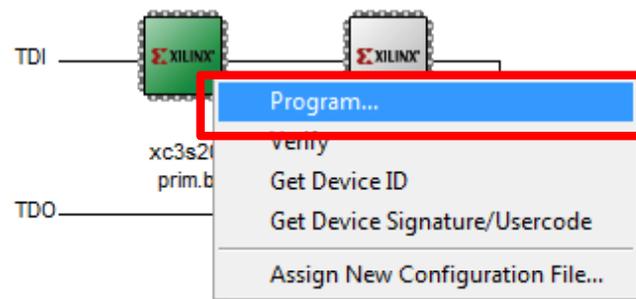


Xilinx ISE – Programiranje FPGA



Xilinx ISE – Programiranje FPGA

Desni klik na vezje xc3s200



Poročilo

- ❑ Besedilo vaje
- ❑ VHDL koda za vsako nalogo
- ❑ RTL shematika

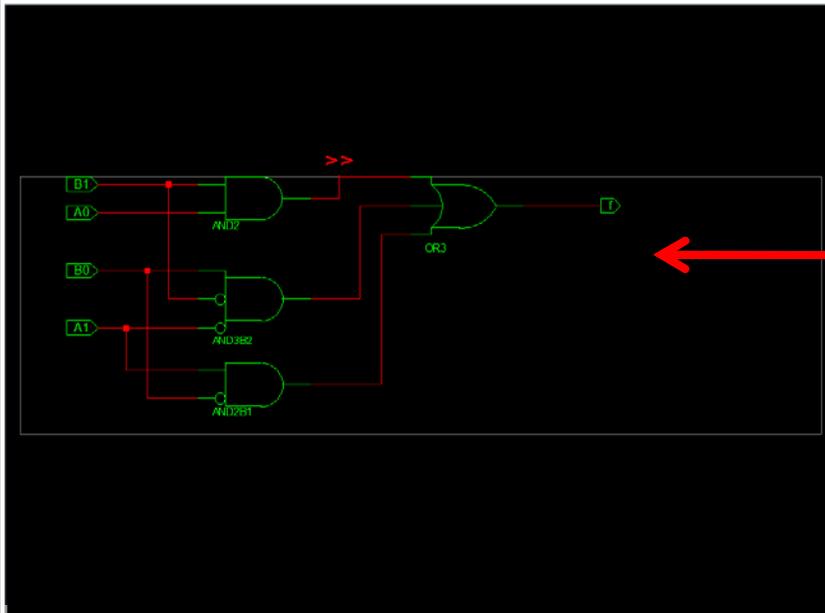
View RTL Schematic

The screenshot shows the Xilinx ISE interface. The 'Sources' window on the left lists the project files, including 'prim - Behavioral (prim.vhd)'. The 'Processes' window on the left shows a tree of actions, with 'View RTL Schematic' highlighted. The main editor window on the right displays VHDL code for a module named 'prim'.

```
1
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    17:59:28 12/14/2010
6  -- Design Name:
7  -- Module Name:    prim - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library d
26 ---- any Xilinx primitives in this cod
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity prim is
31     Port ( A1,A0,B1,B0 : in  STD_LOGIC
32           f : out  STD_LOGIC);
33 end prim;
34
35 architecture Behavioral of prim is
36
37 begin
38 f<=(A1 and A0)or(B1 and B0);
39
40 end Behavioral;
41
42
```

Poročilo

- ❑ Besedilo vaje
- ❑ VHDL koda za vsako nalogo
- ❑ RTL shematika



```
1  -----
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    17:59:28 12/14/2010
6  -- Design Name:
7  -- Module Name:    prim - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library d
26 ---- any Xilinx primitives in this cod
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity prim is
31     Port ( A1,A0,B1,B0 : in  STD_LOGIC
32           f : out  STD_LOGIC);
33 end prim;
34
35 architecture Behavioral of prim is
36
37 begin
38 f<=(A1 and A0)or(B1 and B0);
39
40 end Behavioral;
41
42
```