

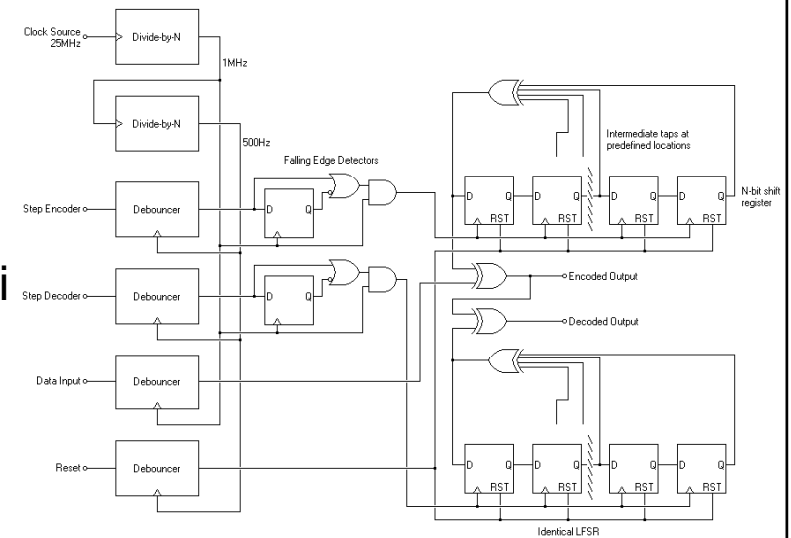
# Digitalna tehnika

## 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 (notB(0));
    f3<= A(0) or (not B(1)) or (not B(0));
    f<= f1 and f2 and f3;
end Behavioral;
```

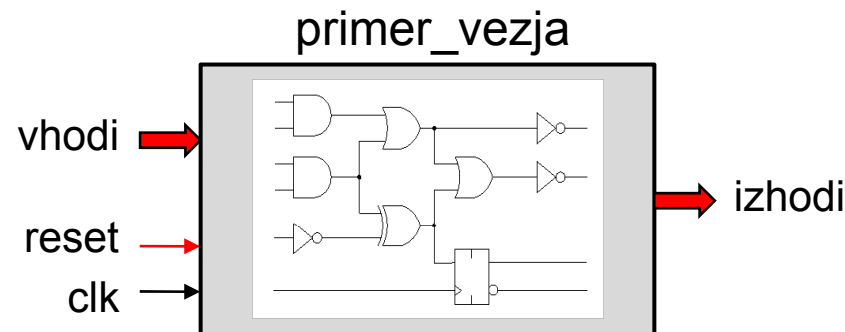
Načrtovanje digitalnih vezij:

- 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;

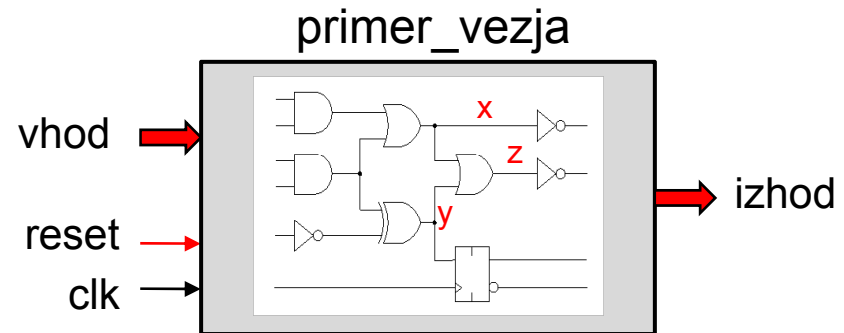
**begin**

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;
```

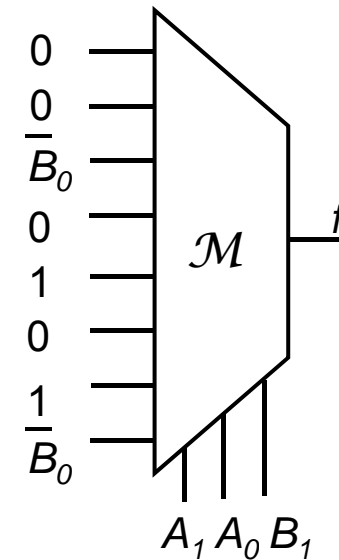
```
w<= x1 and x2;
```

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

## 4c) Primerjalnik z multipleksorjem

Uporabite stavek *when...else*

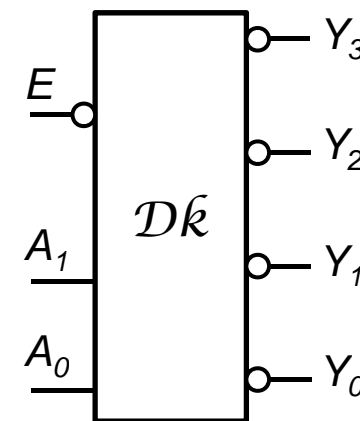
```
ime_spremenljivke <= vrednost1 when izraz1 else
    vrednost2 when izraz2 else
    ...
    vrednost n;
```



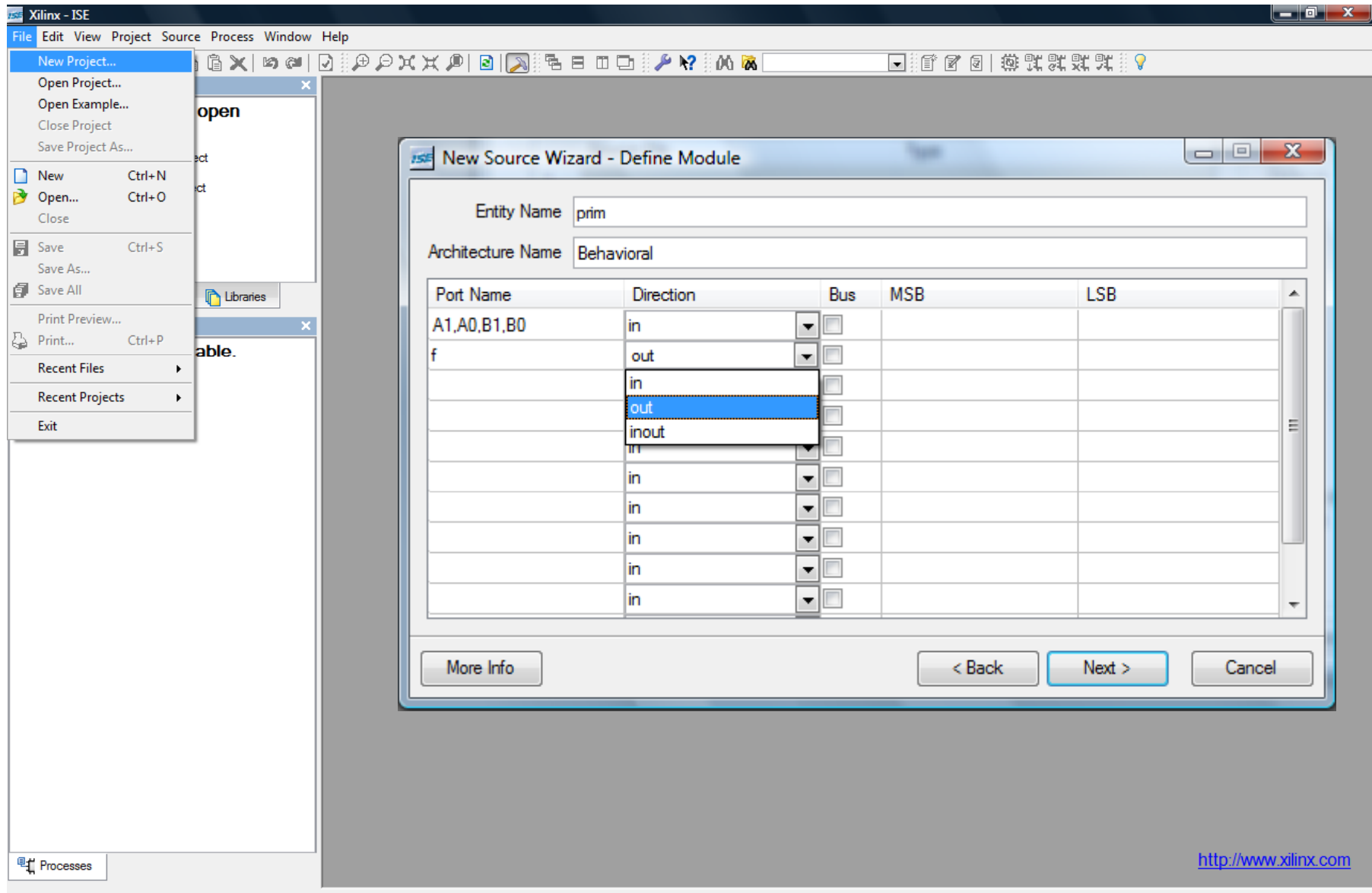
## 4d) Dekodirnik 2-4

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





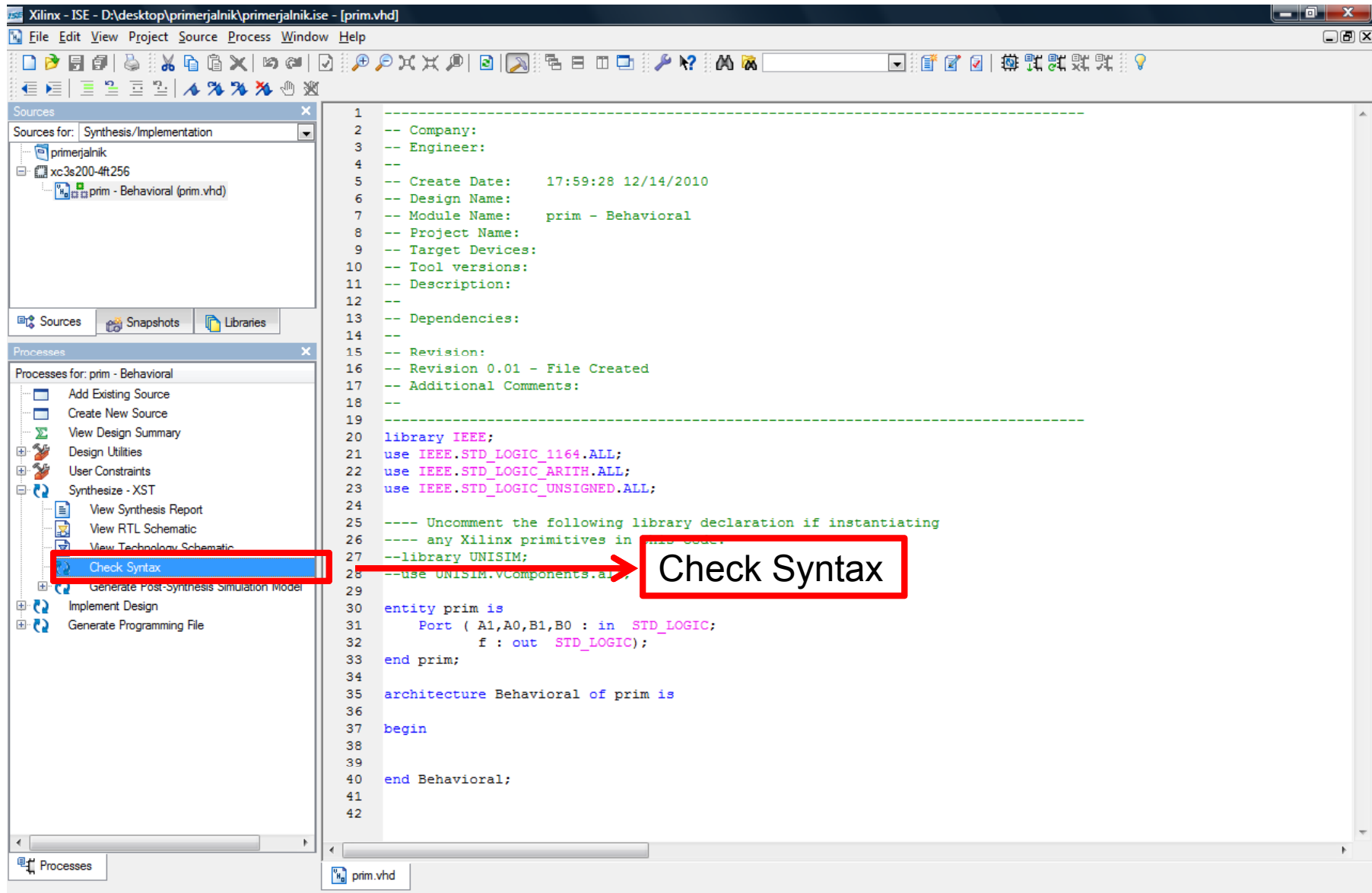
## Xilinx ISE – Programiranje FPGA

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 comment section of the code (lines 2-18). The left sidebar shows the 'Sources' and 'Processes' panels. The 'Sources' panel shows the project structure with 'prim - Behavioral (prim.vhd)' selected. The 'Processes' panel shows the design flow options.

# Xilinx ISE – Programiranje FPGA



# Xilinx ISE – Programiranje FPGA

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

- Sources:** prim.jalnik, xc3s200-4ft256, prim - Behavioral (prim.vhd)
- Design Browser:** I/O Pins, Global Logic, Logic
- Design Object List - I/O Pins:**

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		

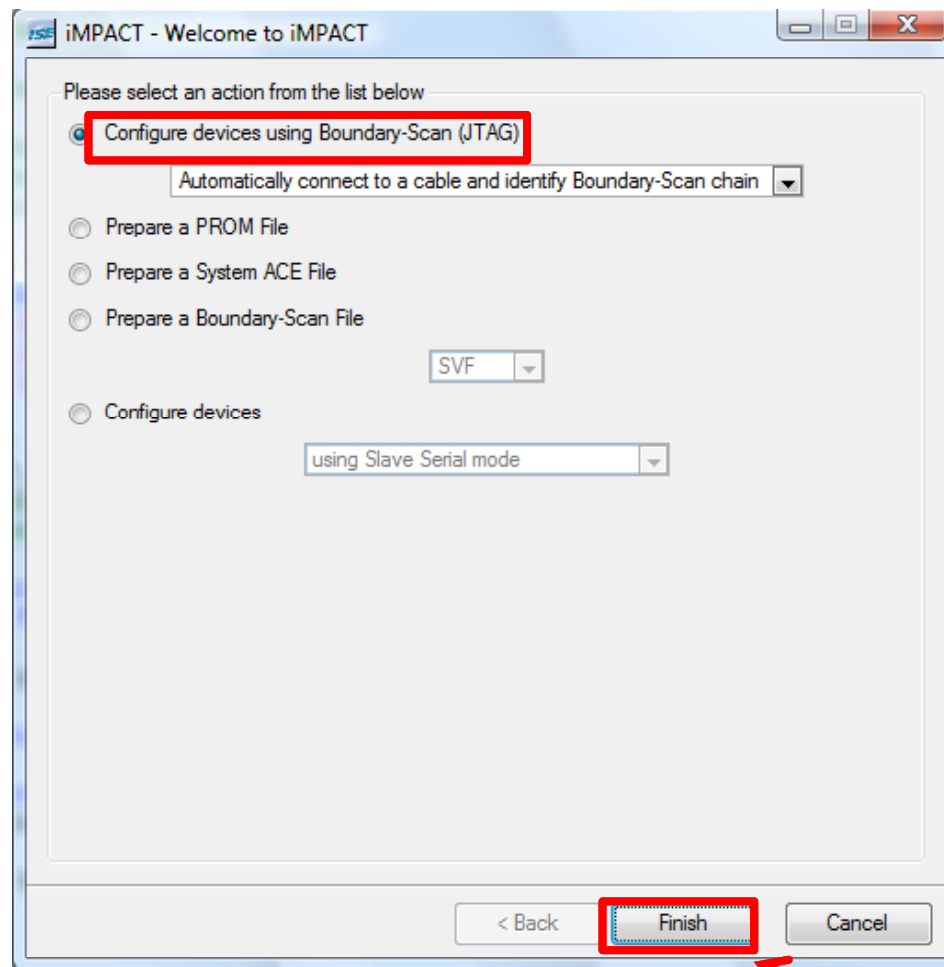
The 'Processes' list on the left includes 'Assign Package Pins', which is highlighted with a red box. A red arrow points from this box to a text box containing 'Assign Package Pins'.

# Xilinx ISE – Programiranje FPGA

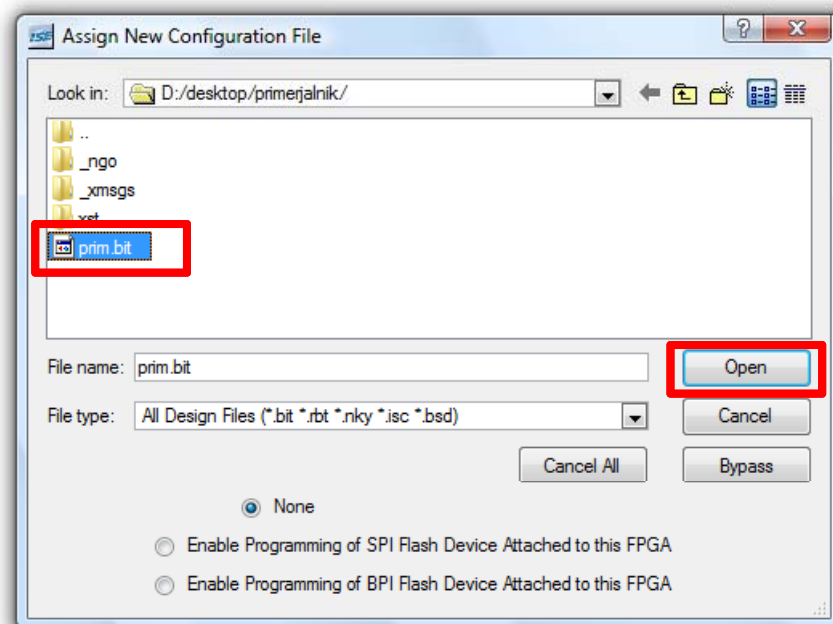
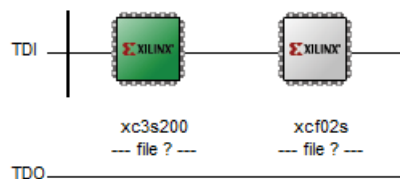
The screenshot shows the Xilinx ISE interface. The 'Sources' window on the left shows the project structure for 'prim.vhd'. The 'Processes' window on the left lists various tasks, with 'Generate PROM, ACE, or JTAG File' highlighted in a red box. An arrow points from this box to a larger red box on the right containing the text 'Generate PROM, ACE or JTAG File'. The main editor window displays the VHDL code for the 'prim' entity, including library declarations for IEEE and UNISIM, and the entity 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
```

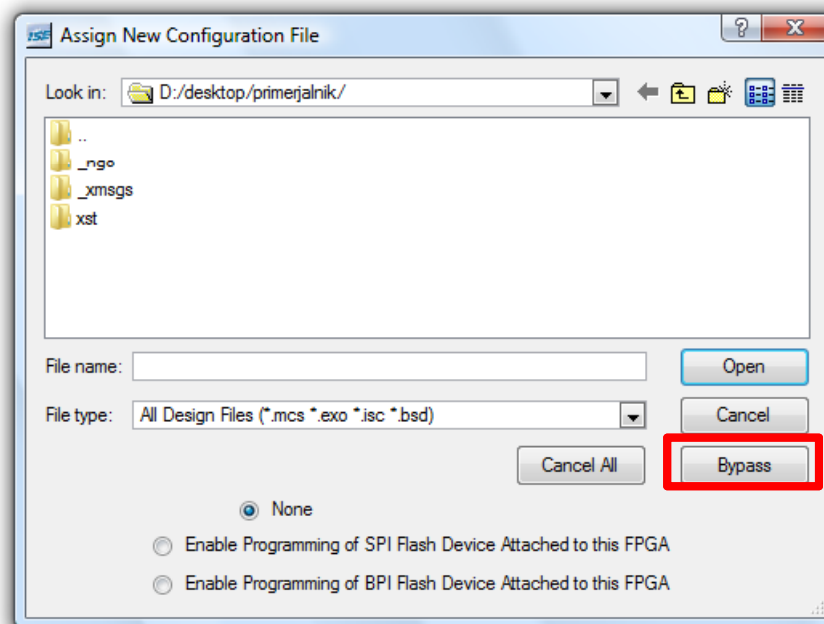
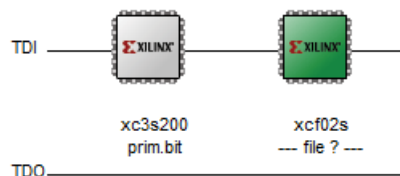
## Xilinx ISE – Programiranje FPGA



## Xilinx ISE – Programiranje FPGA

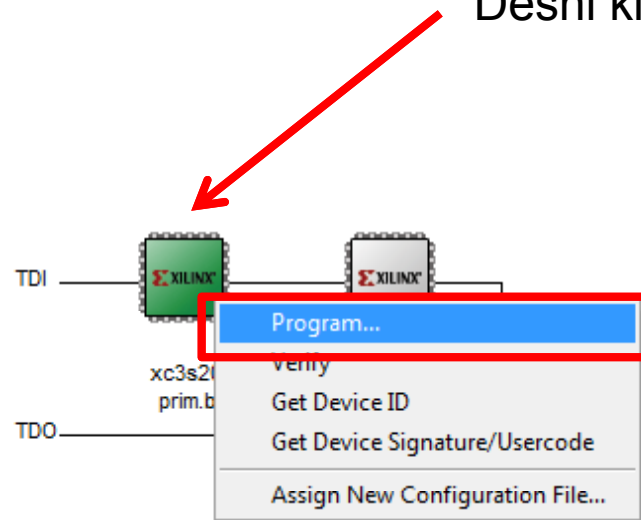


## Xilinx ISE – Programiranje FPGA



# Xilinx ISE – Programiranje FPGA

Desni klik na vezje xc3s200





## Poročilo

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

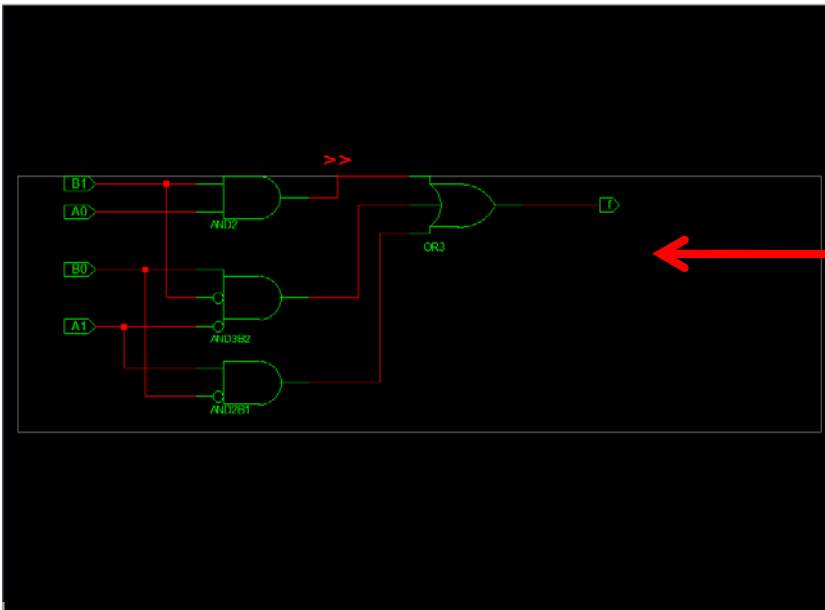
View RTL Schematic

The screenshot shows the Xilinx ISE software interface. The 'Processes' window is open, displaying a tree view of available actions for the 'prim - Behavioral' project. The 'View RTL Schematic' option is highlighted with a red box. The main editor window shows the VHDL code for the 'prim.vhd' file, which includes a header with project information and a behavioral implementation of a logic function. 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 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
```