



Preklopna vezja

4. Vaja

Realizacija sinhronskega sekvenčnega vezja s
programirljivim vezjem FPGA

as. Matej Kranjc

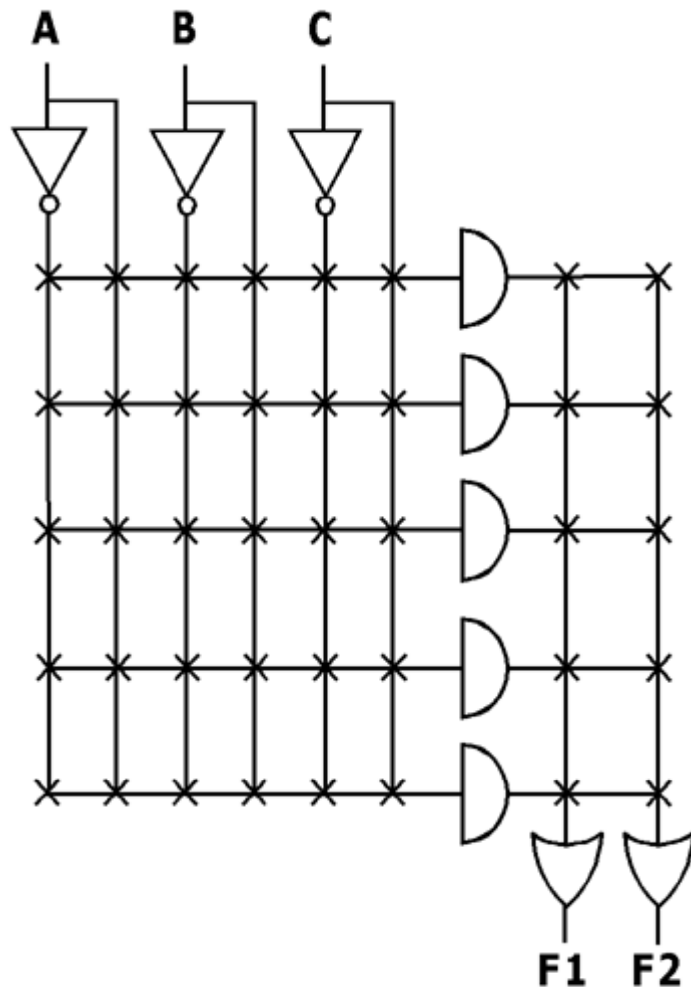


Naloga vaje

1. del: S programirljivim vezjem FPGA zgradite sinhronski števec iz vaje 3.1
2. del: Zgradite sinhronsko sekvenčno vezje, ki bo vklapljalno smernike pri avtomobilu



PLD – Programmable Logic Device

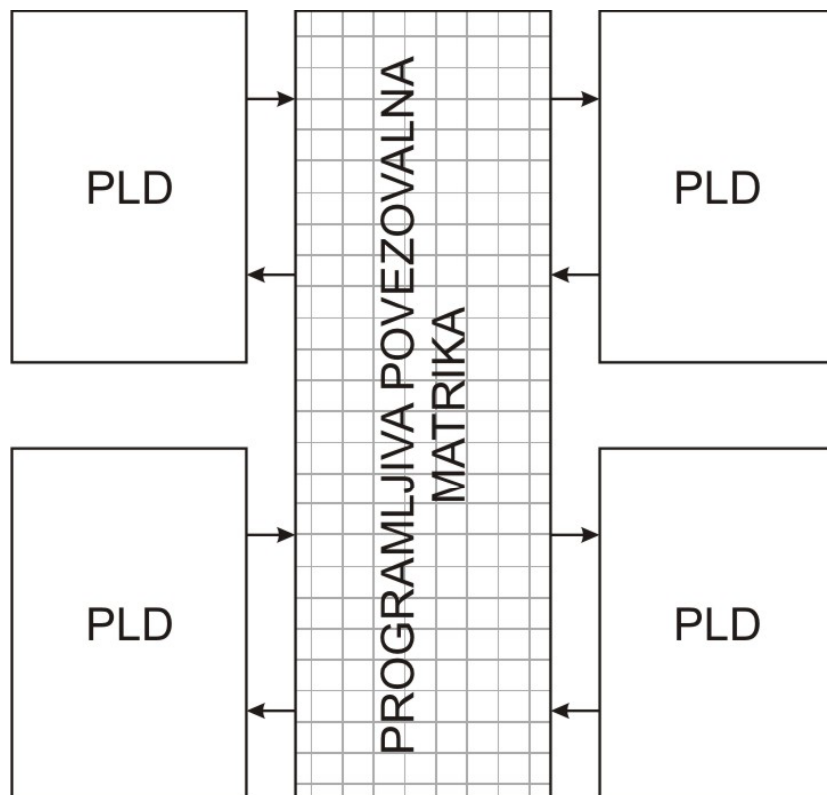


- Stiki povezav določajo končno funkcijo vezja
- Površina vezja se veča s kvadratom št. vhodov
- Nekaj 100 logičnih vrat

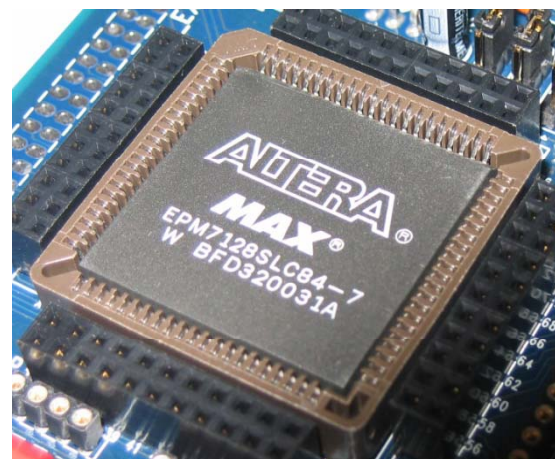




CPLD – Complex Programmable Logic Device

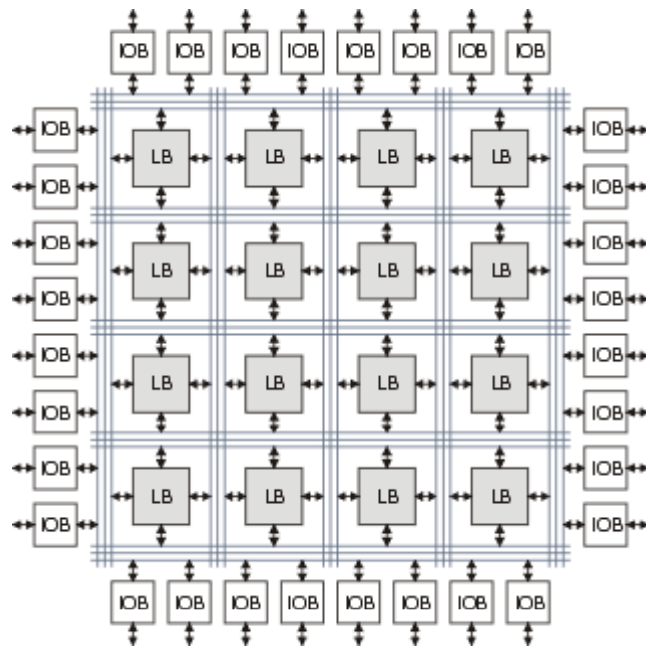


- Več blokov PLD povezanih s programljivo povezovalno matriko
- Izvedba kompleksnih logičnih funkcij
- od 1000 do 20.000 logičnih vrat

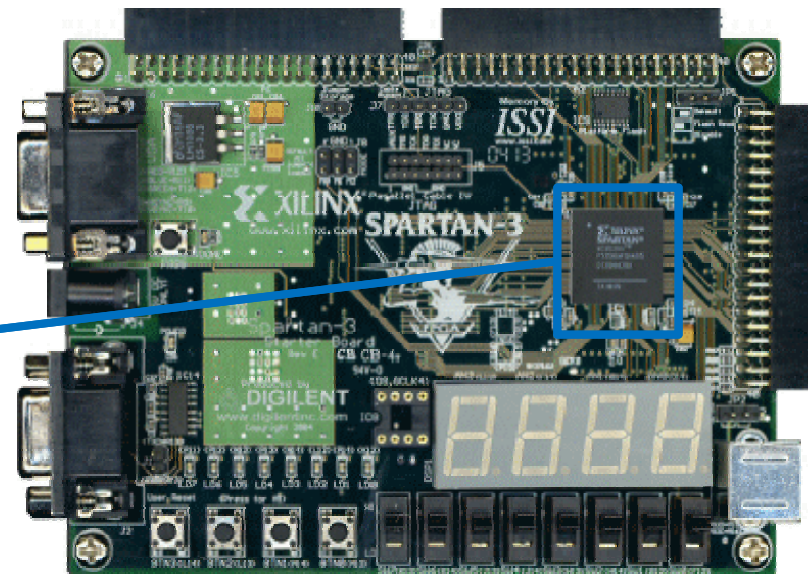




FPGA – Field Programmable Gate Array

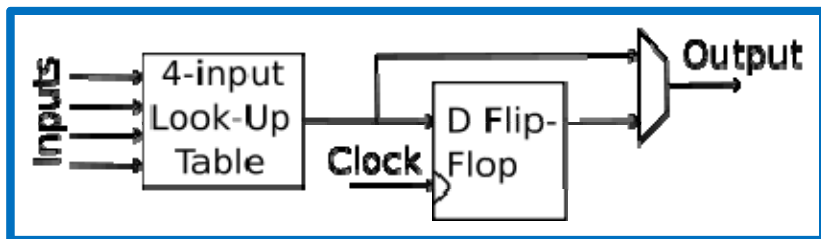
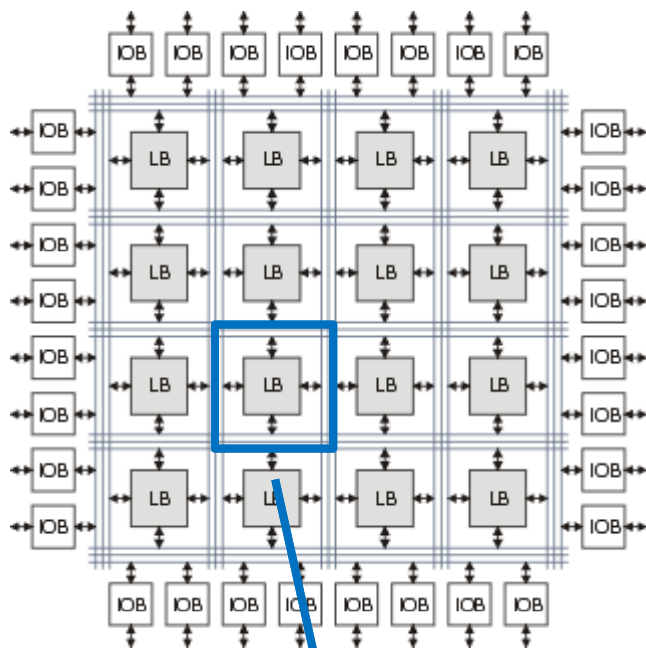


- Matrika logičnih celic (LB) obkroženih z izhodno/vhodnimi celicami
- Celice povezane preko povezovalnega polja
- Nad 20.000 logičnih vrat

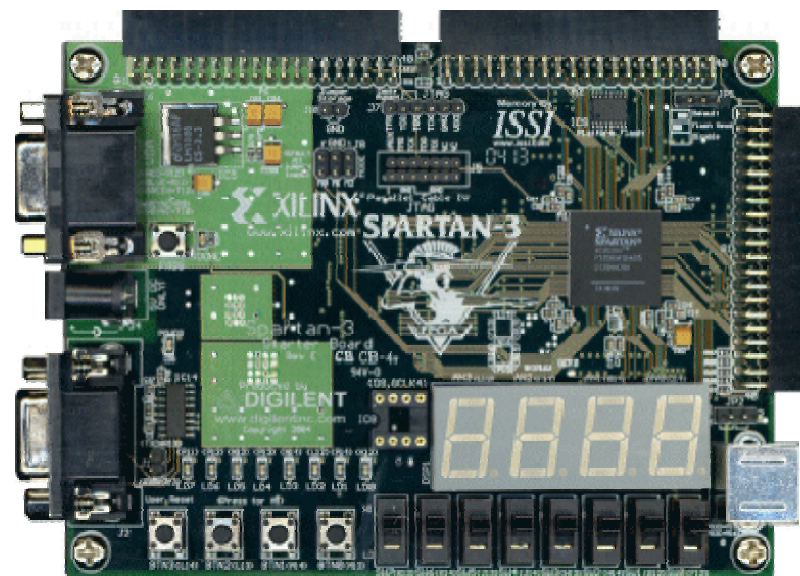




FPGA – Field Programmable Gate Array



- Matrika logičnih celic (LB) obkroženih z izhodno/vhodnimi celicami
- Celice povezane preko povezovalnega polja
- Nad 20.000 logičnih vrat





FPGA – Field Programmable Gate Array

NIVOJI MODELIRANJA

- Tranzistorji
- Logična vrata
- Nivo registrov
- Grafični modeli
- Obnašanje vezja
- Sistemski modeli



FPGA – Field Programmable Gate Array

NIVOJI MODELIRANJA

- Tranzistorji
- Logična vrata
- **Nivo registrov**
- Grafični modeli
- Obnašanje vezja
- Sistemski modeli



```
REG: process (clk, reset)
begin
  if reset='1' then
    stanje <= mir;
  elsif clk'event and clk='1' then
    stanje <= naslednje;
  end if;
end process;

PREHODI: process (stanje, T)
begin
  case stanje is
    when mir =>
      izhod <= '0';
      if T = '1' then
        naslednje <= impulz;
      else
        naslednje <= mir;
      end if;
  end case;
end process;
```

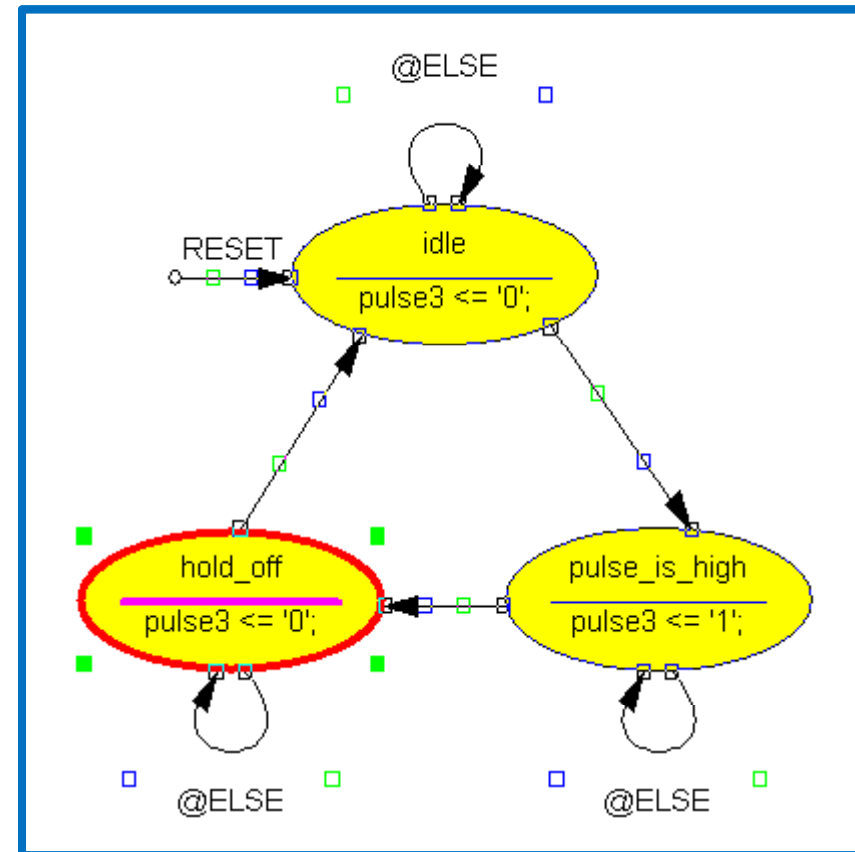
VHDL koda



FPGA – Field Programmable Gate Array

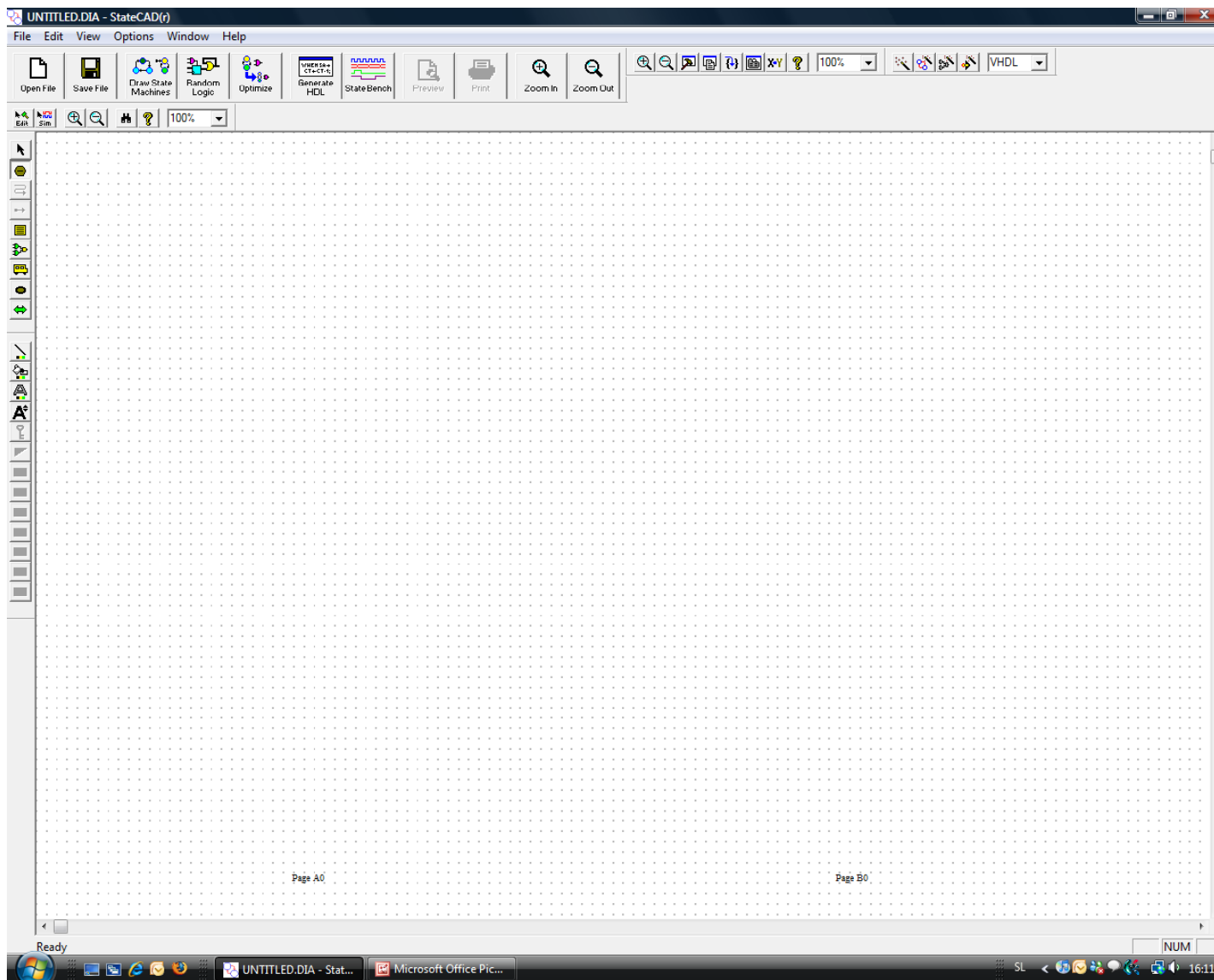
NIVOJI MODELIRANJA

- Tranzistorji
- Logična vrata
- Nivo registrov
- **Grafični modeli**
- Obnašanje vezja
- Sistemski modeli



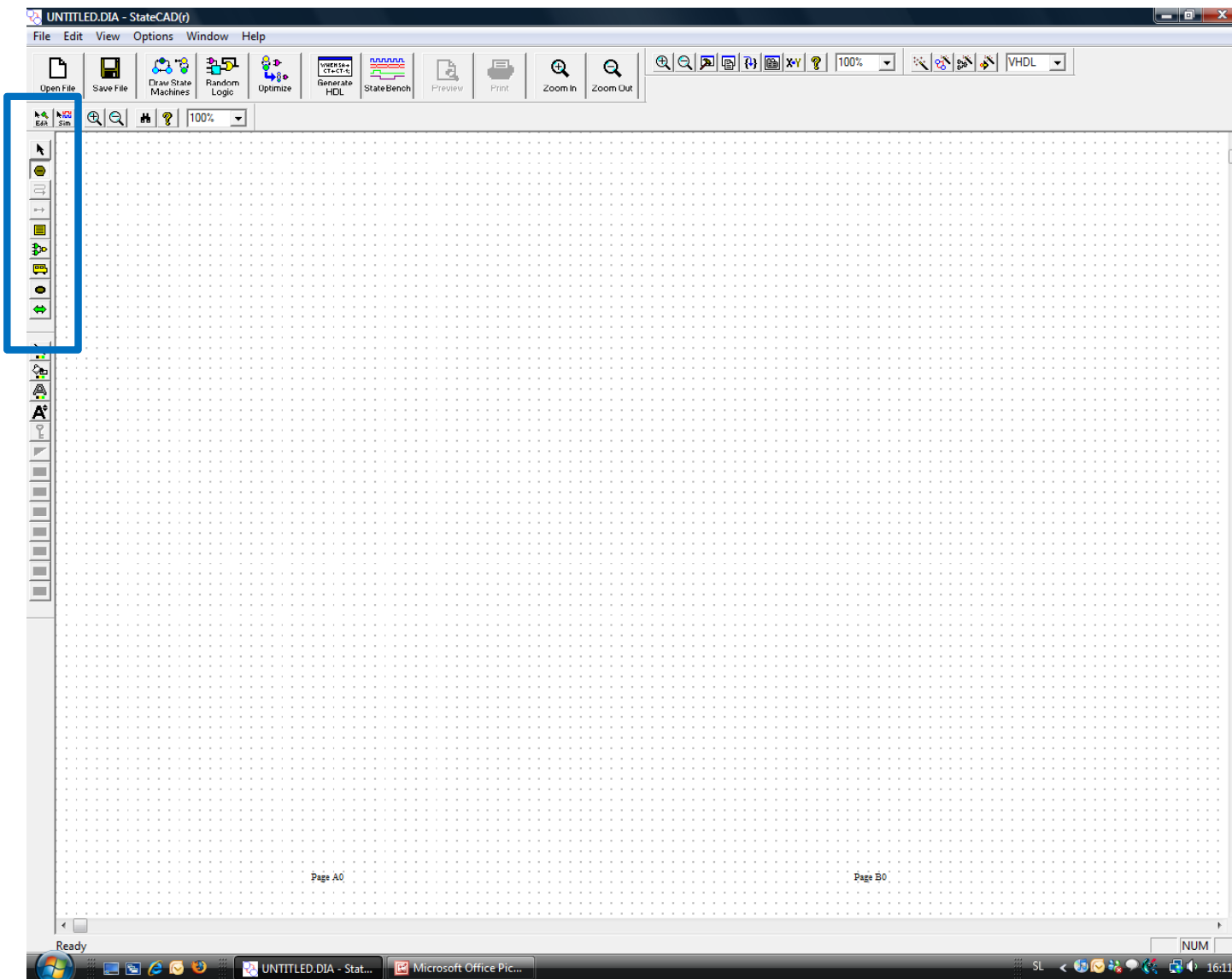


StateCAD



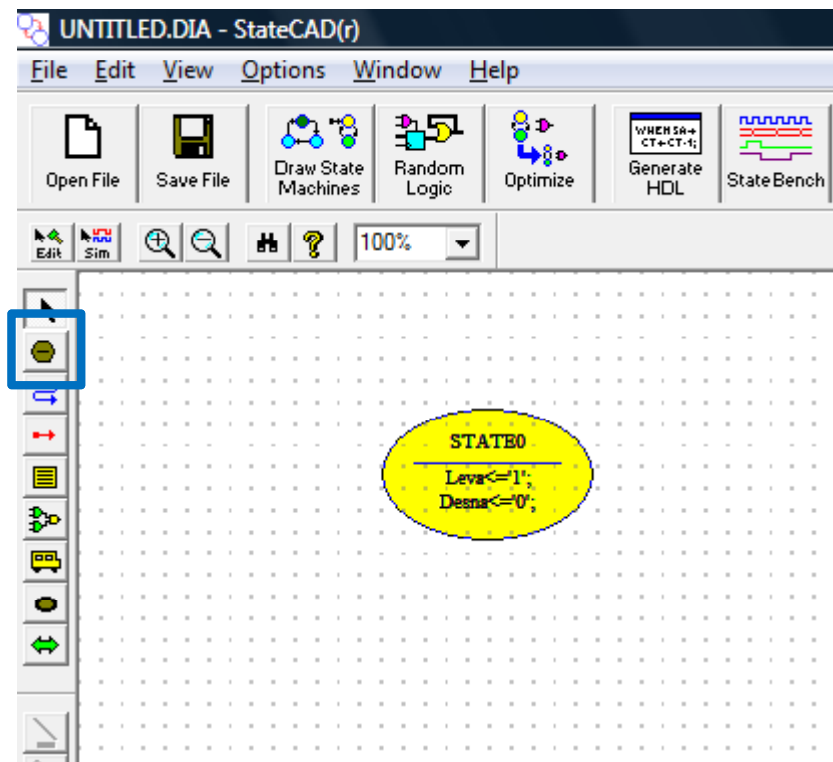


StateCAD



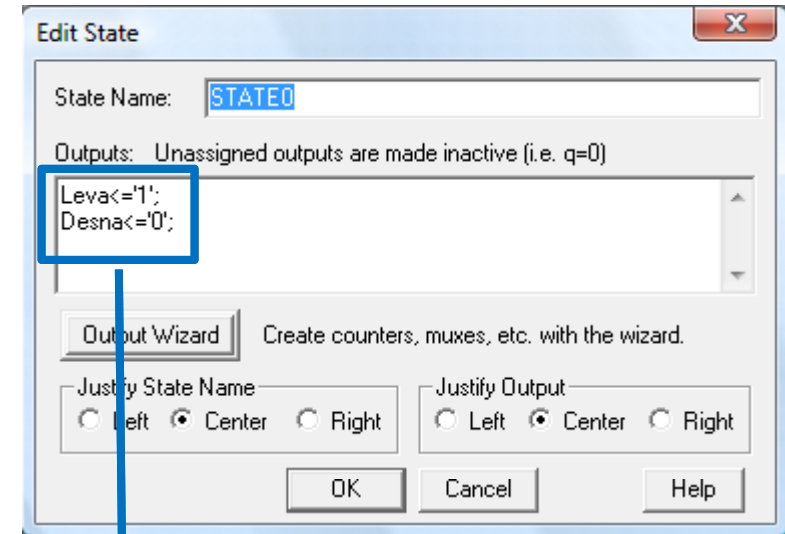
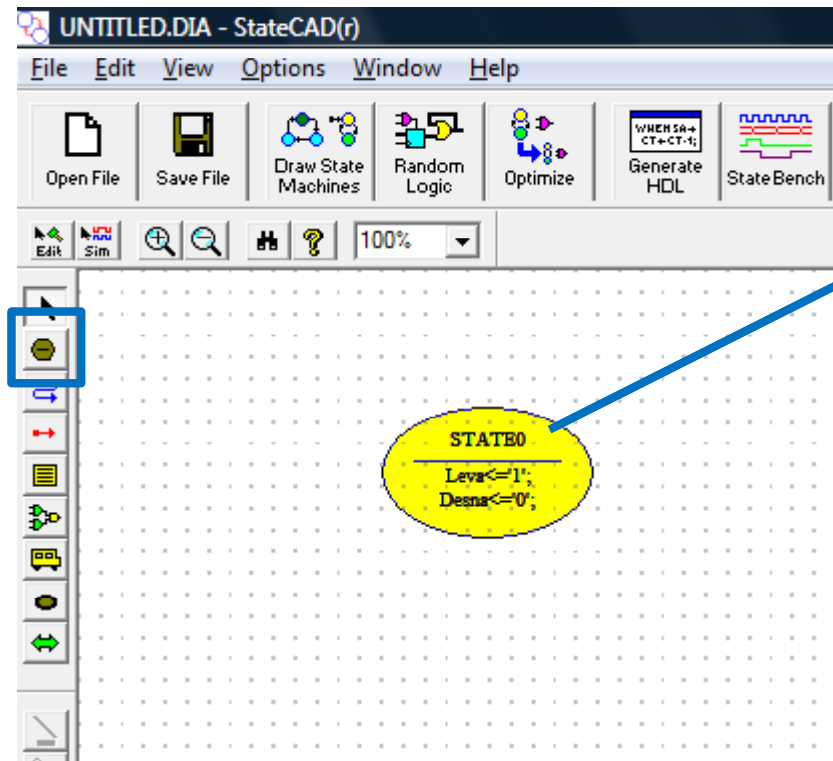


StateCAD – Definiranje stanj





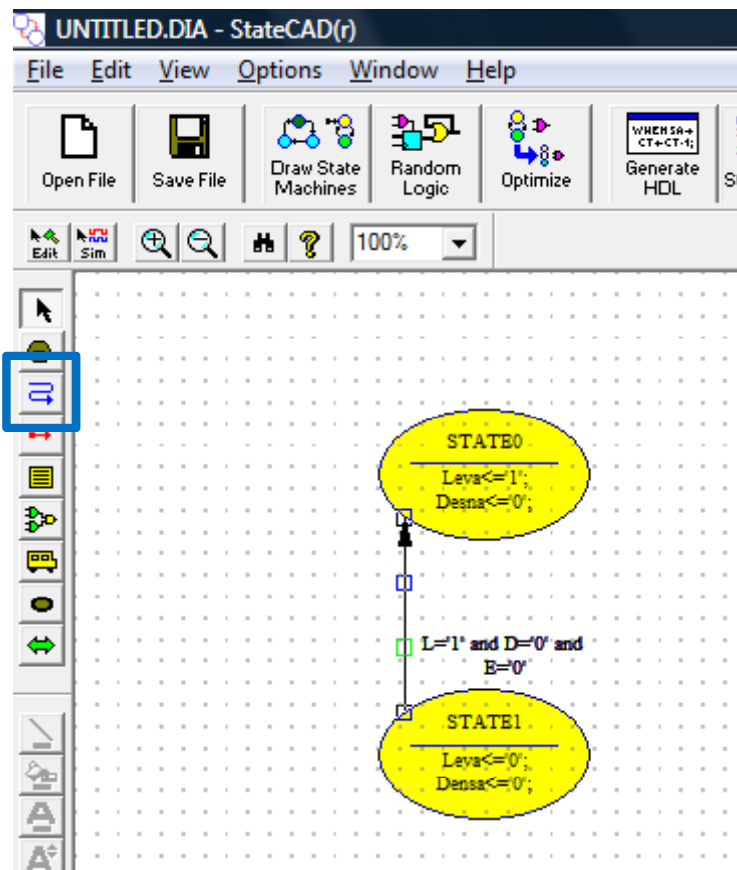
StateCAD – Definiranje stanj



**Leva <= '1';
Desna <= '0';**

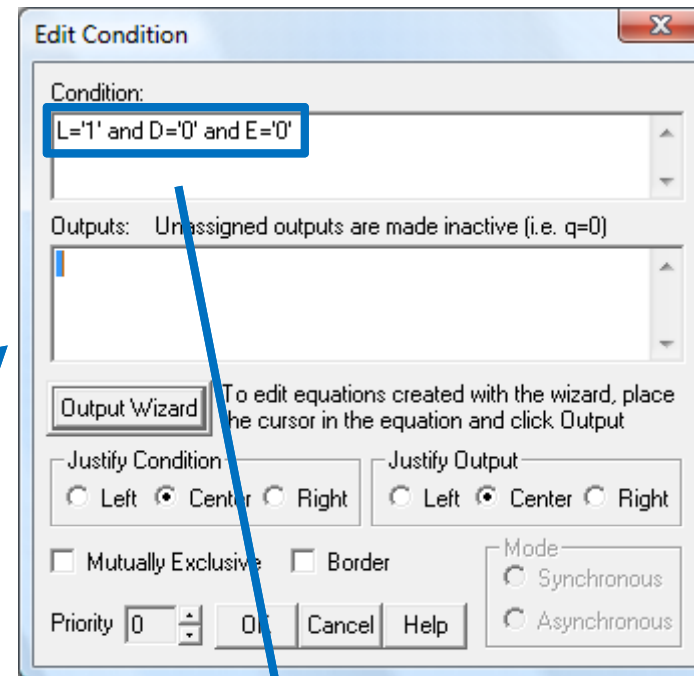
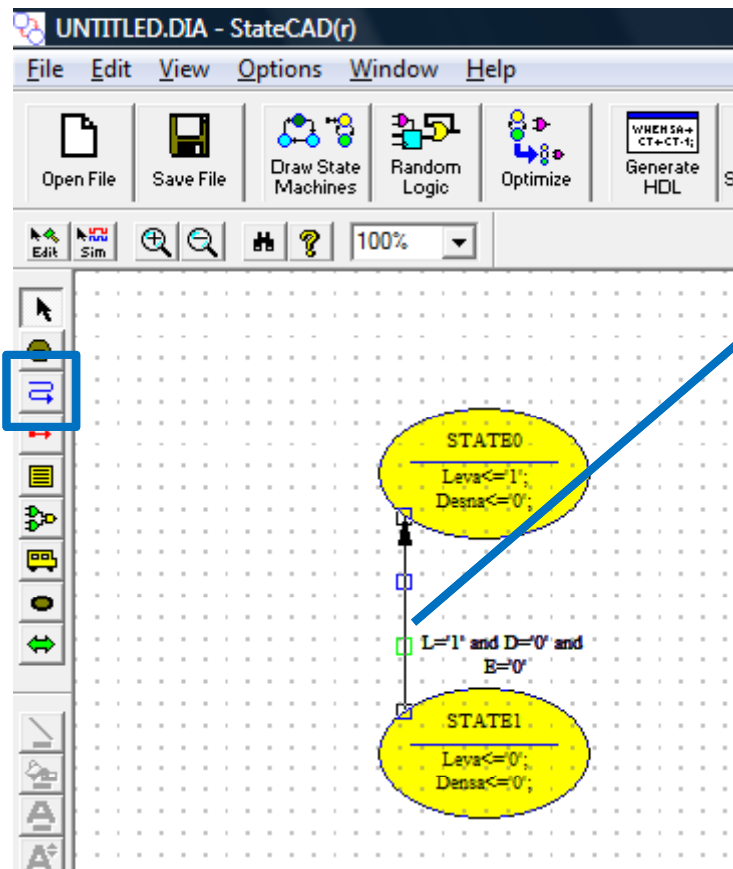


StateCAD – Definiranje prehodov





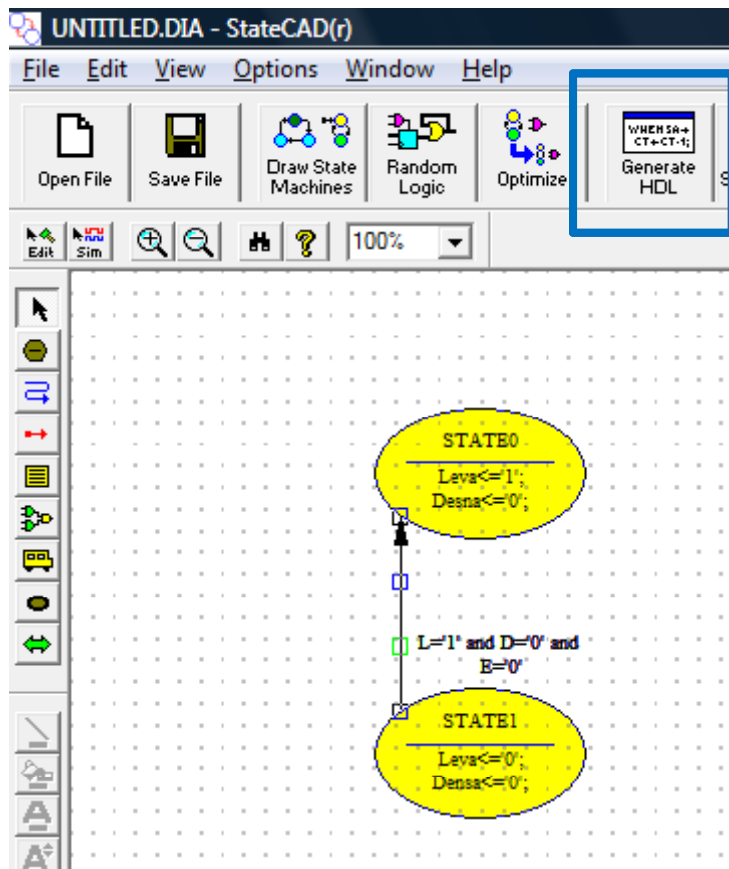
StateCAD – Definiranje prehodov



L='1' and D='0' and E='0'



StateCAD – Generiranje VHDL kode



Ko končamo z risanjem diagrama prehajanja stanj, ga moramo pretvoriti v VHDL kodo.



Xilinx ISE – Programiranje Xilinx vezja

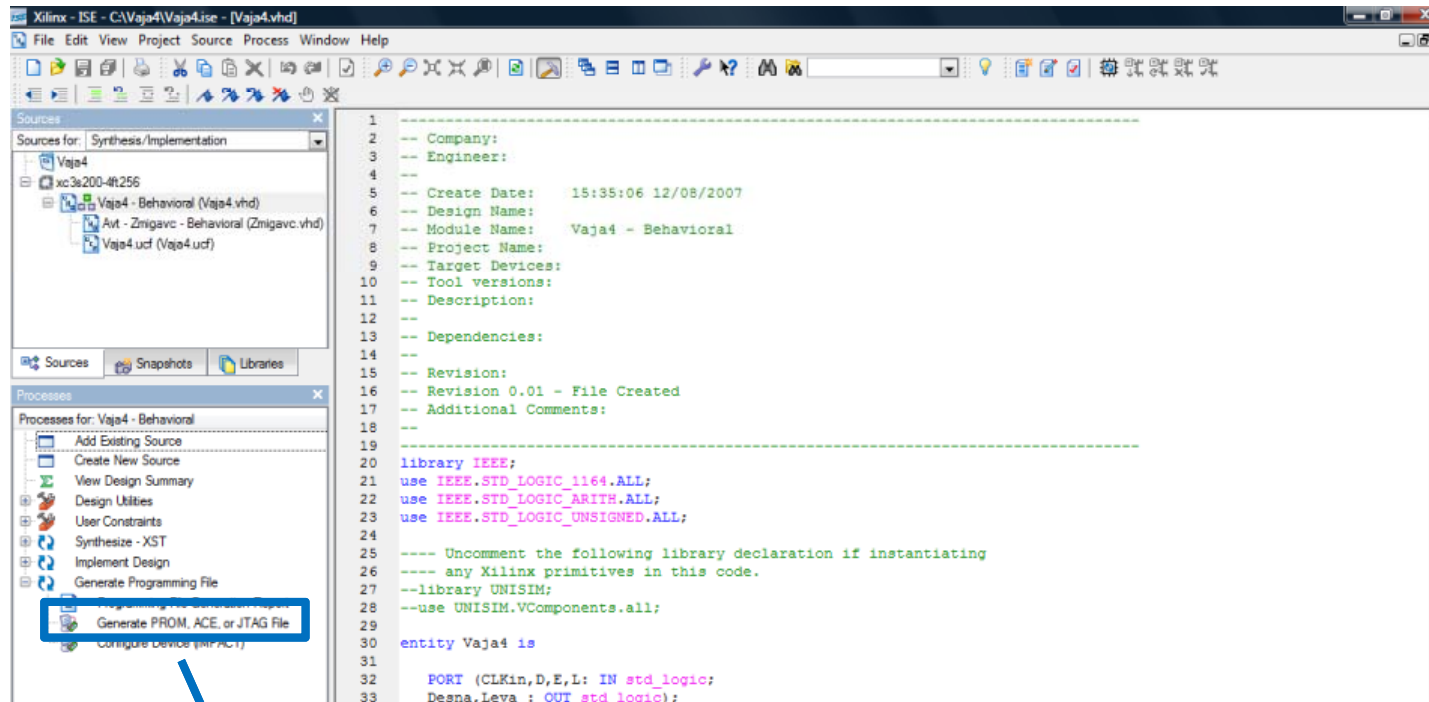
The screenshot displays the Xilinx ISE software interface. The main window shows a VHDL code editor with the following code:

```
1  -----  
2  -- Company:  
3  -- Engineer:  
4  --  
5  -- Create Date:    15:35:06 12/08/2007  
6  -- Design Name:  
7  -- Module Name:    Vaja4 - 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 Vaja4 is  
31  
32     PORT (CLKin,D,E,L: IN std_logic;  
33          Desna,Leva : OUT std_logic);  
34  
35 end Vaja4;  
36  
37 architecture Behavioral of Vaja4 is  
38  
39     signal stevec : std_logic_vector (26 downto 0);  
40     signal ura : std_logic;  
41  
42     component Zmigavc  
43     PORT (CLK,D,E,L: IN std_logic;  
44          Desna,Leva : OUT std_logic);  
45
```

The interface includes a 'Sources' panel on the left showing the project structure for 'Vaja4', including files like 'Vaja4.vhd' and 'Zmigavc.vhd'. Below it is the 'Processes' panel with options like 'Synthesize - XST' and 'Implement Design'. At the bottom, a 'Transcript' window shows messages such as 'Started: "Launching ISE Text Editor to edit Zmigavc.vhd"'.



Xilinx ISE – Programiranje Xilinx vezja



Generate PROM, ACE or JTAG File



Xilinx ISE – Programiranje Xilinx vezja

```
1  -- C:\VAJA4\ZMIGAVC.vhd
2  -- VHDL code created by Xilinx's StateCAD 9.1i
3  -- Sat Dec 08 17:29:05 2007
4
5  -- This VHDL code (for use with Synopsys) was generated using:
6  -- one-hot state assignment with boolean code format.
7  -- Minimization is enabled, implied else is enabled,
8  -- and outputs are speed optimized.
9
10 LIBR
11 USE
12
13 LIBR
14 USE
15
16 ENTI
17 P
18
19 END;
20
21 ARCH
22 -- S
23 S
24
25 S
26 BEGI
27 P
28
29 B
30
31
32
33
34
35
36
37
38 E
39
40 B
41 B
42
43 IF (( E='0' AND L='1' AND D='1' AND (STATE0='1')) OR ( E='0' AND D='0' AN
44 L='0' AND (STATE0='1')) OR ( L='0' AND E='0' AND (STATE1='1')) OR (
45 AND (STATE2='1')) OR ( D='0' AND E='0' AND (STATE3='1')) THEN
```

Please select an action from the list below

- Configure devices using Boundary-Scan (JTAG)
- Prepare a PROM File
- Prepare a System ACE File
- Prepare a Boundary-Scan File
- Configure devices

Automatically connect to a cable and identify Boundary-Scan chain

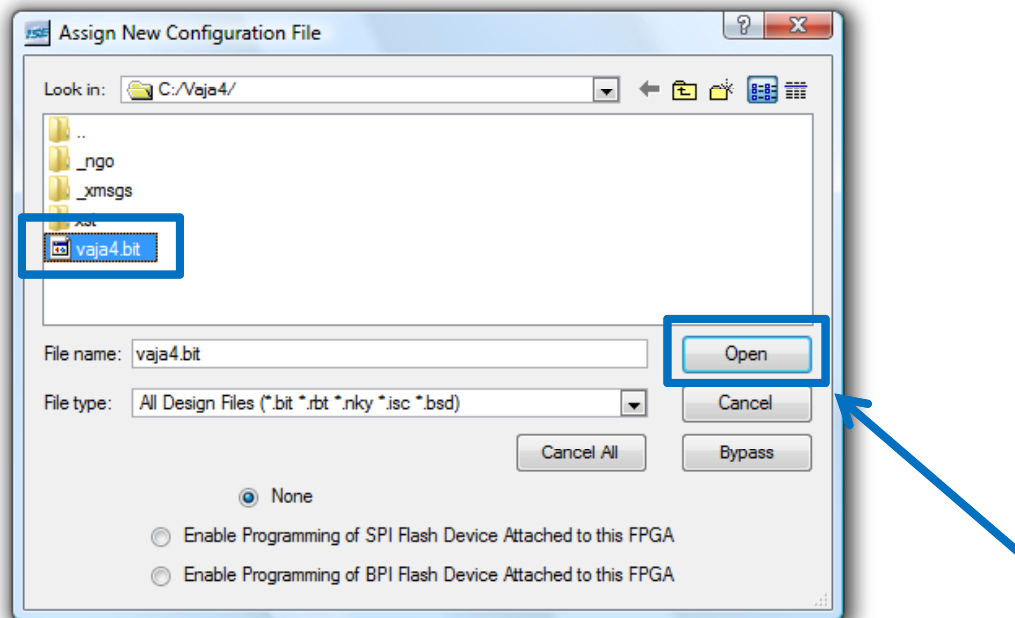
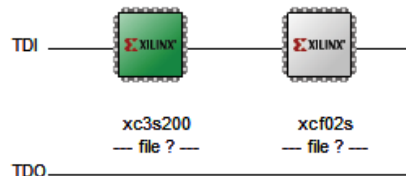
SVF

using Slave Serial mode

< Back Finish Cancel

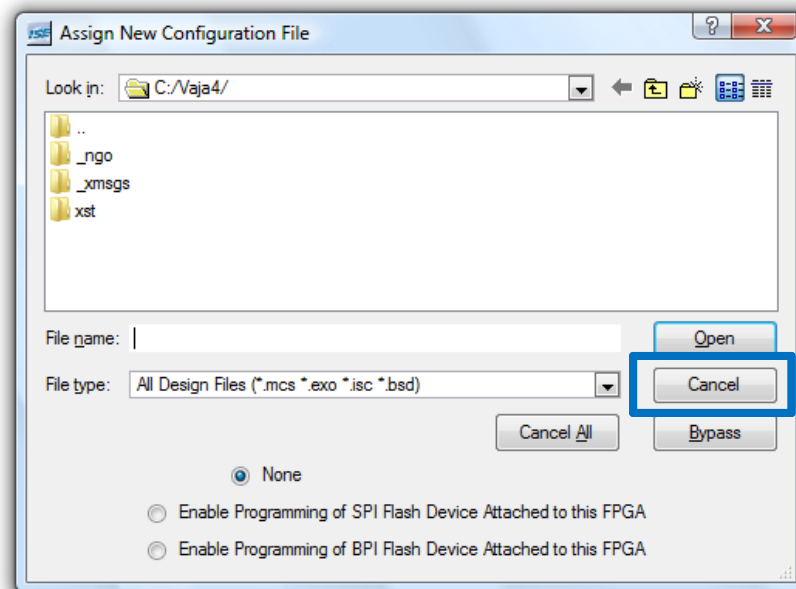
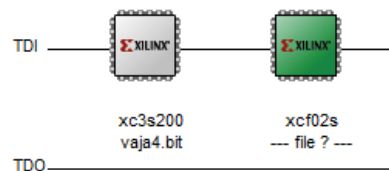


Xilinx ISE – Programiranje Xilinx vezja





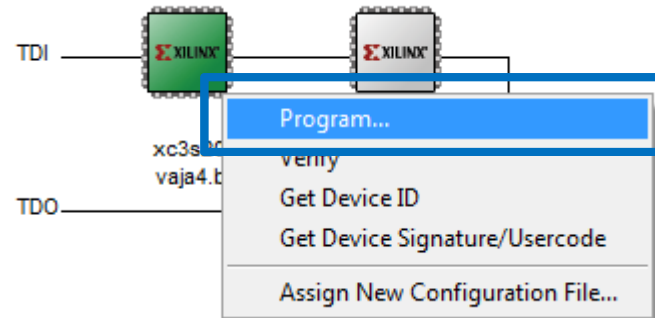
Xilinx ISE – Programiranje Xilinx vezja





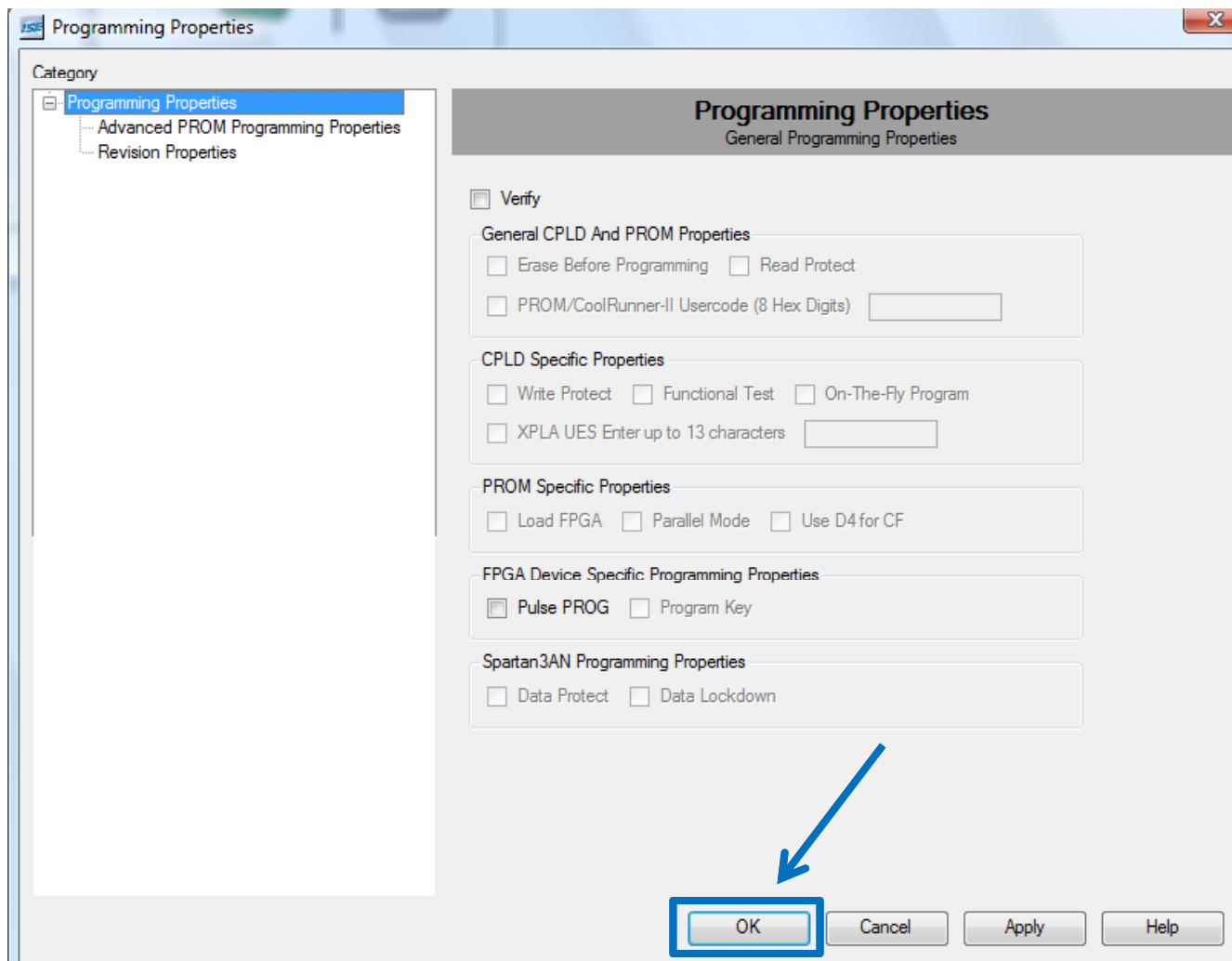
Xilinx ISE – Programiranje Xilinx vezja

Desni klik na vezje xc3s200





Xilinx ISE – Programiranje Xilinx vezja





Poročilo

- ❑ Besedilo vaje
- ❑ Tabela prehajanja stanj
- ❑ Diagram prehajanja stanj (le osnovni prehodi)
- ❑ Predlagajte minimalno možno realizacijo z logičnimi vrati in T pomnilnimi celicami