

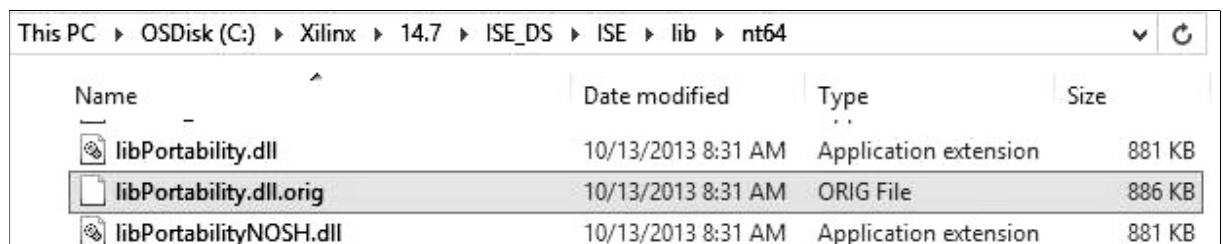
TUTORIAL – wstępny projekt w języku VHDL




(wersja on-line)

Dla systemu operacyjnego Windows 10 należy przed uruchomieniem systemu projektowego wykonać poniższe zmiany:

ISE 14.7 64-bit - Turning off SmartHeap:

- 1) Navigate to the following ISE install directory: <install_path>\Xilinx\14.7\ISE_DS\ISE\lib\nt64\
- 2) Rename the file "libPortability.dll" to "libPortability.dll.orig".
- 3) Copy the "libPortabilityNOSH.dll" file to the same folder, and rename it to "libPortability.dll".



Name	Date modified	Type	Size
 libPortability.dll	10/13/2013 8:31 AM	Application extension	881 KB
 libPortability.dll.orig	10/13/2013 8:31 AM	ORIG File	886 KB
 libPortabilityNOSH.dll	10/13/2013 8:31 AM	Application extension	881 KB

- 4) Repeat steps 1-3 in the following folder: <install_path>\Xilinx\14.7\ISE_DS\common\lib\nt64\

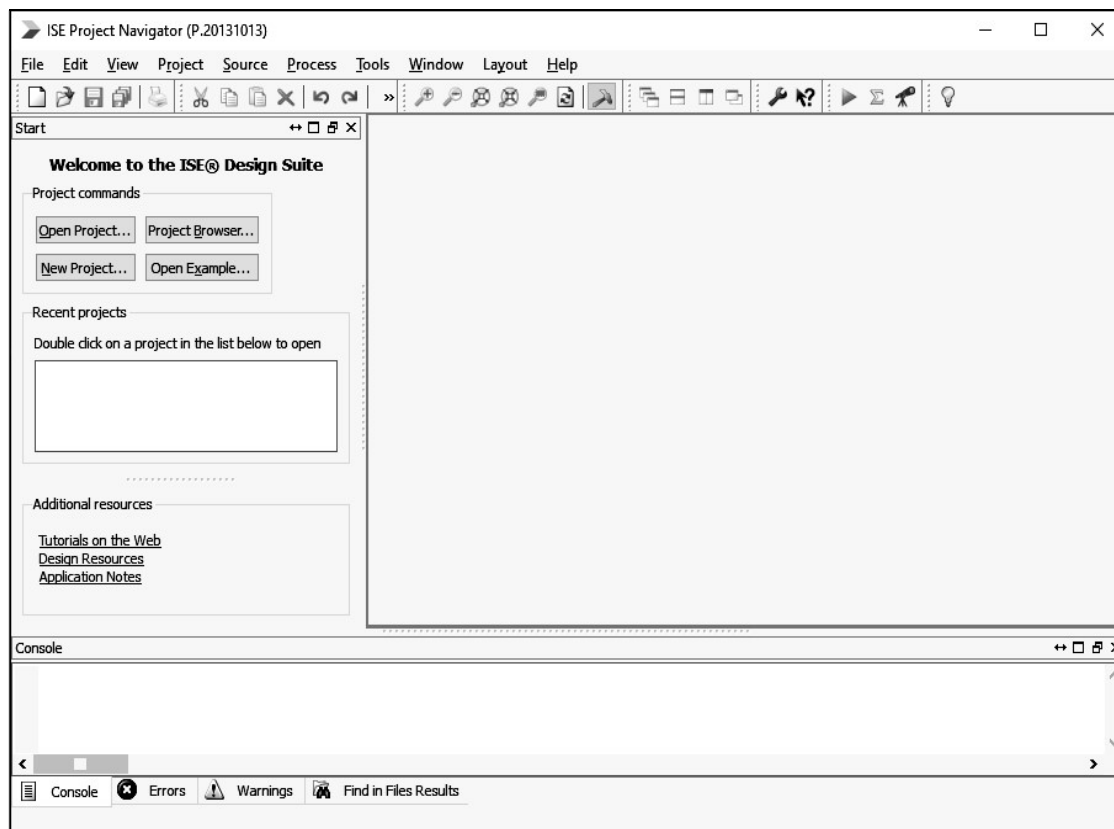
The above steps substitute the original "libPortability.dll" with a "libPortability.dll" file that has SmartHeap disabled, the NOSmartHeap (NOSH) version.

This does not negatively impact the operation of the tools, and should successfully work around the ISE 14.7 crash documented above.

Uruchomienie systemu projektowego ISE Design Suite 14.7

Na wstępie należy utworzyć katalog dla projektu...

System uruchamia się poprzez dwukrotne kliknięcie lewym klawiszem myszki na ikonie **ISE Design Suite**, lub wybierając z rozwijanego menu: **Start** → **Xilinx Design Tools** → **64-bit Project Navigator**. Po uruchomieniu programu pojawia się nawigator projektu **ISE Project Navigator**.



Plik licencyjny

W czasie pierwszego uruchomienia systemu projektowego należy dołączyć plik licencyjny poprzez: **Help** → **Manage License...**, w kolejnym oknie **Xilinx License Configuration Manager** wybrać zakładkę **Manage Licenses**, nacisnąć **Copy License** i wskazać plik licencyjny **Xilinx.lic**.

Utworzenie nowego projektu typu HDL

1. Wybrać **File** → **New Project...** uruchamiając aplikację **New Project Wizard**
2. W polu **Location**: ustawić utworzony folder dla projektu
3. W polu **Name**: wpisać nazwę projektu, np. **witek**
4. Sprawdzić czy w polu **Top-level source type**: jest wybrana opcja **HDL**
5. Nacisnąć przycisk **Next**
6. Wypełnić pola okna **Project Settings**, gdzie najważniejsze ustawienia to **Family** → Spartan3 , **Device** → XC3S200 , **Package** → FT256 oraz **Preferred Language** → VHDL

New Project Wizard

← Project Settings
Specify device and project properties.

Select the device and design flow for the project

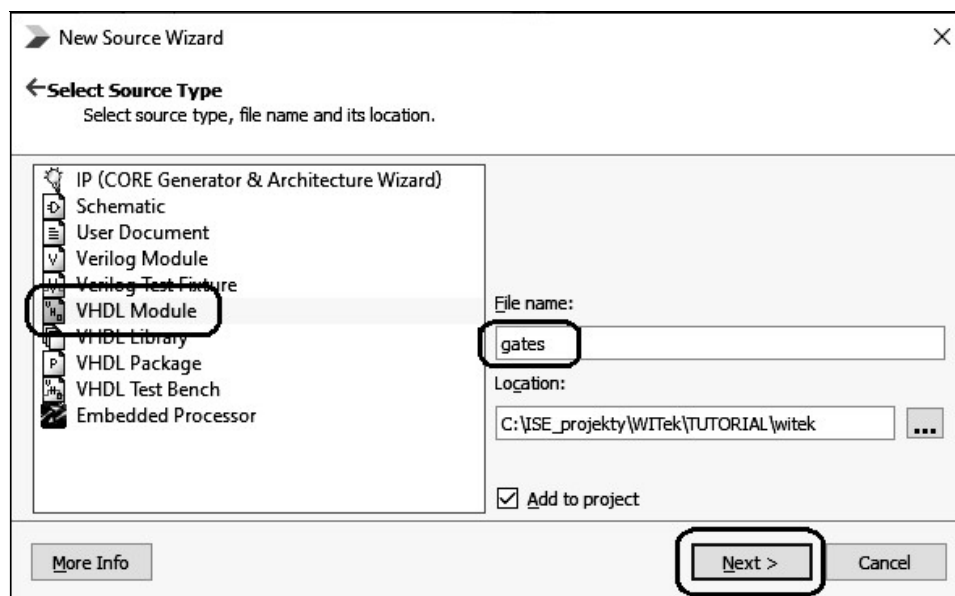
Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3
Device	XC3S200
Package	FT256
Speed	-
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	Isim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store All Values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info < Back Next > Cancel

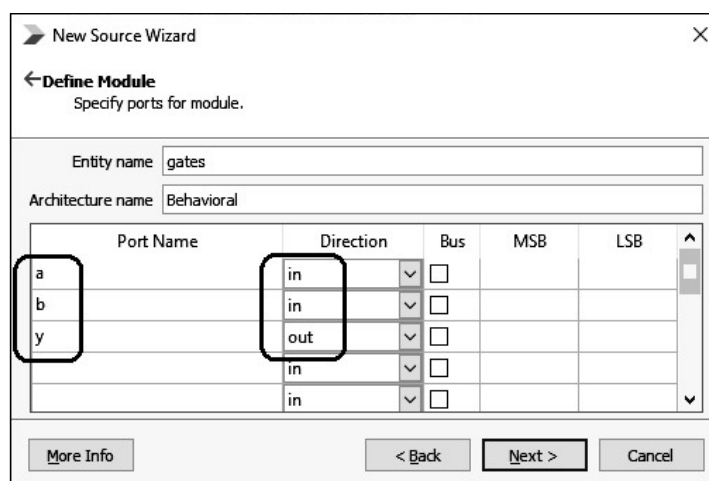
7. Nacisnąć przycisk **Next**
8. W oknie **Project Summary** nacisnąć **Finish**

Opis działania układu w języku VHDL

1. Wybrać **Project** → **New Source...** uruchamiając **New Source Wizard**
2. W oknie **Select Source Type** zaznaczyć typ **VHDL Module**
3. W polu **File name:** wpisać nazwę **gates**
4. W polu **Location** powinien być wybrany folder projektu



5. Nacisnąć **Next**
6. W oknie **Define Module** wypełnić odpowiednie kolumny nazwami sygnałów (**Port Name**) i ich parametrami (**Direction**), jak poniżej



7. Nacisnąć **Next**
8. W oknie **Summary** nacisnąć **Finish**
9. W utworzonym pliku **gates.vhd** opisać działanie układu jak poniżej (można usunąć „zielony” tekst :), zapisać plik

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

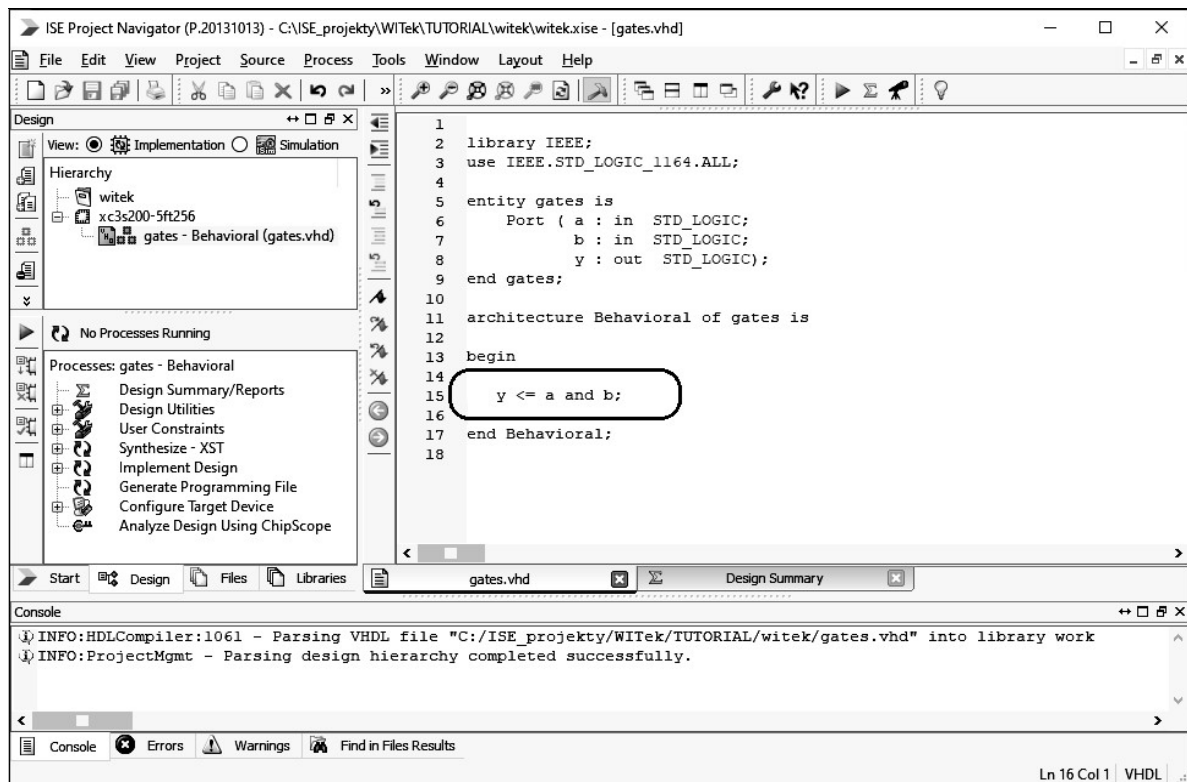
entity gates is
  port( a : in std_logic;
        b : in std_logic;
        y : out std_logic );
end gates;

architecture Behavioral of gates is
begin

  y <= a and b;

end Behavioral;

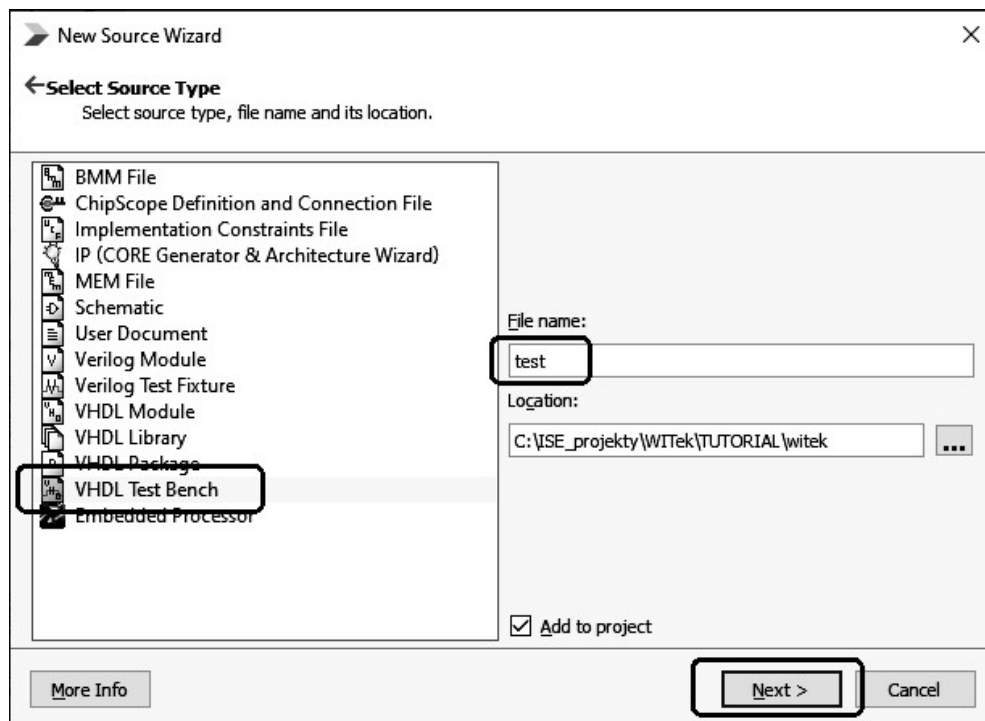
```



10. W celu sprawdzenia poprawności kodu w języku VHDL należy w obszarze **Hierarchy** zaznaczyć plik **gates – Behavioral (gates.vhd)**, następnie w polu **Processes** rozwinąć wiersz **Synthesize – XST** i uruchomić **Check Syntax**, poprawić ewentualne błędy...

Symulacja działania układu

1. Wybrać **Project** → **New Source...** uruchamiając **New Source Wizard**
2. W oknie **Select Source Type** zaznaczyć typ **VHDL Test Bench**



3. W polu **File name:** wpisać nazwę **test**
4. W polu **Location** powinien być wybrany folder projektu...
5. Nacisnąć **Next**
6. W kolejnym oknie **Associate Source** powinien być zaznaczony **gates**, nacisnąć **Next**
7. W oknie **Summary** nacisnąć **Finish**
8. W utworzonym module testowym **test.vhd** można zastosować przykładowy algorytm testowy jak poniżej. Moduł ustala stany na wejściach **a** i **b**, którymi są kolejne kombinacje binarne zmieniane co 50 ns. Ostatnia instrukcja **wait** zatrzymuje symulację.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS

    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT gates
    PORT( a : IN  std_logic;
          b : IN  std_logic;
          y : OUT std_logic );
    END COMPONENT;

    -- Inputs
    signal a : std_logic := '0';
    signal b : std_logic := '0';

```

```
-- Outputs
signal y : std_logic;

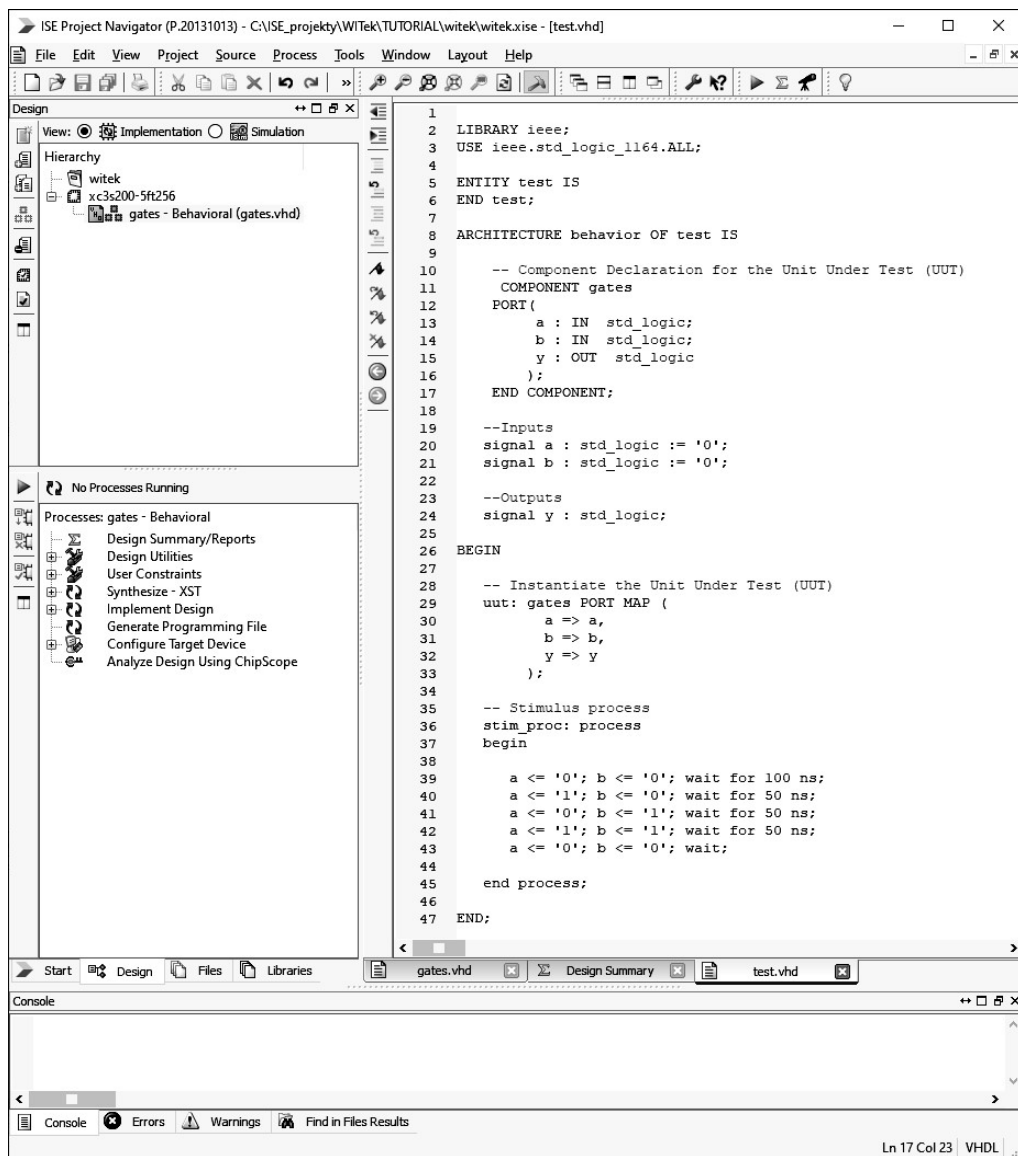
BEGIN
  -- Instantiate the Unit Under Test (UUT)
  uut: gates PORT MAP ( a => a, b => b, y => y );

  -- Stimulus process
  stim_proc: process
  begin

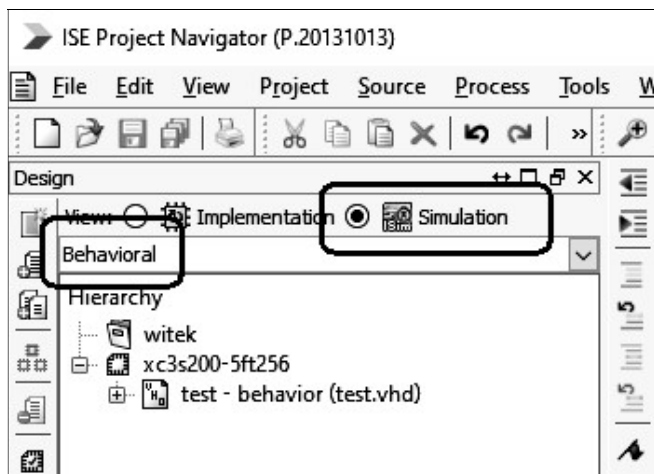
    a<='0'; b<='0'; wait for 100 ns;
    a<='0'; b<='1'; wait for 50 ns;
    a<='1'; b<='0'; wait for 50 ns;
    a<='1'; b<='1'; wait for 50 ns;
    a<='0'; b<='0'; wait;

  end process;

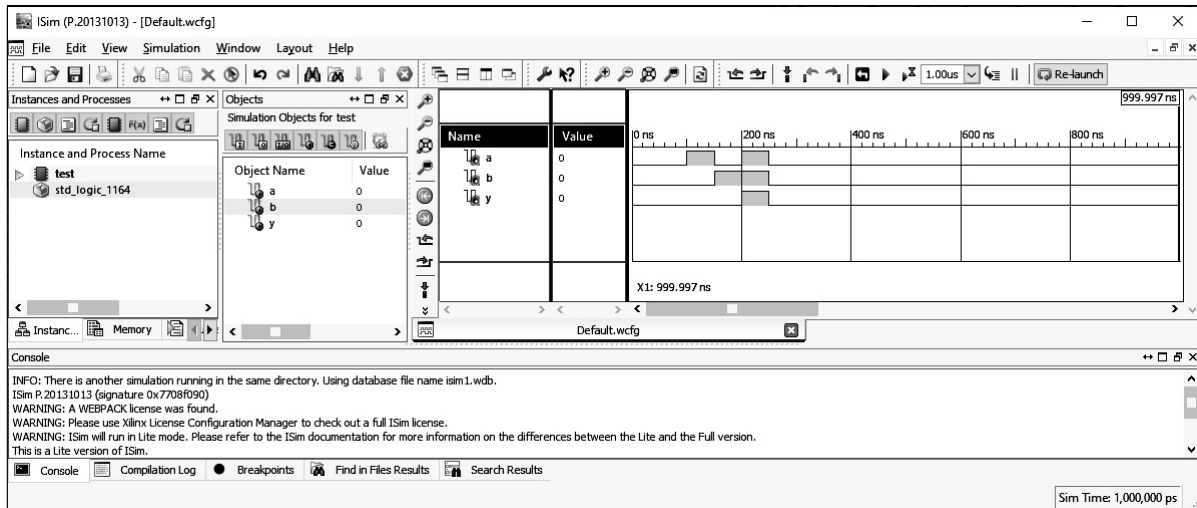
END;
```



9. Następnie w polu **View**: zaznaczyć **Simulation**, a w rozwijanym menu wybrać **Behavioral**

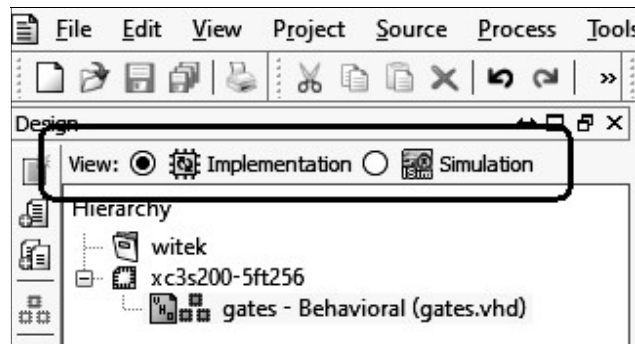


10. W obszarze **Hierarchy** zaznaczyć plik **test – behavior (test.vhd)**
 11. W polu **Processes** rozwinąć **ISim Simulator** i uruchomić **Behavioral Check Syntax**, poprawić ewentualne błędy...
 12. Uruchomić symulację poprzez **Simulate Behavioral Model**
 13. Wynik symulacji zaobserwować w postaci przebiegów sygnałów...

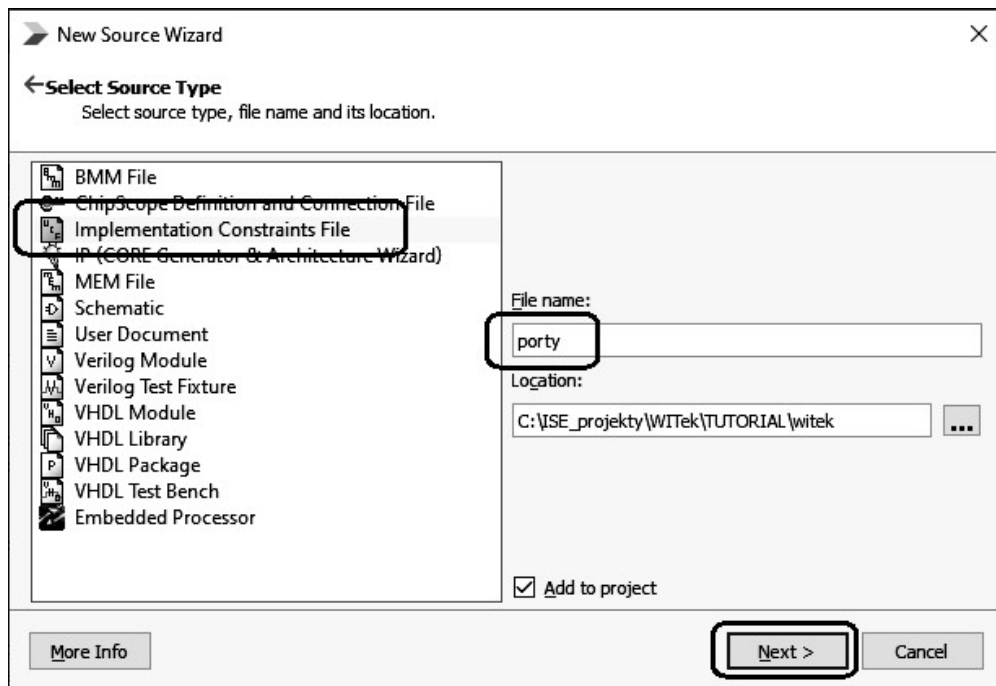


Implementacja projektu w układzie FPGA

1. W polu **View:** wybrać **Implementation** (wyjście z symulacji, powrót do projektowania...)



2. Wybrać **Project** → **New Source...** uruchamiając **New Source Wizard**
3. W oknie **Select Source Type** zaznaczyć typ **Implementation Constraints File**



4. W polu **File name:** wpisać nazwę **porty**
5. W polu **Location** powinien być wybrany folder projektu
6. Naciśnąć **Next**, a następnie **Finish**
7. Do projektu został dołączony plik tekstowy **porty.ucf**
(rozwinąć wiersz **gates – Behavioral (...)** w polu **Hierarchy**)
8. Jako zawartość pliku **porty.ucf** wpisać poniższy zapis, będący przyporządkowaniem końcówek układu FPGA do sygnałów projektu, zapisać i zamknąć plik

```
NET "a" LOC="F12";
NET "b" LOC="G12";
NET "y" LOC="K12";
```

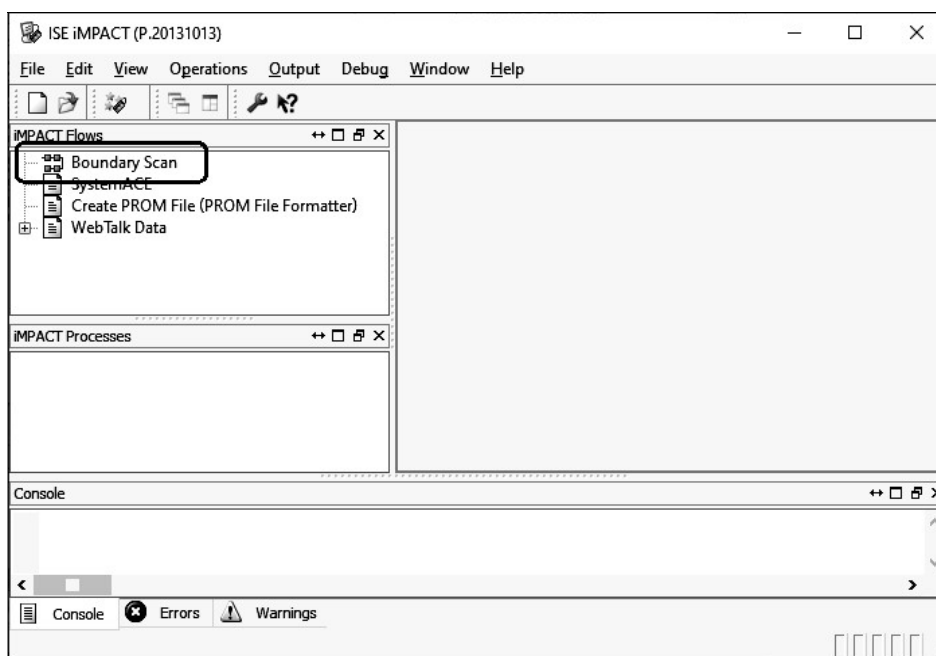
9. W obszarze **Hierarchy** zaznaczyć wiersz **gates – Behavioral (...)**
10. Następnie w polu **Processes** uruchomić **Generate Programming File**

Uwaga!

Podczas pierwszej implementacji po uruchomieniu systemu projektowego **ISE Design Suite** należy ponownie uruchomić **Generate Programming File**, wybierając prawy przycisk myszki **Rerun All**. Taka operacja wynika z błędnego przypisywania końcówek układu scalonego do sygnałów projektu, ale po ponownej re-implementacji problem „znika”.

Konfiguracja układu FPGA

1. **Sprawdzić podłączenie płytki testowej!**
2. Następnie w polu **Processes** rozwinąć wiersz **Configure Target Device** i uruchomić **Manage Configuration Project (iMPACT)**, jeśli pojawi się ostrzeżenie – nacisnąć **OK**
3. Pojawia się aplikacja **ISE iMPACT**, w polu **iMPACT Flows** uruchomić **Boundary Scan**



4. Następnie wybrać **File** → **Initialize Chain**
5. Jeśli pojawi się okno **Assign New Configuration Files Query Dialog**, to nacisnąć **Yes**
6. Następnie pojawia się okno, w którym należy wskazać katalog projektu i wybrać plik **gates.bit**, nacisnąć **Open**
7. W kolejnym oknie nacisnąć **Cancel**
8. W oknie **Device Programming Properties - ...** zatwierdzić proponowane opcje konfigurowania układów FPGA i PROM poprzez **OK**
9. Następnie nacisnąć prawym klawiszem myszki na symbolu układu FPGA i wybrać **Program** z listy poleceń. Konfigurowanie w toku...
10. Sprawdzić praktycznie działanie projektu...

HARMONOGRAM

LAB 1	1) Projekt wstępny z instrukcji laboratoryjnej. 2) Projekt zawierający dwie bramki n -wejściowe typu OR i AND. 3) Konwerter 3-bitowego kodu binarnego na kod „1 z 8” (aktywny stan 1 lub 0). 4) Sumator dwóch liczb 4-bitowych.
LAB 2	1) 25-bitowy licznik binarny z kasowaniem asynchronicznym. (zastosować sygnał zegarowy z płyty projektowej, osiem najbardziej znaczących bitów licznika wyprowadzić na diody LED) 2) 8-bitowy licznik Johnsona. (jako dzielnik częstotliwości sygnału zegarowego zastosować N -bitowy licznik binarny, dobrać N) 3) 8-bitowy licznik pierścieniowy typu „krążąca jedynka” . (zastosować dzielnik częstotliwości...) 4) Zmodyfikować punkt 3) tak, aby za pomocą suwaka wybierać pomiędzy „krążącą jedynką” a „krążącym zerem”
LAB 3	Projekt cyfrowego układu do pomiaru wielkości fizycznej (czas, częstotliwość) na przykładzie projektu stopera elektronicznego. Odczyt wyniku pomiaru przy użyciu wyświetlacza 7-segmentowego.