

Quartus

Rafał Walkowiak

IIIn PP

Listopad 2017

Altera Quartus

- Narzędzie projektowe dla FPGA i CPLD
- Umożliwia:
 - wprowadzenie projektu,
 - syntezę logiczną i symulację funkcjonalną,
 - przydział do układów logicznych i ich łącznie,
 - symulację czasową,
 - analizę czasową,
 - zarządzanie użytkowaniem mocy i
 - programowanie układu FPGA

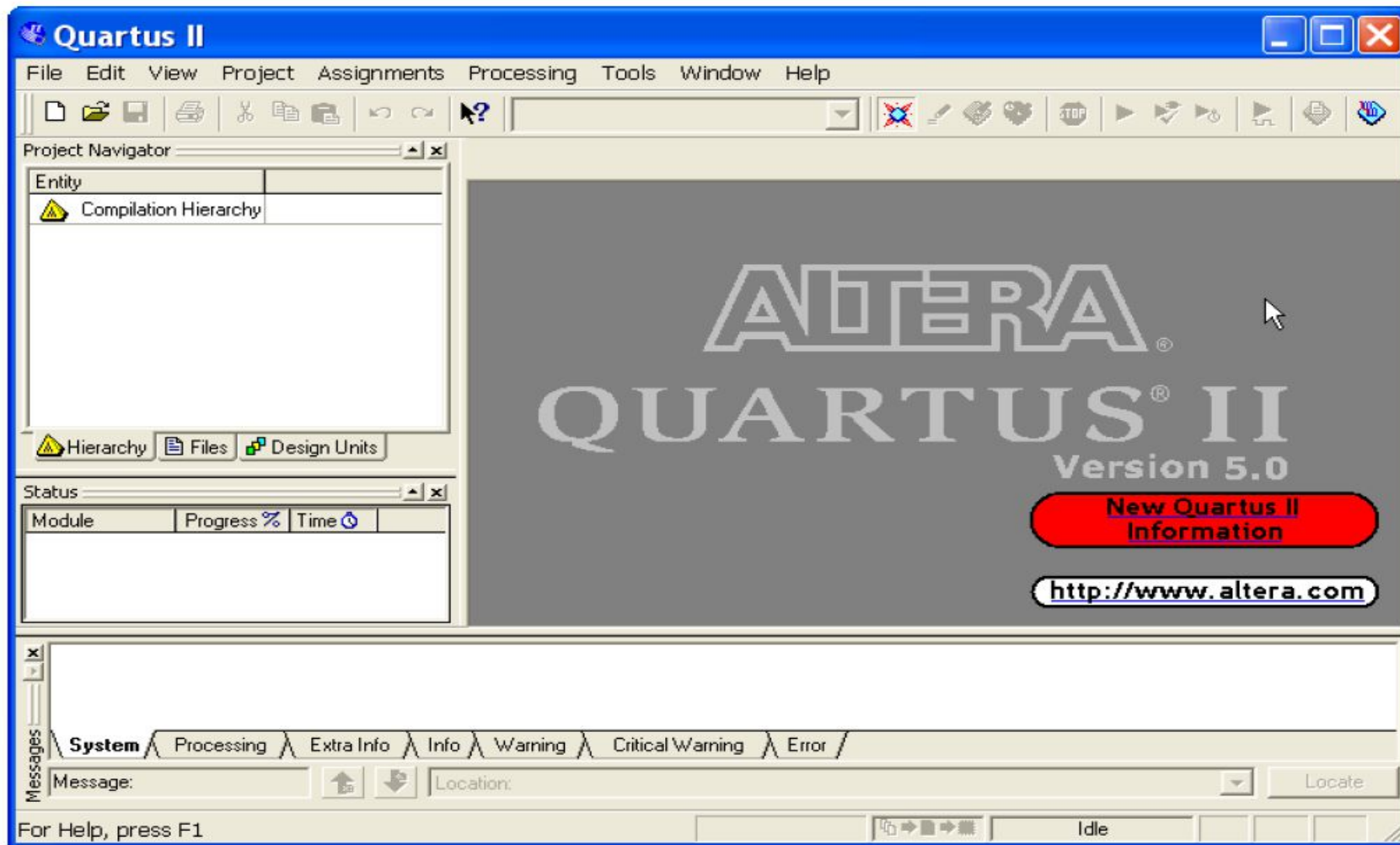
Altera Quartus

- **Używamy (2015/2016) QUARTUS wersji 13.0 sp1 gdyż współpracuje z symulatorem Altera University Program**
- **Możliwość darmowego pobrania programów (cele edukacyjne) ze strony altera.com**

Quartus interfejs użytkownika

- Okno główne: pliki, raporty , inne okna
- Projekt nawigator: hierarchia, pliki projektu, skróty komend
- Okno statusu: status przetwarzania zadań
- Okno komunikatów: informacje, błędy, ostrzeżenia
- Dynamicznie zmieniające się menu

Quartus okno główne



Projekt w Quartus

Praca zorganizowana w ramach projektów.

Projekt składa się z:

- plików projektowych i bibliotek,
- informacji o parametrach i ograniczeniach,
- hierarchii projektu i informacjach o wydaniach.

Etapy procesu projektowania w PLD

1. Wprowadzanie projektu

- Schemat – edytor blokowy
- Język definiowania sprzętu HDL
- Formaty reprezentacji projektu w systemach EDA

2. Symulacja na poziomie przesłań międzyrejestrowych RTL (czy projekt poprawny logicznie?)

- weryfikacja logiczna modelu bez opóźnień

3. Synteza (przydział elementów logicznych modelu do standardowych elementów)

- minimalizacja modelu logicznego
- optymalizacja modelu logicznego
- przydział elementów logicznych do podstawowych elementów składowych układu cyfrowego

Etapy procesu projektowania dla PLD

4. **Umieszczanie i łączenie** (moduły ang. fitter router)
 - przydział elementów logicznych do jednostek logicznych sprzętu
 - uwzględnienie wprowadzonych ograniczeń na realizację projektu
 - realizacja połączeń
5. **Analiza czasowa** (czy spełnione są ograniczenia czasowe?):
 1. Badanie, optymalizacja i prezentacja efektywności czasowej
 2. sprawdzanie i informowanie o przekroczeniach ograniczeń czasowych
6. **Symulacja czasowa:**
 - Sprawdza poprawność czasową logiki
 - Korzysta z listy połączeń uwzględniających czas
 - Wymaga wektorów wymuszeń na wejściach układu
7. **Programowanie i badanie sprzętu**

Podstawowe działania (1)

TWORZENIE NOWEGO PROJEKTU

- File> New Project
 - katalog,
 - nazwa projektu,
 - nazwa elementu najwyższego poziomu w hierarchii projektu
 - dołączenie plików projektu (np. plików komponentów biblioteki)
 - określenie układu docelowego - EP2C35F672C6 dla DE2, EP2C70F896C6 dla DE2-70

Tworzenie nowego projektu za pomocą kreatora (katalog, nazwa, jednostka główna)

New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

D:\projekty_quartus\RS

What is the name of this project?

RS

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

RS

Use Existing Project Settings ...

< Back Next > Finish Cancel

Tworzenie nowego projektu za pomocą kreatora projektu

New Project Wizard: Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family
Family: Cyclone II
Devices: All

Target device
☐ Auto device selected by the Filter
☒ Specific device selected in 'Available devices' list

Show in 'Available device' list
Package: Any
Pin count: Any
Speed grade: Any
☒ Show advanced devices
☐ HardCopy compatible only

Available devices:

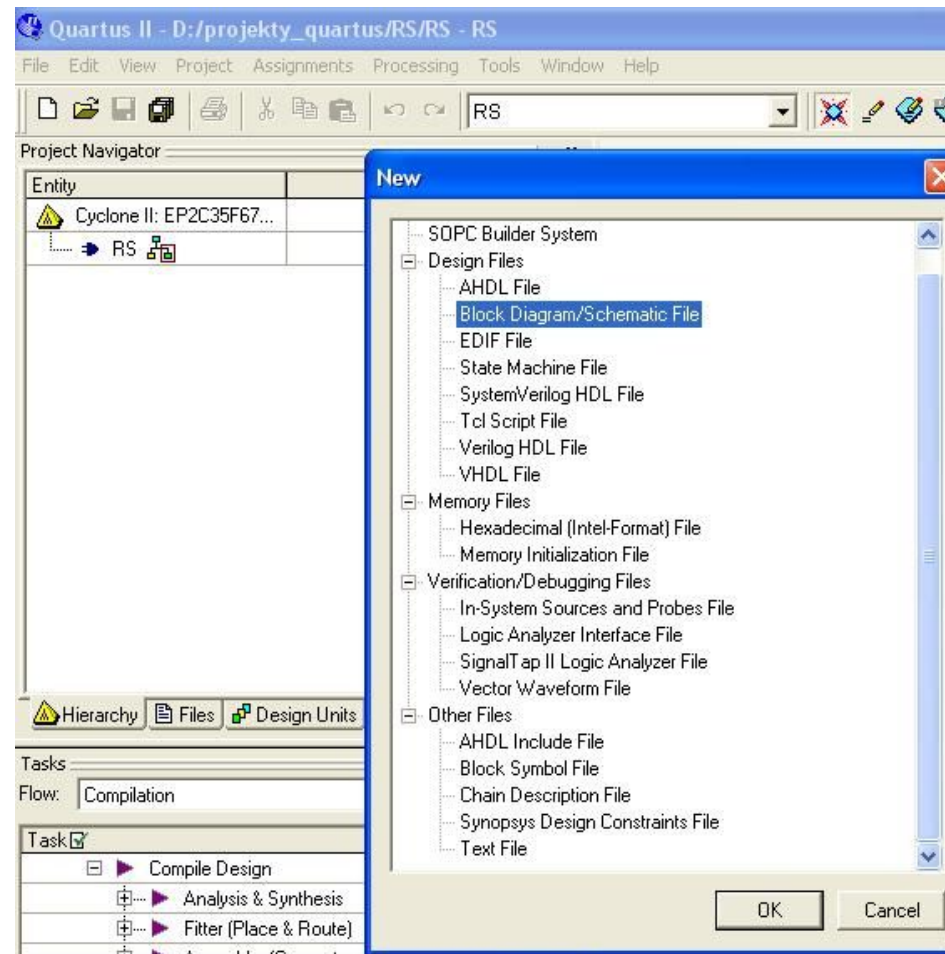
Name	Core v...	LEs	User I/...	Memor...	Embed...	PLL
EP2C20F484C8	1.2V	18752	315	239616	52	4
EP2C20F484I8	1.2V	18752	315	239616	52	4
EP2C20Q240C8	1.2V	18752	142	239616	52	4
EP2C35F484C6	1.2V	33216	322	483840	70	4
EP2C35F484C7	1.2V	33216	322	483840	70	4
EP2C35F484C8	1.2V	33216	322	483840	70	4
EP2C35F484I8	1.2V	33216	322	483840	70	4
EP2C35F672C6	1.2V	33216	475	483840	70	4
EP2C35F672C7	1.2V	33216	475	483840	70	4

Companion device
HardCopy:
☒ Limit DSP & RAM to HardCopy device resources

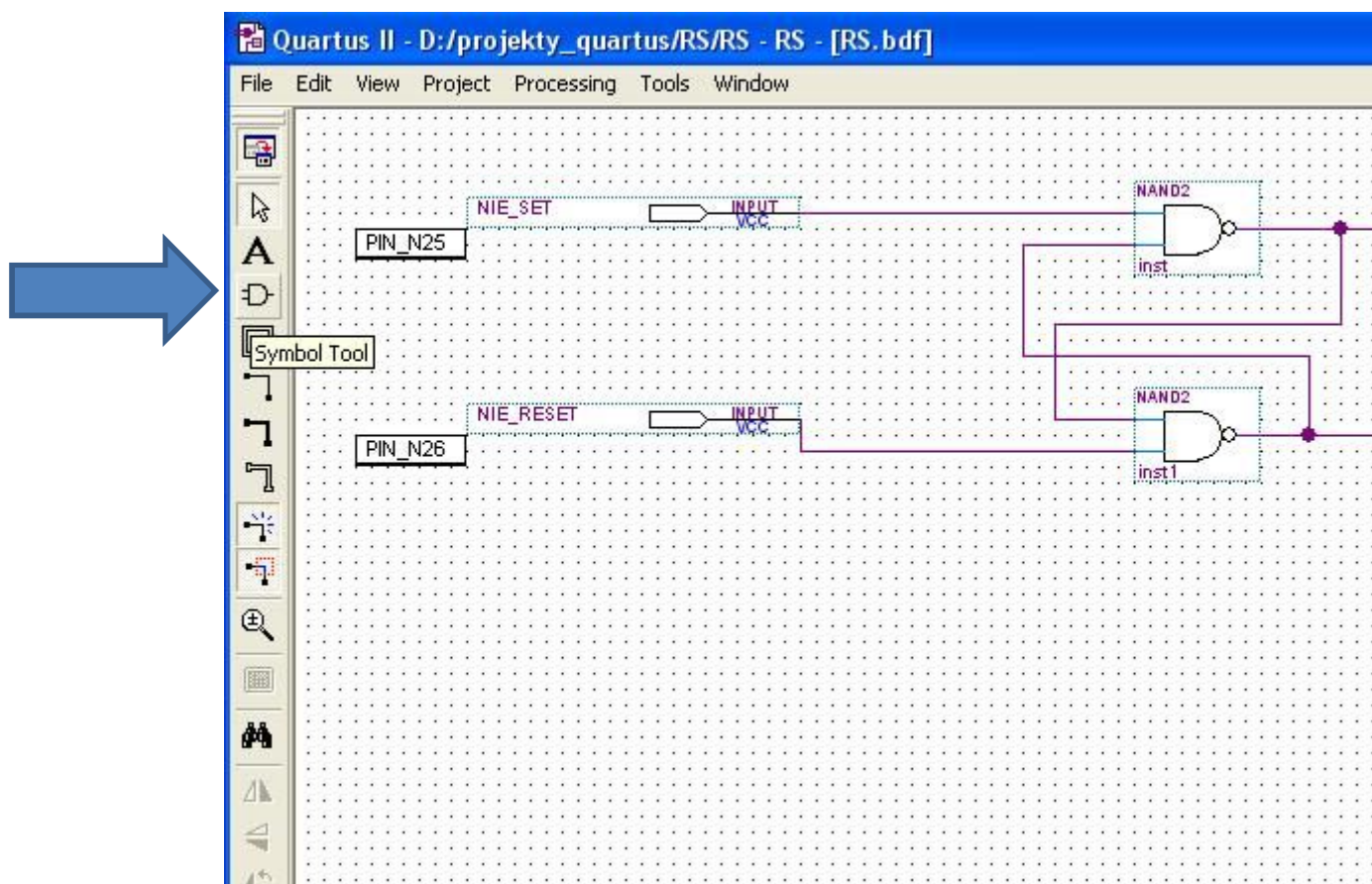
< Back Next > Finish Cancel

UWAGA: DLA DE2 EP2C35F672C6 , DLA DE2-70 EP2C70F896C6

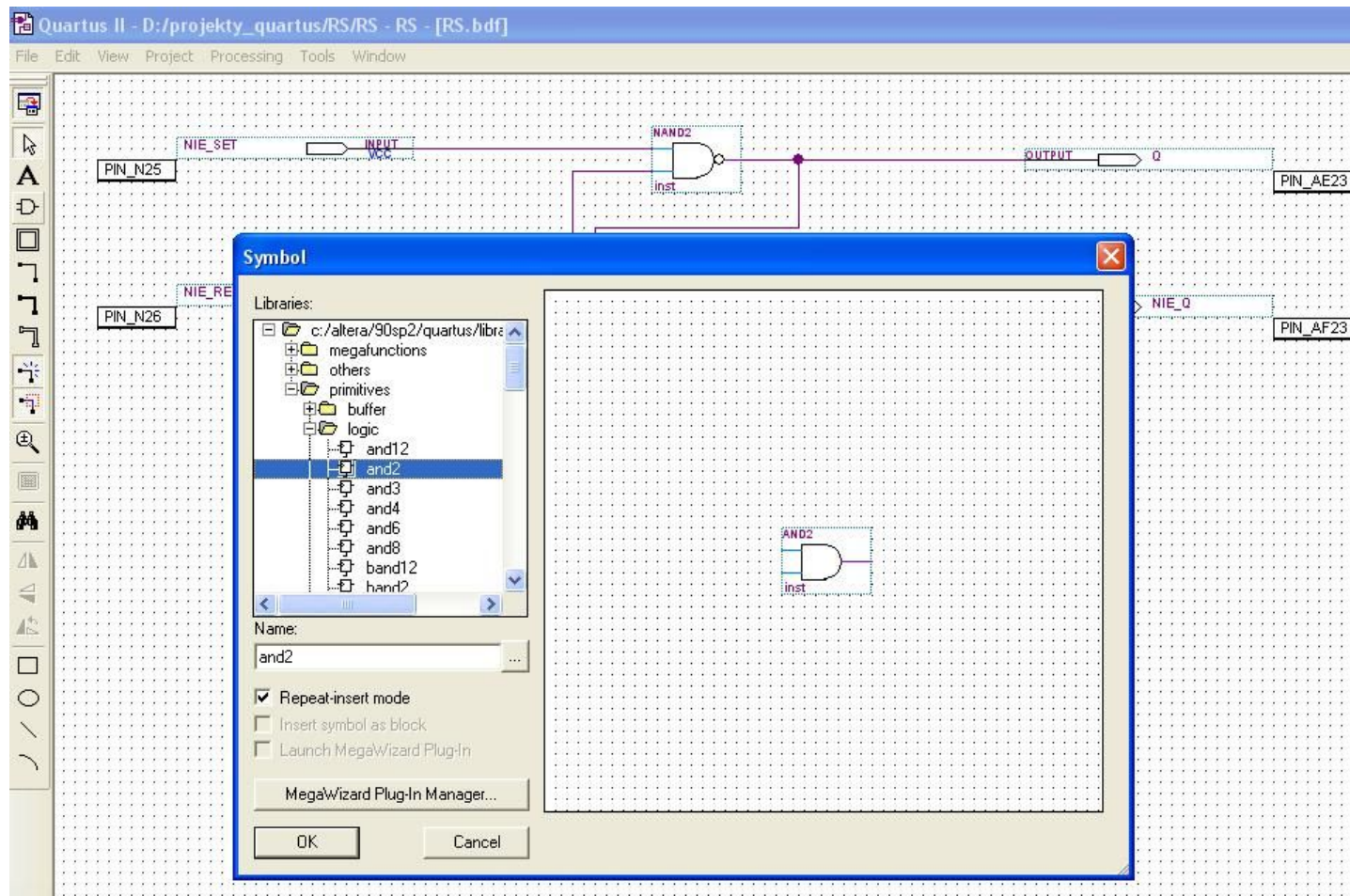
Tworzenie nowego projektu za pomocą kreatora pliki schematu blokowego



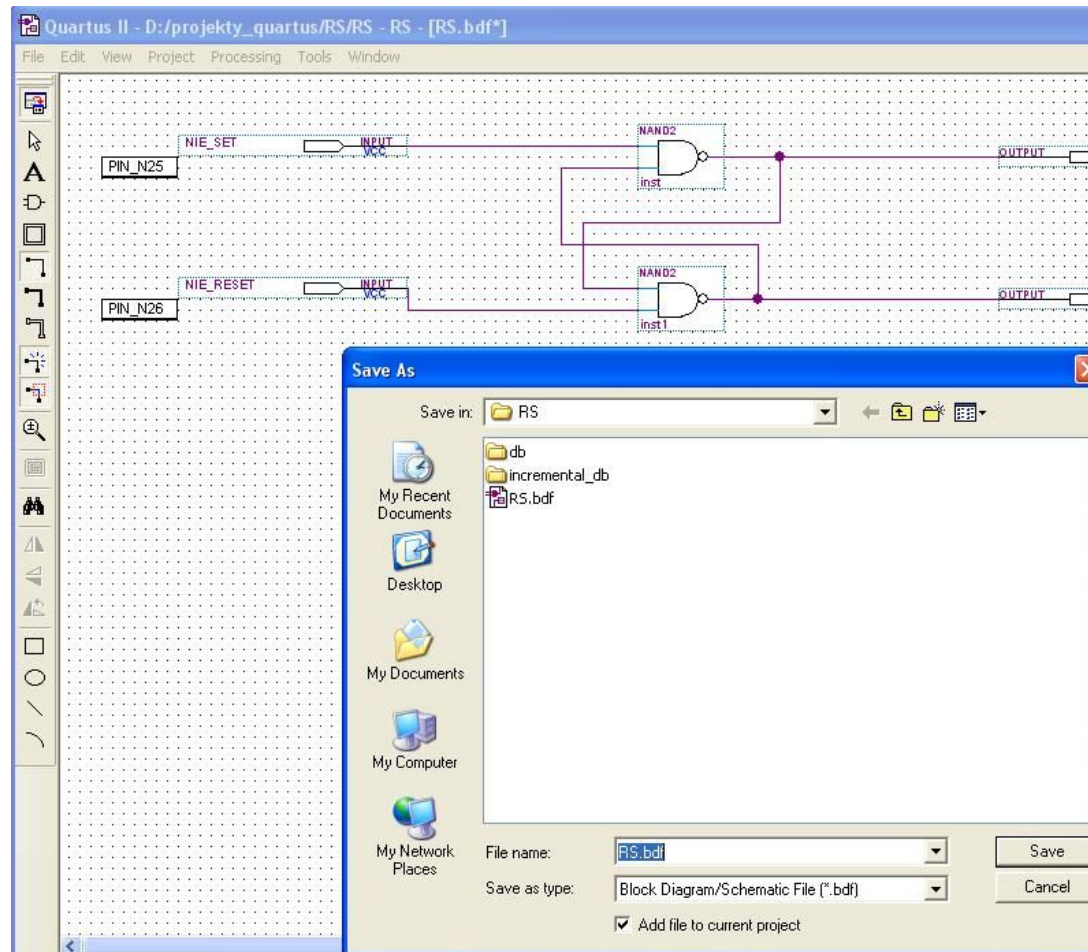
Schemat blokowy – umieszczanie podstawowych elementów logicznych 1



Schemat blokowy – umieszczanie podstawowych elementów logicznych 2



Schemat blokowy – zapis pliku w ramach projektu



Wprowadzenie powiązań sygnałów i wyprowadzeń układu FPGA

Component	DE0	DE0-Nano	DE1	DE2	DE2-70	DE2-115
SW_0	PIN_J6	PIN_M1	PIN_L22	PIN_N25	PIN_AA23	PIN_AB28
SW_1	PIN_H5	PIN_T8	PIN_L21	PIN_N26	PIN_AB26	PIN_AC28
$LEDG_0$	PIN_J1	PIN_A15	PIN_U22	PIN_AE22	PIN_W27	PIN_E21

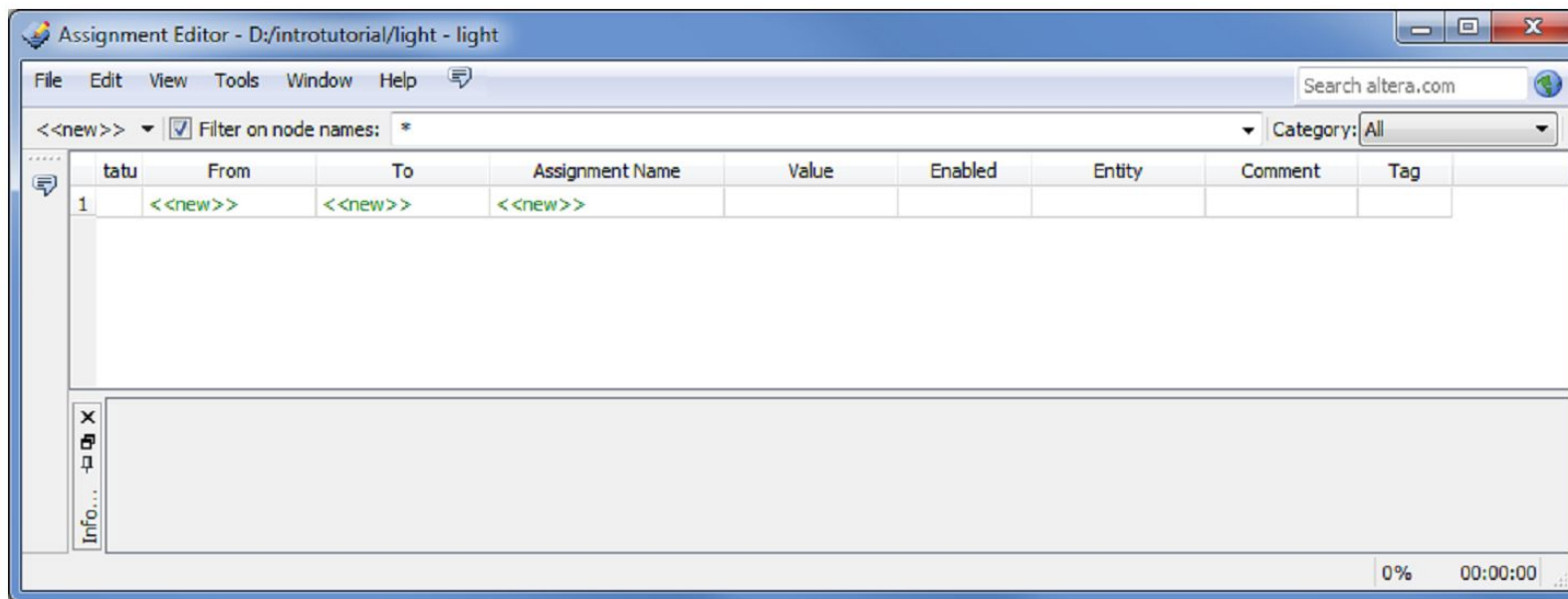
Table 2. DE-Series Pin Assignments

Przykładowe przyporządkowania sygnałów podłączonych do elementów płytki do wyprowadzeń układu FPGA (różne wersje układów). Pozostałe informacje w plikach [DE2_pin_assignments.csv](#), [DE2_70_pin_assignments.csv](#)

Wprowadzenie powiązań sygnałów układu i wyprowadzeń układu FPGA

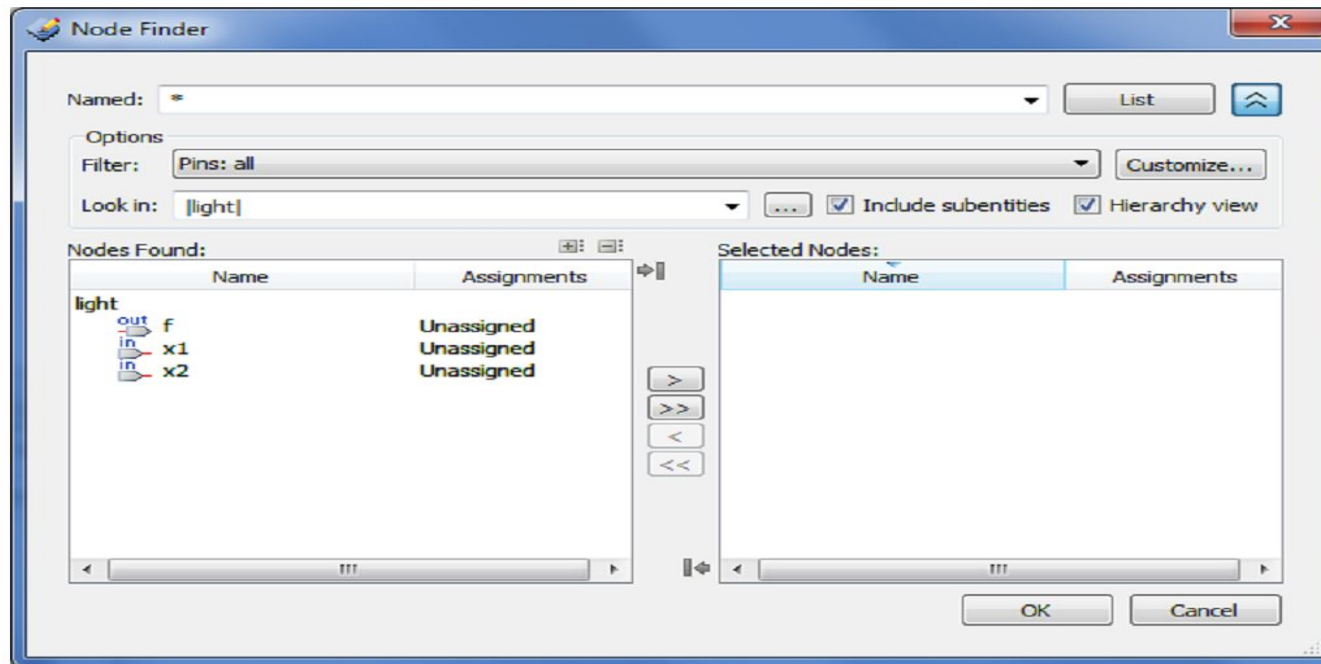
- Kroki realizowane w Assignment editor opisano na kolejnych stronach
- Wg podręcznika: Quartus II Introduction Using Schematic Designs

1. wybrać: Assignments > Assignment Editor



W rozwijanym menu Category wybrać All.

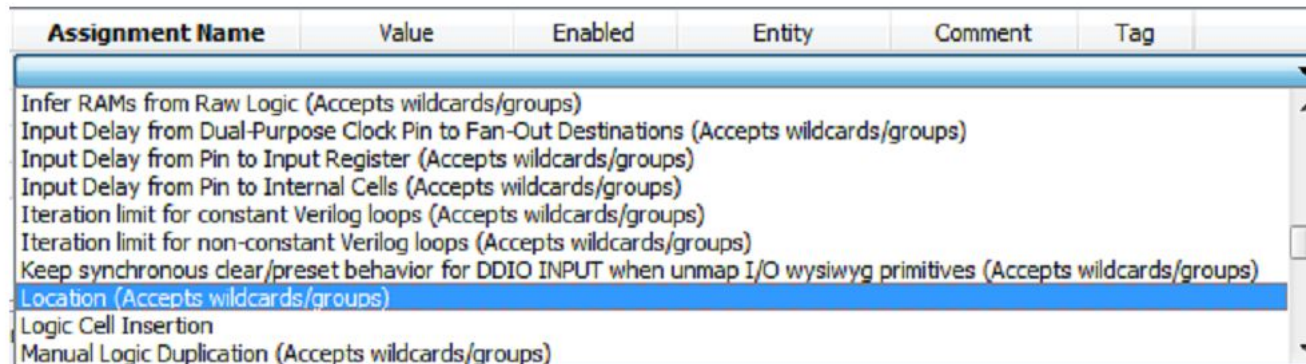
Wbrać przycisk <<new>> w lewym górnym rogu okna tworząc nowy wpis w tabeli. Wybrać (dwuklik) pole w kolumnie oznaczonej nagłówkiem To dla uzyskania przycisku Node Finder. Po wybraniu przycisku pojawi się okno jak na poniższym rysunku:



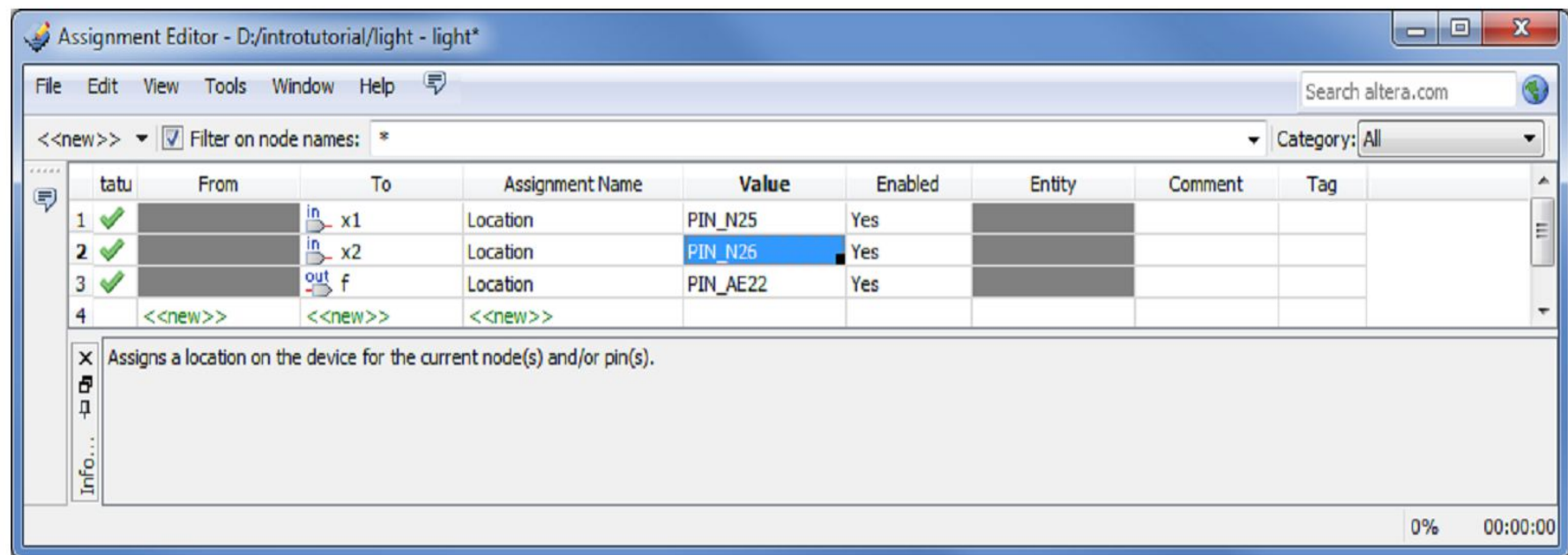
W oknie filtru wybrać: Pins: all

Pojawiające się w oknie **Nodes found** nazwy wyprowadzeń układu skopiować strzałką do okna **Selected Nodes**. Po potwierdzeniu OK wybrane sygnały pojawią się w kolumnie **To** w Assignment Editor.

Wybranie (dwuklik) w Assignment Editor pola na prawo od nazwy sygnału w kolumnie Assignment Name spowoduje pojawienie się poniższego menu w którym wpisując pierwszą literę nazwy **L** należy znaleźć i wybrać opcję Location (Accepts wildcards/groups). (należy powtórzyć dla każdego sygnału ze schematu uzyskując **Location** w kolumnie Assignment Name)



W odpowiadjącej sygnałowi kolumnie Value należy wpisać (uzyskany np. z pliku [DE2 pin assignments.csv](#) lub [DE2 70 pin assignments.csv](#)) opis wyprowadzenia układu FPGA do którego chcemy podłączyć sygnał projektowanego urządzenia.



Zamykając okno Assignment Editor powodujemy zapis przyporządkowań do pliku.

Format pliku przyporządkowań (*.QSF) sygnałów projektu do wyprowadzeń układu FPGA jest pokazany poniżej. Ten plik tekstowy można przygotować również poza programem Quartus i wczytać opcją Import Assignments.

```
set_location_assignment PIN_N26 -to b //b podłączono do SW1
set_location_assignment PIN_P25 -to c //c podłączono do SW2
set_location_assignment PIN_AE22 -to y0 //y0 podłączono do LEDG0
set_location_assignment PIN_AF22 -to y1 //y1 podłączono do LEDG1
set_location_assignment PIN_W19 -to y2 //y2 podłączono do LEDG2
set_location_assignment PIN_N25 -to a //a podłączono do SW0
```

Kompilacja projektu – analiza wejścia, synteza i optymalizacja modelu logicznego, przydział do sprzętu, asemblacja, analiza czasowa

The screenshot displays the Quartus II software interface during a compilation process. The main window is titled "Quartus II - D:/projekty_quartus/RS/RS - RS - [Compilation Report - Flow Summary]". The interface includes a menu bar, a toolbar, and several panels.

Project Navigator: Shows the project hierarchy with "Entity" (Cyclone II: EP2C70F896C6), "Logic Cells" (2 [2]), and "Dedicated Logic" (0 [0]).

Compilation Report - Flow Summary: A tree view on the left lists the compilation steps: Compilation Report, Legal Notice, Flow Summary, Flow Settings, Flow Non-Default Global Settings, Flow Elapsed Time, Flow OS Summary, Flow Log, Analysis & Synthesis, Fitter, Assembler, and Timing Analyzer.

Flow Summary: A table on the right provides a summary of the compilation results:

Flow Status	Successful - Fri Oct 01 18:29:34 2010
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	RS
Top-level Entity Name	RS
Family	Cyclone II
Device	EP2C70F896C6
Timing Models	Final
Met timing requirements	Yes
Total logic elements	2 / 68,416 (< 1 %)
Total combinational functions	2 / 68,416 (< 1 %)
Dedicated logic registers	0 / 68,416 (0 %)
Total registers	0
Total pins	4 / 622 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

Tasks: A table on the left lists the tasks and their completion status:

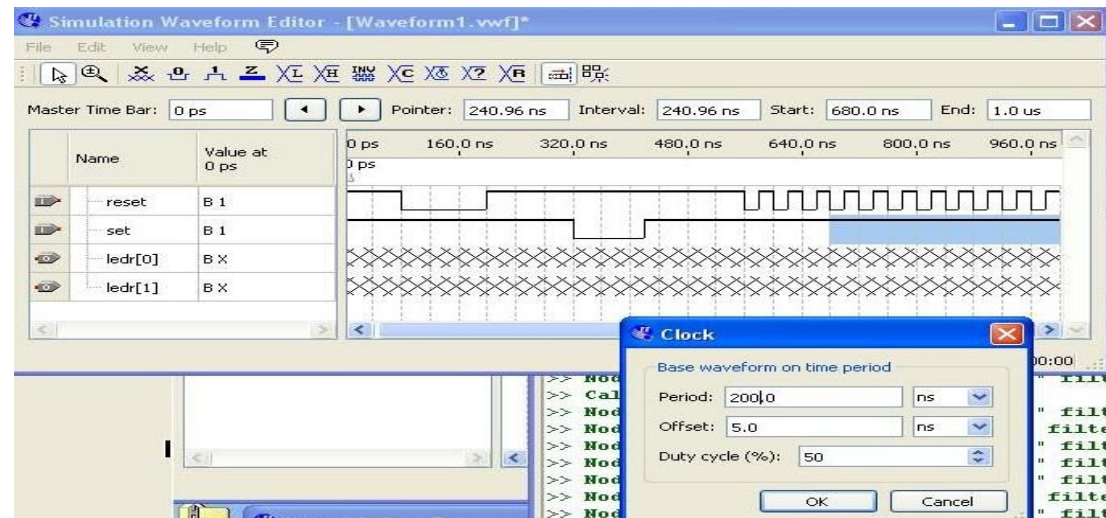
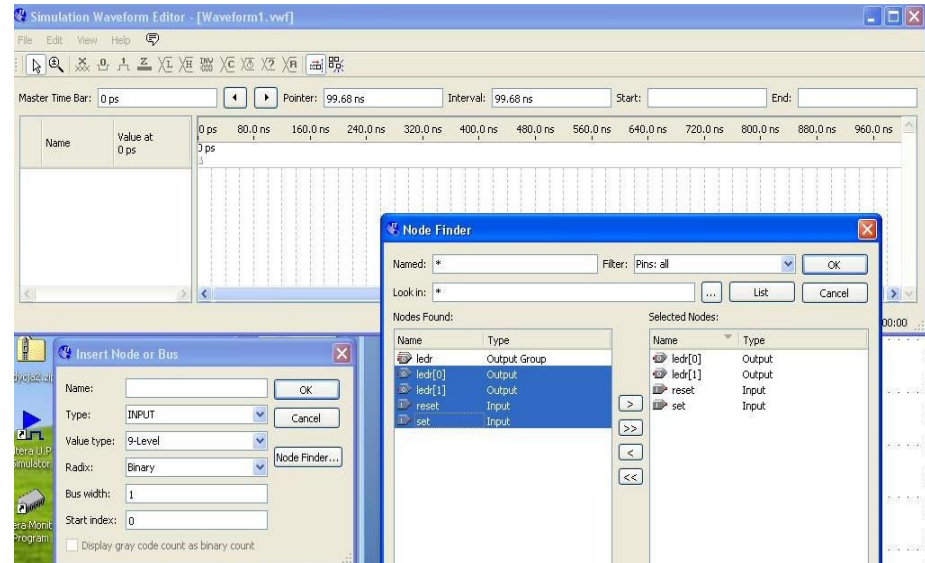
Task	Time
Compile Design	00:00
Analysis & Synthesis	00:00
Fitter (Place & Route)	00:00
Assembler (Generate programming files)	00:00
Classic Timing Analysis	00:00
EDA Netlist Writer	

Messages: A list of messages at the bottom shows the compilation progress, including information about the fitter, timing requirements, and a warning about ignored locations or region assignments.

Quartus II Dialog: A small dialog box in the center of the screen displays the message: "Full Compilation was successful (536 warnings)".

Symulacja - Altera U.P. Simulator

- File>Open Project
- File>New simulation Input File
- Edit >Insert > Insert Node or Bus
 - Node Finder



Podstawowe działania - symulator

Definiowanie wektora testowego: zbioru sygnałów, okresu symulacji i wymuszeń: Altera U.P. Simulator

- Przydatne opcje:
 - Edit> End time
 - View>Fit in window
 - Edit>Insert Node or Bus Node
 - View>Snap to grid
 - Selection Tool – ikona w kształcie strzałki
 - Waveform editing Tool – ikona ze zmieniającym się stanem magistrali
 - Edit>Value - edycja stanu wymuszenia

Symulacja - Altera U.P. Simulator

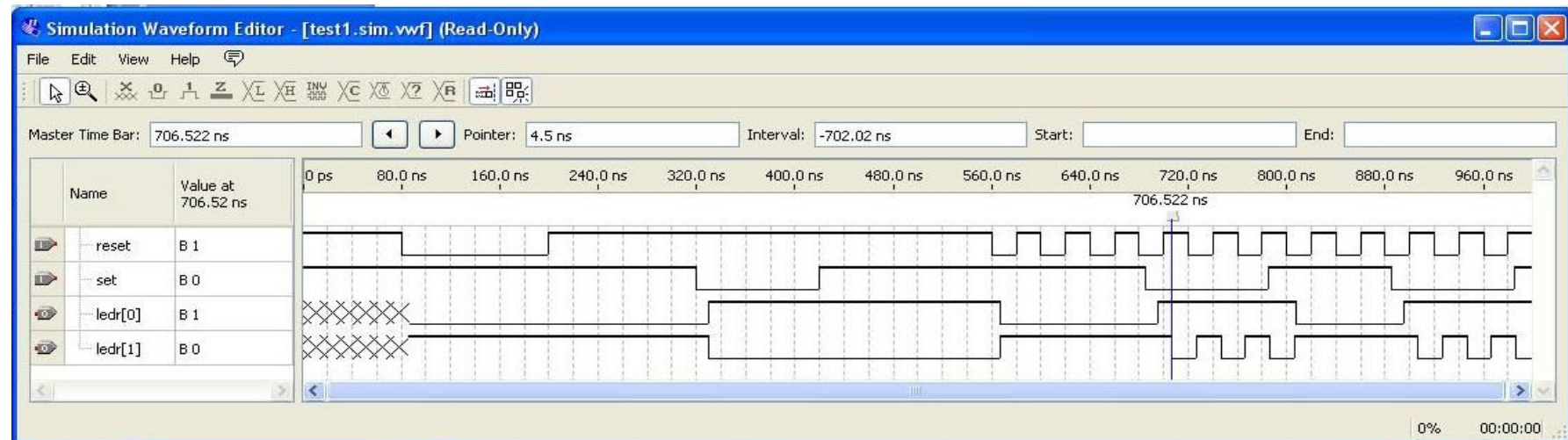
Użyteczne opcje:

- Edit > Set End Time
- Edit > Grid Size
- Edit > Snap to Grid
- Edit > Snap to Transition

Przygotowanie i kroki symulacji:

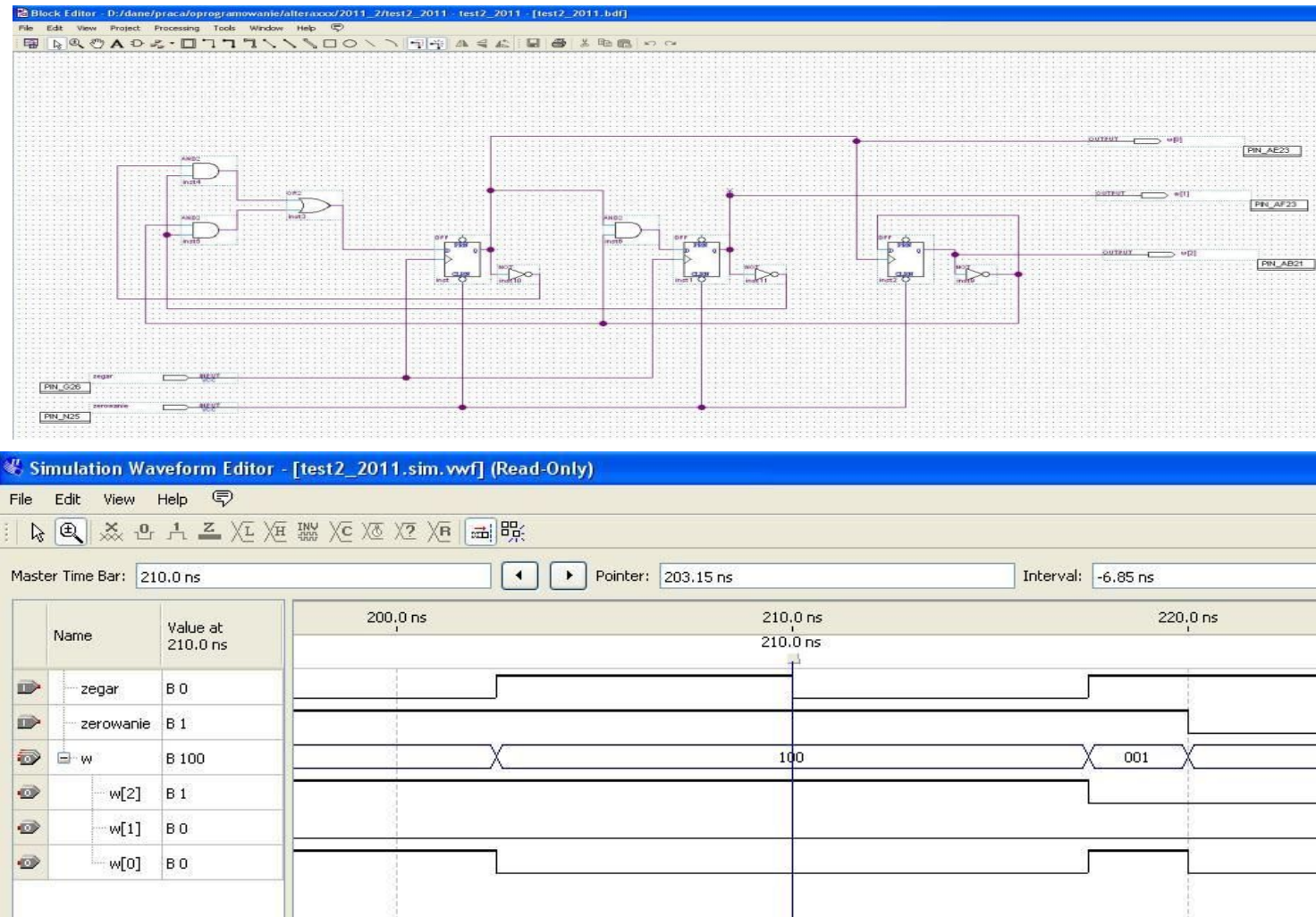
- Assign > Simulation Settings — określenie pliku wektorów testowych oraz typu symulacji
- Processing > Generate Simulation Netlist
- Processing > Start Simulation

Symulacja - Altera U.P. Simulator

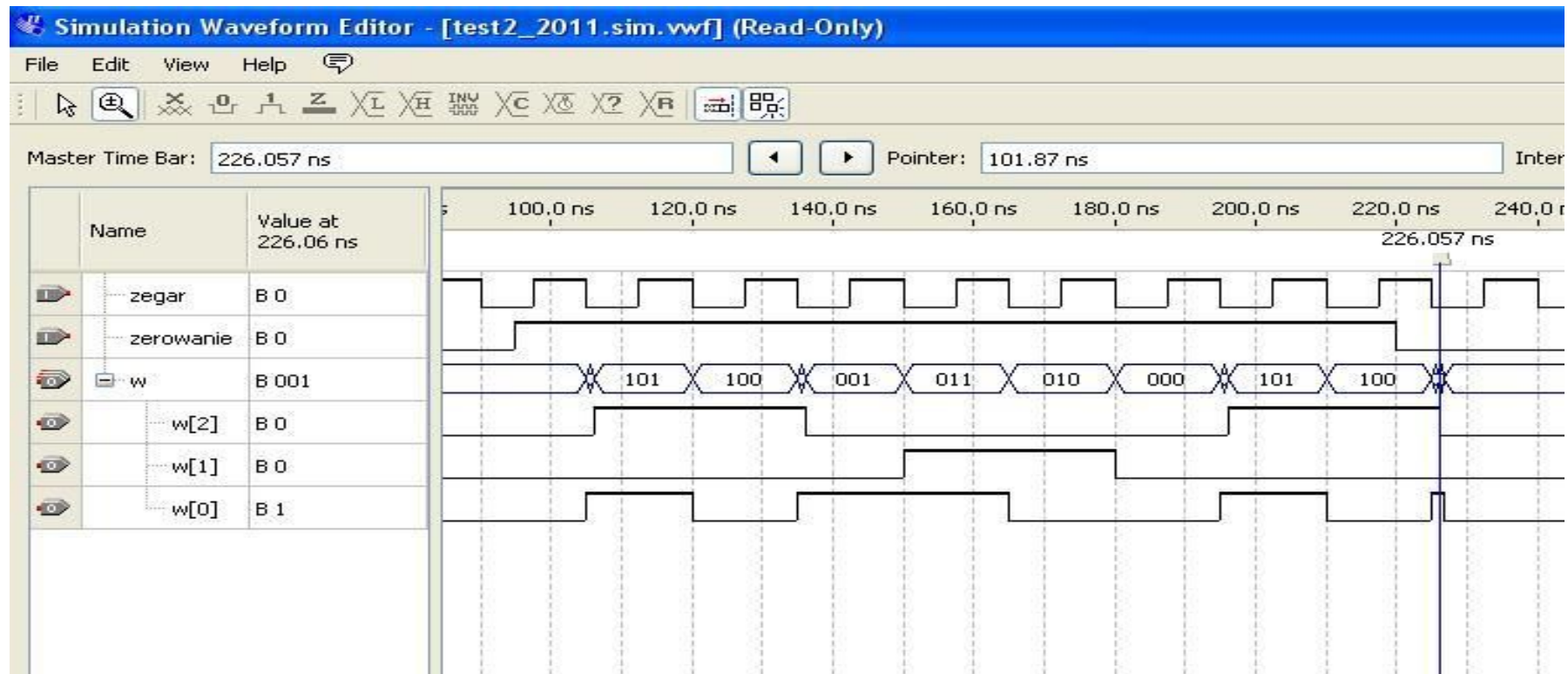


- Wyniki symulacji widoczne w Simulation Waveform Editor
- Szersze informacje na temat symulacji i edycji pliku wektorów testowych w:
- *Introduction to Simulation of VHDL Designs: Altera corporation University Program May 2011*

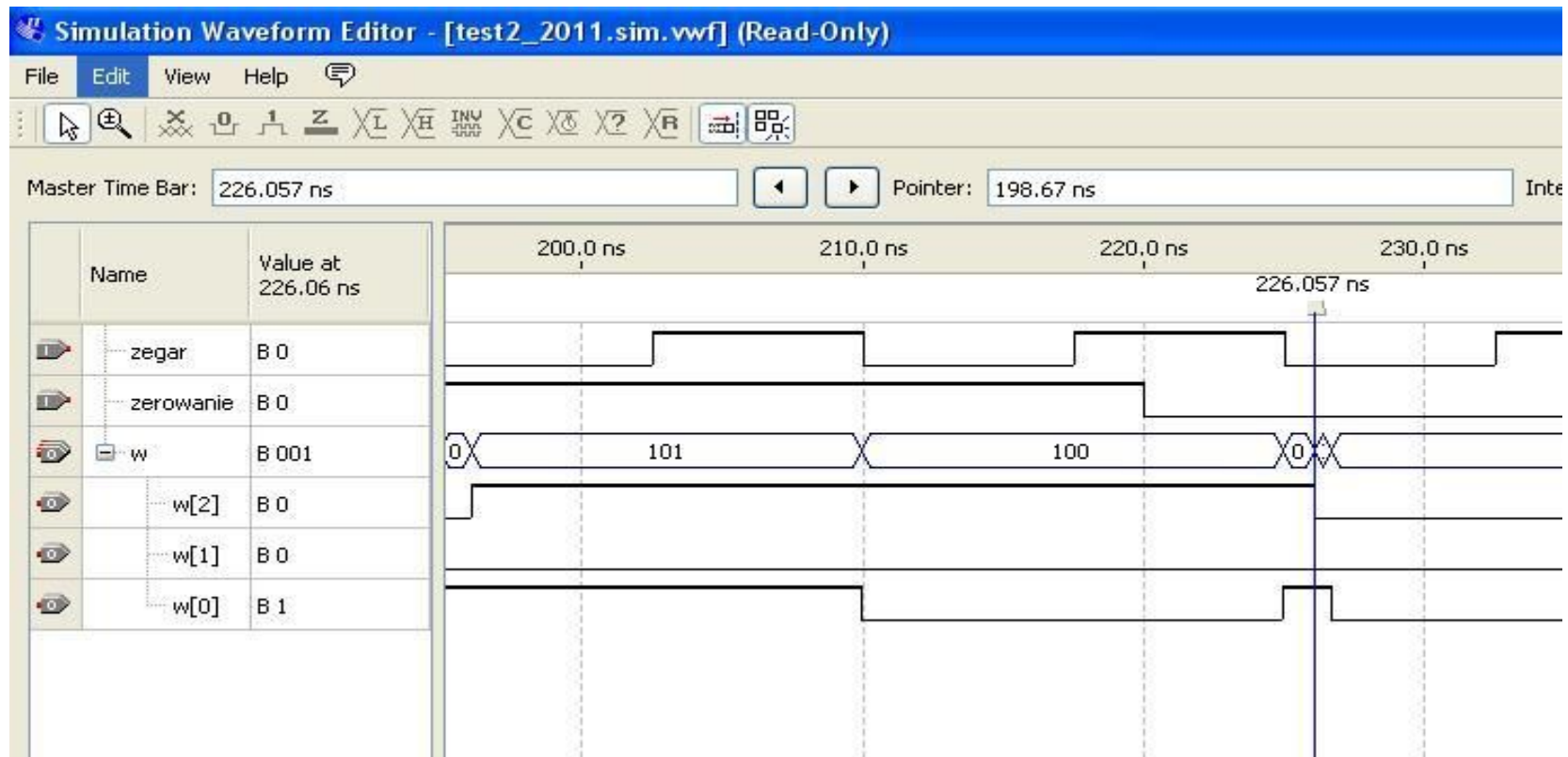
Przykłady symulacji – symulacja funkcjonalna logika działania



Przykłady symulacji – symulacja czasowa - realizacja w sprzęcie



Przykłady symulacji – symulacja czasowa (powiększenie)



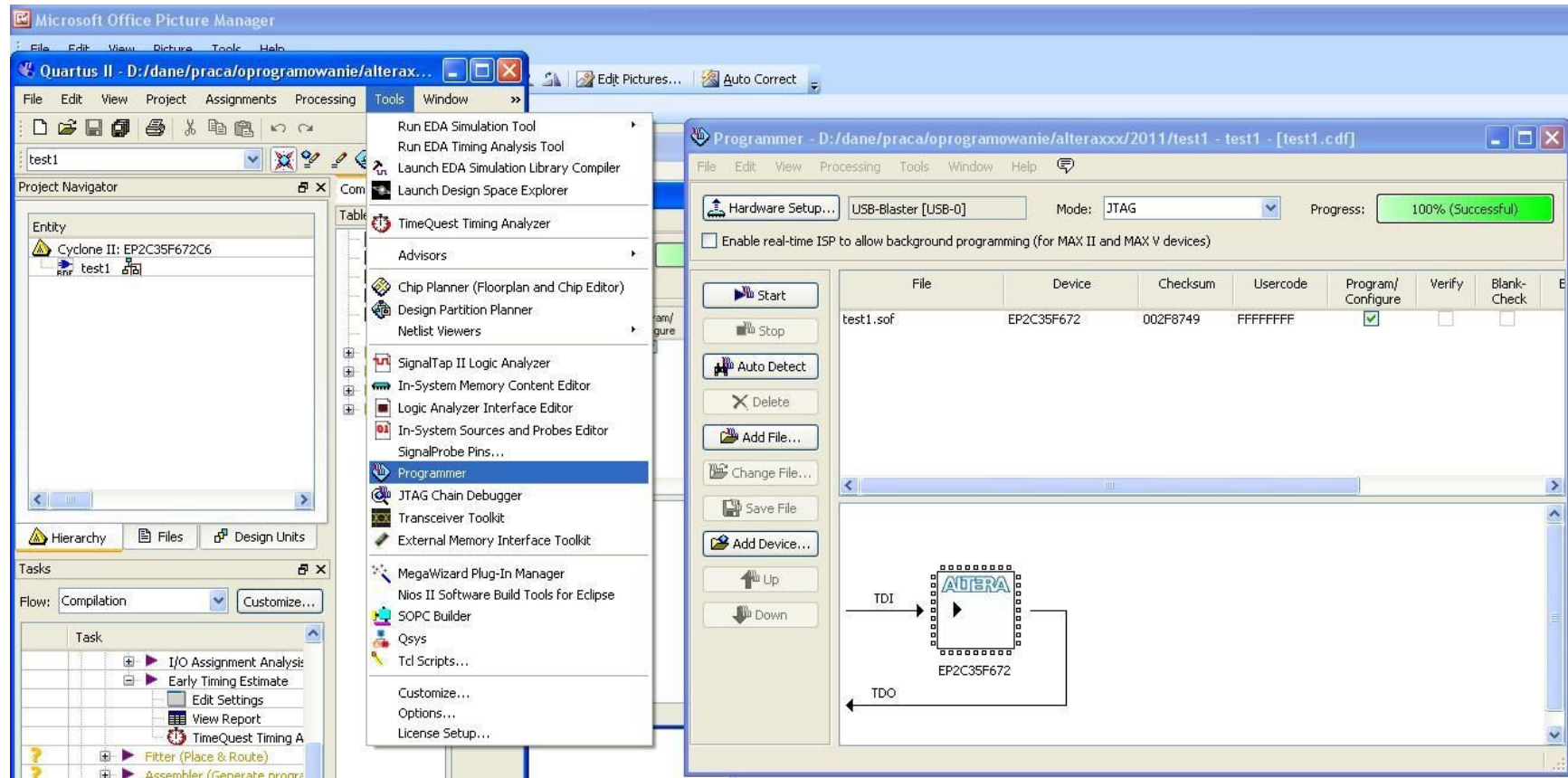
Podstawowe działania ALTERA

- **Programowanie i konfiguracja FPGA**

Tryb: JTAG – ładowanie danych konfiguracyjnych układu bezpośrednio do FPGA, pozostają do wyłączenia zasilania. Do programowania w tym trybie: przełącznik trybu programowania płyty DE2 w stan RUN

- Tools>Programmer
- Hardware Setup > USB-blaster
- Korzystamy z *.sof (SRAM object file) (Add file)
- Ustawienie przełącznika DE2 w stan RUN
- Program/Configure
- Start

Programowanie FPGA



Tools/Programmer ; Wykorzystujemy interfejs USB-Blaster

Podstawowe działania dla VHDL(1)

- **Nowy plik VHDL:**
 - File>New>VHDL
 - określenie nazwy: File>Save As > Box Add file to current project
- **Umieszczanie wzorców elementów w pliku VHDL:**
 - Edit>Insert Template>VHDL
- **Dodawanie/usuwanie plików do/z projektu:**
 - Assignments>Settings>Files
 - Project>Add/Remove Files in Project

Podstawowe działania VHDL (2)

- **Przydział sygnałów do wyprowadzeń (pinów) programowanego układu FPGA:**
 - Assignments> Import Assignments
 - Wybrać plik *DE2_pin_assignments.csv* (właściwy dla wykorzystywanej karty)
 - wymagane oznaczenia sygnałów w projekcie zgodne z standardowymi (zawartymi w pliku) nazwami elementów podłączonych do pinów (nózek) programowanego układu
- **Kompilacja:**
 - Processing>Start Compilation

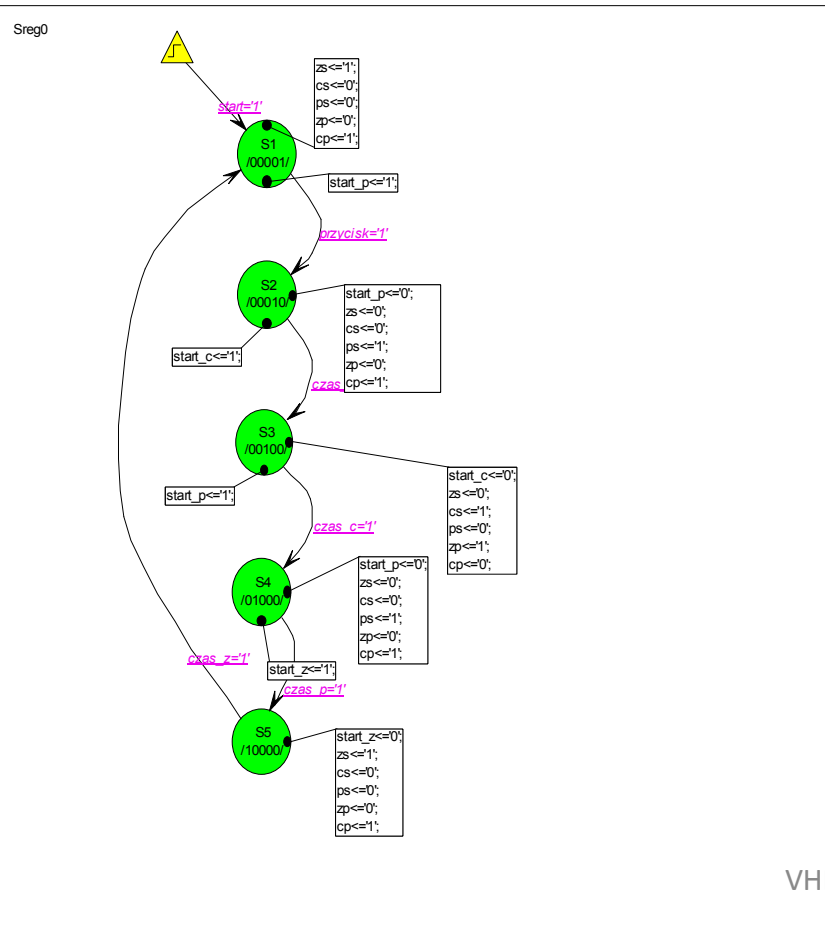
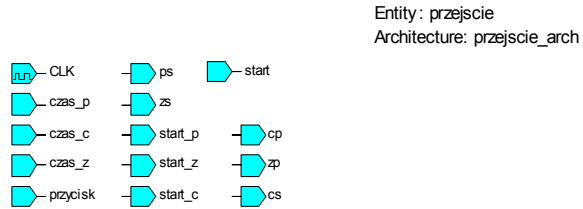
Polecane źródła informacji na temat DE2 i Quartus

- **Prezentacje na temat podstaw korzystania z Quartus**
 - *Quartus II Introduction Using VHDL Design*
 - *Introduction to Simulation of VHDL Designs: Altera corporation University Program*
May 2011

Oprogramowanie dostępne w celach edukacyjnych na
www.altera.com w sekcji Altera University Program Design
Software

- **Podręcznik użytkownika płyty DE2**
 - *DE2 Development and Education Board User Manual*

Przekształcenie grafu stanów w opis automatu w VHDL



```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

```

entity przejście is

port (

CLK: in STD_LOGIC;

czas_c: in STD_LOGIC;

...

end przejście;

architecture przejście_arch of przejście is

attribute enum_encoding: string;

type Sreg0_type is (

S1, S2, S3, S4, S5

);

attribute enum_encoding of Sreg0_type: type is

"00001 " &

-- S1

...

signal Sreg0: Sreg0_type;

begin

Sreg0_machine: process (CLK)

begin

if CLK'event and CLK = '1' then

if start='1' then

...

end process;

-- signal assignment statements for combinatorial outputs

zs_assignment:

zs <= '1' when (Sreg0 = S1) else

...

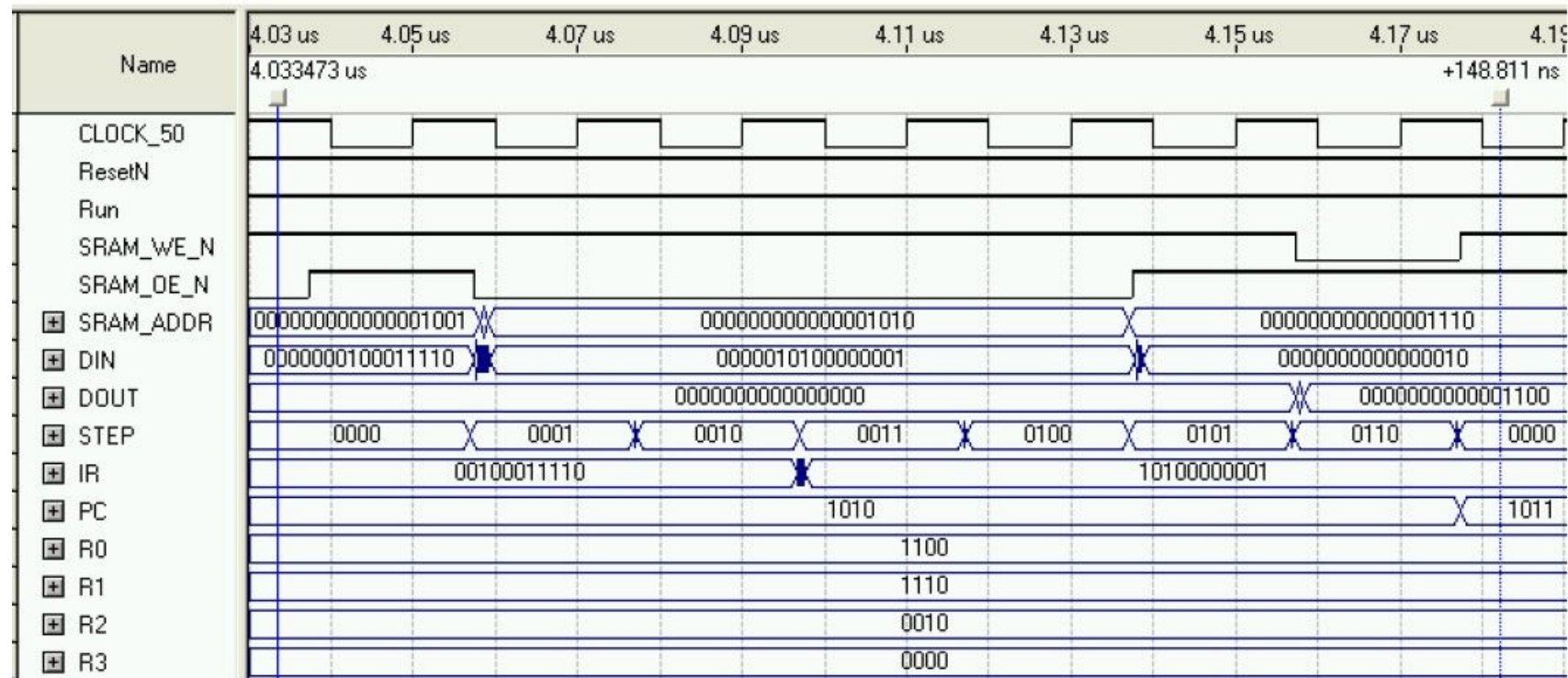
end przejście_arch;

VHDL, DE2, Quartus

Przykładowe wyniki pracy:

Symulacja pracy prostego procesora,

zapis zawartości rejestru do pamięci adresowanej zawartością rejestru



Implementacja VHDL i DE2 na laboratorium PTC, autor: student 3 roku informatyki PP 2010/2011