

Pracownia elektryczna i elektroniczna

(laboratorium techniki cyfrowej)

Projektowanie układów programowalnych FPGA w środowisku QUARTUS II (instrukcja obsługi programu)



opracowała mgr inż. Irena Hoja

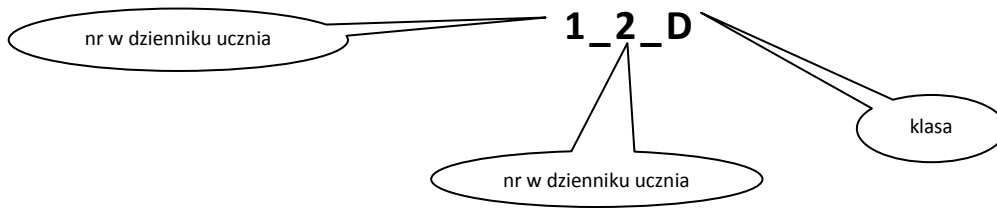
Gdańsk 2011r.

Spis treści

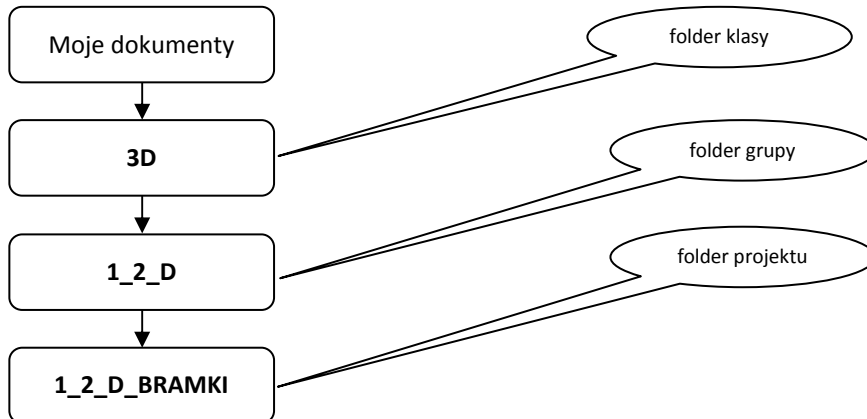
Przygotowanie do pracy w programie Quartus.....	3
Tworzenie nowego projektu	3
Specyfikacja projektu w edytorze graficznym.....	5
Symulacja realizowanego projektu.....	8
Konfiguracja układu FPGA	11
Materiały źródłowe.....	12

Przygotowanie do pracy w programie Quartus

1. Utworzyć na dysku komputera (w katalogu *Moje dokumenty*) folder dla klasy np.: **3D**
2. W folderze klasy zakładamy foldery dla poszczególnych grup laboratoryjnych, np.: **1_2_D**



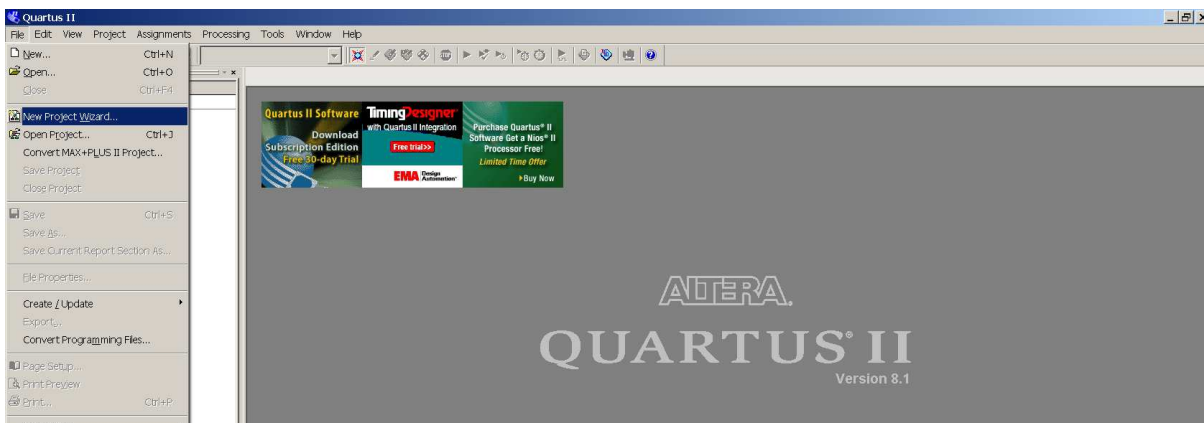
3. W folderze grupy będą zakładane katalogi realizowanych **projektów** implementowanych w układach programowalnych FPGA np.: **1_2_D_BRAMKI**



4. Należy pamiętać o tym, aby powstające kolejne pliki w programie Quartus II graficzne bądź symulacyjne **były zapisywane zawsze w katalogu projektu (1_2_D_BRAMKI)** pod taką samą nazwą:
 - **1_2_D_BRAMKI.bdf**
 - **1_2_D_BRAMKI.vwf**

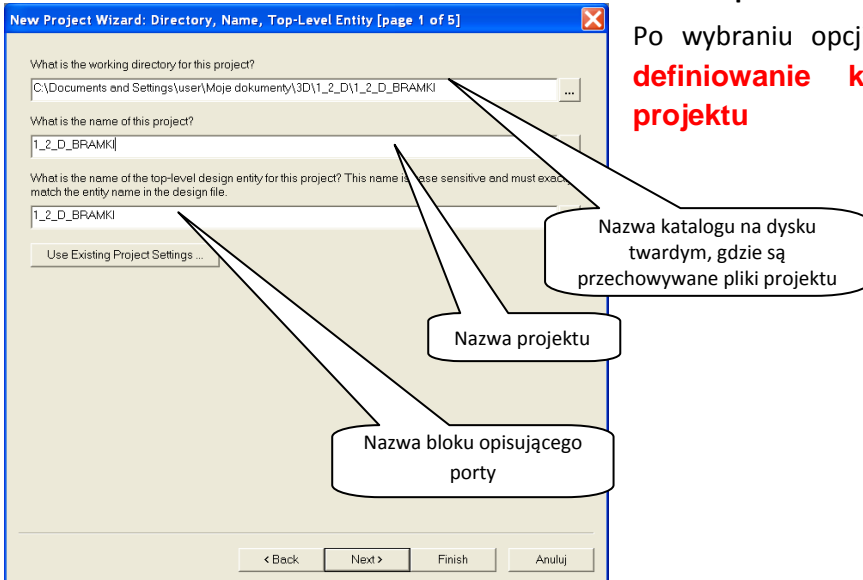
Tworzenie nowego projektu

1. Uruchomić kreator *New Project Wizard* (*File > New Project Wizard*)



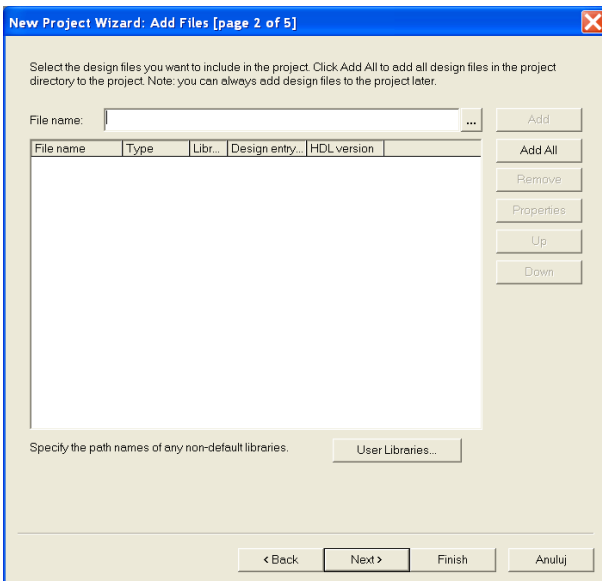
2. Wprowadzenie katalogu i nazwy projektu

Po wybraniu opcji *Next* uzyskamy okno umożliwiające **definiowanie katalogu i nazwy dla nowego projektu**



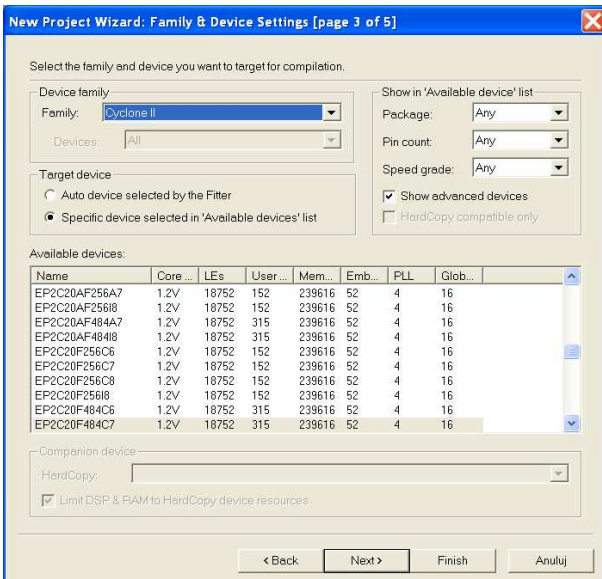
3. Dodanie do projektu istniejących plików

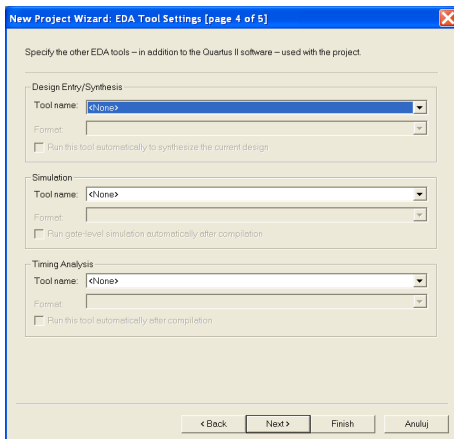
Kolejne okno dialogowe umożliwia dodanie do tworzonego projektu istniejących plików. W naszym przypadku nie mamy do dyspozycji gotowego pliku źródłowego – wybieramy *Next*



4. Wybór układu programowalnego

To okno dialogowe umożliwia wybór rodziny układów programowalnych, a następnie konkretnego układu, na którym dokonana będzie implementacja. W naszym przypadku należy wybrać rodzinę układów programowalnych FPGA **Cyclone II** i układ **EP2C20F484C7**. Konfiguracja ta równoważna jest z wybraniem struktury FPGA, która wchodzi w skład zestawu edukacyjnego **Altera_DE_1**



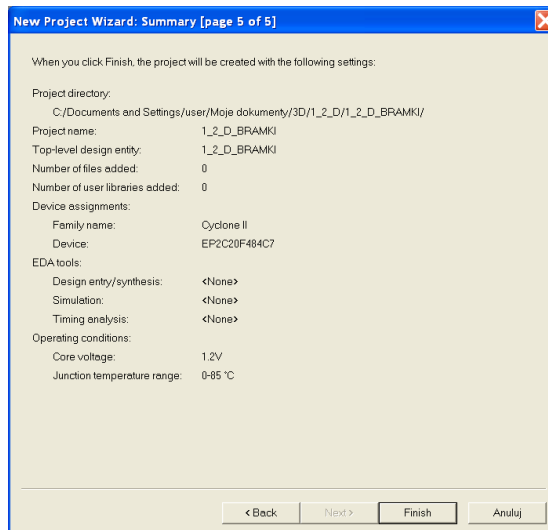


5. Specyfikacja narzędzi firm innych niż Altera

Chcąc wyłącznie korzystać z narzędzi oferowanych w ramach środowiska Quartus_II, nie zaznaczając żadnego pola wybieramy *Next*

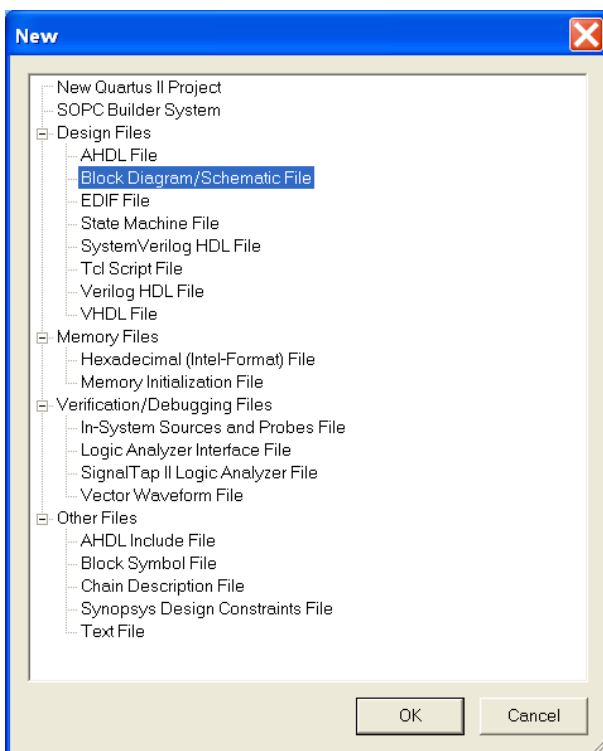
6. Podsumowanie ustawień tworzonego projektu

W ostatnim oknie dialogowym kreatora, w którym przedstawiono wybrane ustawienia projektu należy wybrać opcję *Finish*

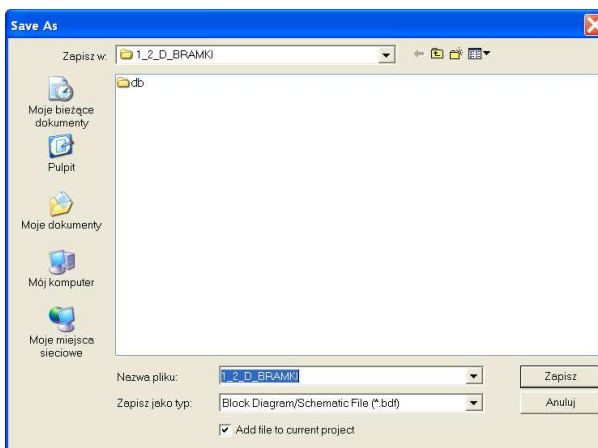


Specyfikacja projektu w edytorze graficznym

1. W celu uruchomienia edytora graficznego wybieramy: *File > New*. W oknie dialogowym należy wybrać *Block Diagram/Schematic File*. Schemat zapisujemy komendą *File > Save As*, pod nazwą *np. 1_2_D_BRAMKI.bdf* w katalogu **1_2_D_BRAMKI**

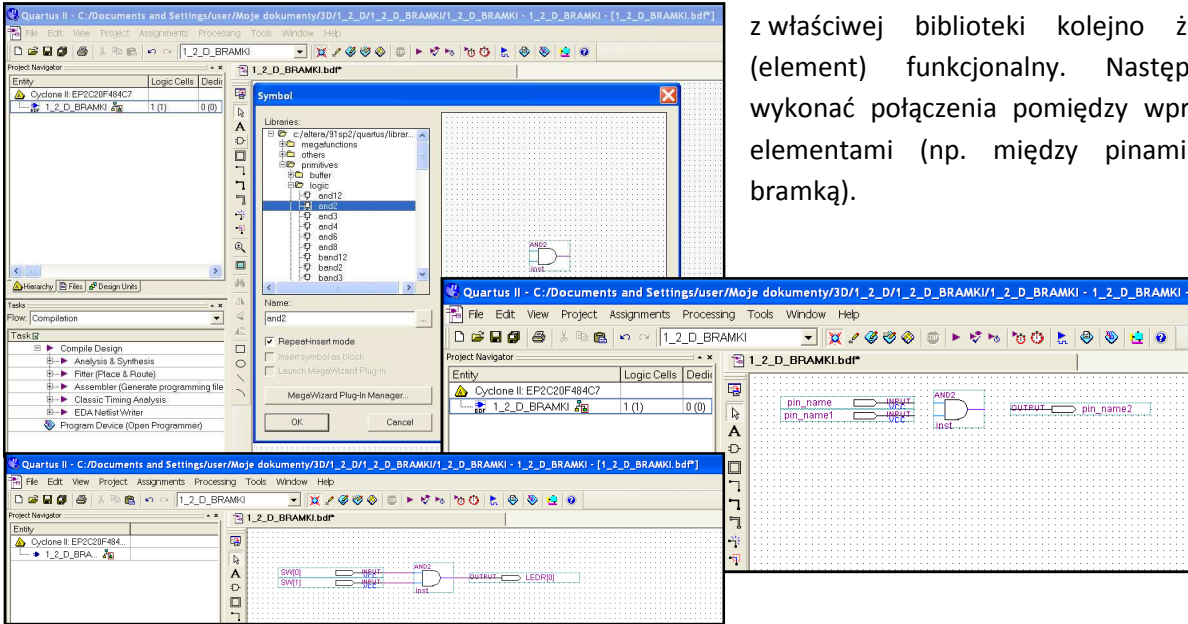


2. Przed zapisaniem pliku należy wybrać opcję *Add file to current Project*

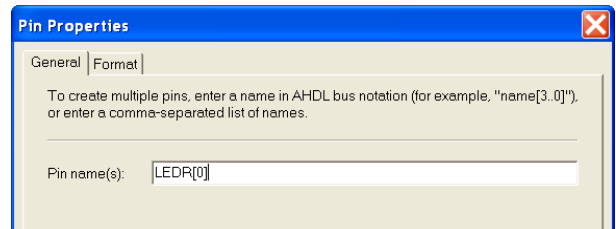
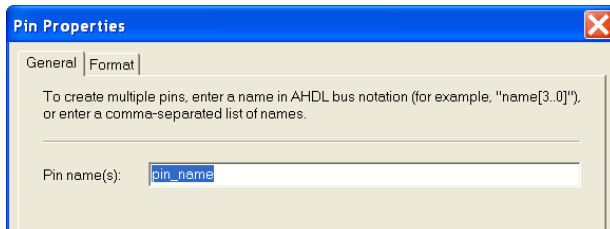


3. Edytowanie schematu


- W celu wprowadzenia symboli elementów tworzonego schematu klikamy dwukrotnie lewym przyciskiem myszy na pole edytora i wybieramy z właściwej biblioteki kolejno żądany blok (element) funkcjonalny. Następnie należy wykonać połączenia pomiędzy wprowadzonymi elementami (np. między pinami a badaną bramką).

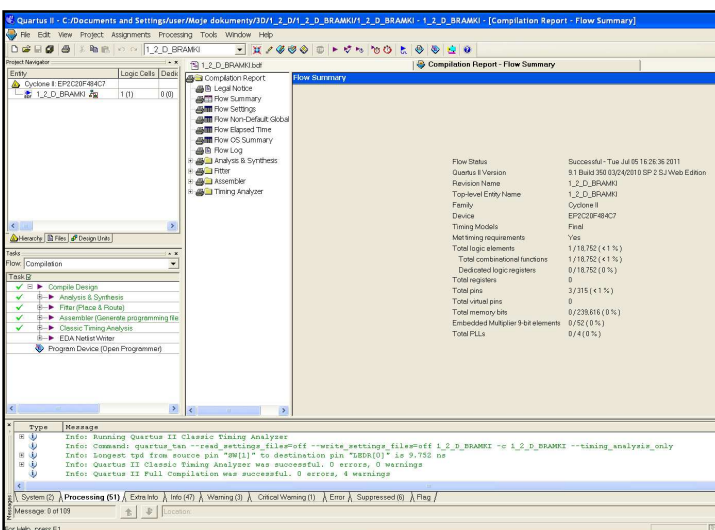


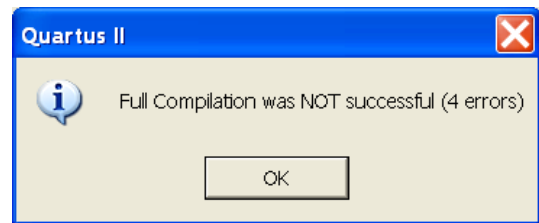
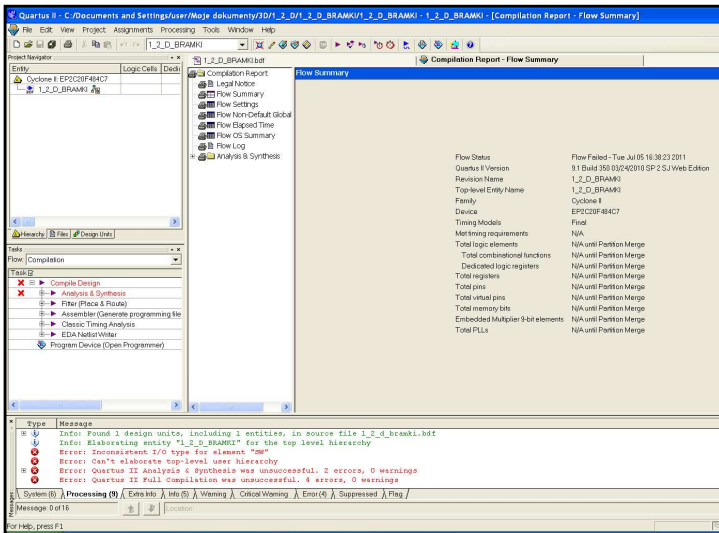
- Pinom wejściowym i wyjściowym nadajemy odpowiednie nazwy klikając na nie lewym przyciskiem myszy: np. wejściom – nazwy SW[0] i SW[1], a wyjściu LEDR[0]. Nazwy te muszą być zgodne z nazwami stosowanymi w zestawie DE_1.



4. Kompilacja projektu

Kompilujemy stworzony projekt wybierając ikonę . Po kompilacji wyświetla się nam okno dialogowe z podsumowaniem. Kompilacja może być udana lub nie, tak jak to przedstawiają okna dialogowe przedstawione poniżej.





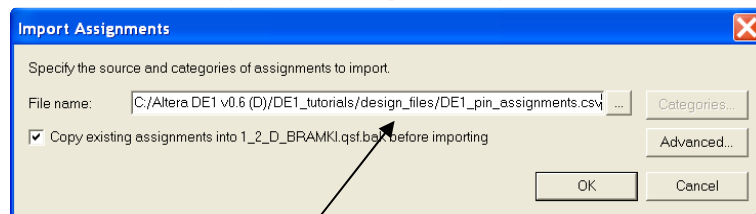
W przypadku nieudanej kompilacji w raporcie zostają wyszczególnione zaistniałe błędy w projekcie.

5. Przyprządowanie pinów

- Korzystając z informacji zawartych w dokumentacji zestawu DE_1 [1] należy przyporządkować wejściom SW[0], SW[1] oraz wyjściu LEDR[0] fizyczne piny układu EP2C20F484C7.


UWAGA! Nieodpowiednie przyporządkowanie pinów układu scalonego może spowodować po uruchomieniu programu uszkodzenie zestawu laboratoryjnego DE1. Kompilator nie zgłosi żadnego ostrzeżenia w tym przypadku. Szczególnie niebezpieczne jest zdefiniowanie pinu będącego wejściem układu scalonego jako pinu wyjściowego.

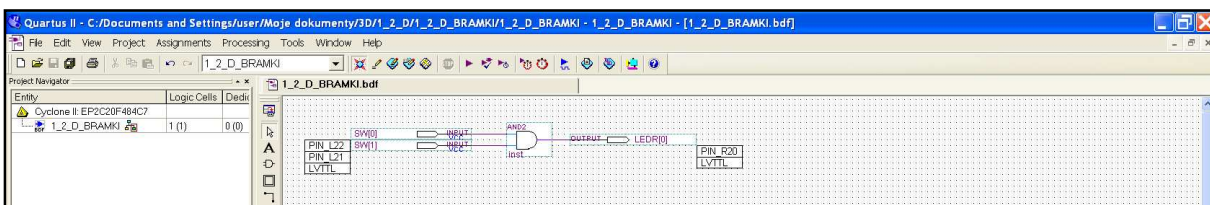
- Aby uniknąć pomyłek należy zaimportować gotowy plik *DE1_pin_assignments.csv*. W pliku tym zdefiniowano wszystkie piny układu EP2C20F484C7, przy czym nadano im nazwy zgodne z nazwami pinów zawartymi w [1 Z menu *Assignments > Import Assignments*.



Adres na komputerze w pracowni !!!

mój_komputer/dokumenty_udostepniane/Altera DE1 v0.6(D)/DE1_tutorials/design_files/DE1_pin_assignments.csv

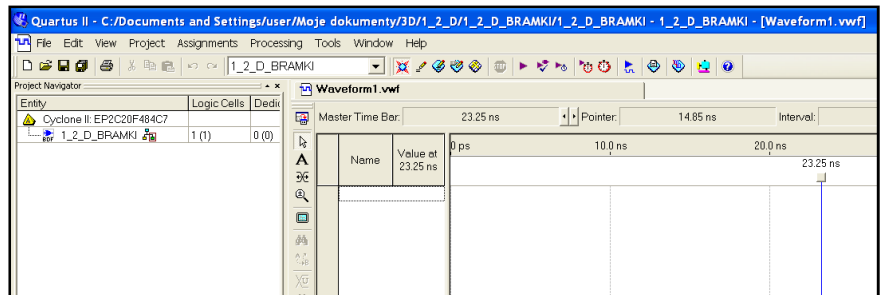
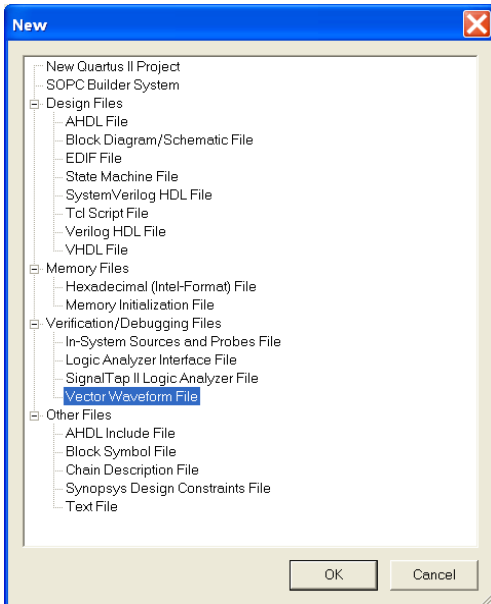
- Po przyporządkowaniu pinów ponownie kompilujemy stworzony projekt wybierając ikonę: . Schemat projektu przedstawiono poniżej.



Symulacja realizowanego projektu

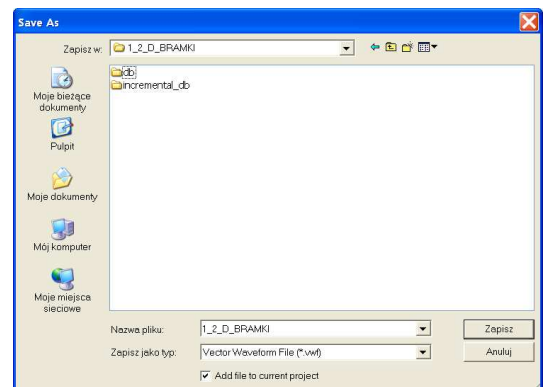
Symulacja projektowanego układu ma na celu potwierdzenie, że układ został zbudowany prawidłowo. Wykonuje się ją przed zaimplementowaniem projektu w układzie programowalnym. Proces weryfikacji projektu należy rozpocząć od utworzenia pliku zawierającego testowe przebiegi czasowe wszystkich wejść oraz odpowiadające im przebiegi czasowe wszystkich (lub wybranych do analizy) wyjść projektowanego układu, którym należy nadać wartość nieokreśloną. Wartości wyjść zostaną określone w trakcie symulacji. Aby utworzyć nowy plik zawierający przebiegi czasowe:

1. Otwieramy okno edytora symulacji *Waveform Editor* wybierając z menu *File > New*, a następnie w oknie dialogowym *Vector Waveform File*.



2. Zapisywanie pliku *Vector Waveform File*

Wybieramy z menu *File > Save As* i zapisujemy tworzone wektory testowe (*test vectors*) pod nazwą *1_2_D_BRAMKI.vwf* w katalogu projektu **1_2_D_BRAMKI !!!!**



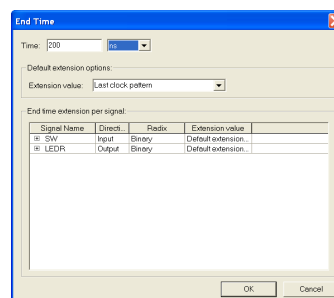
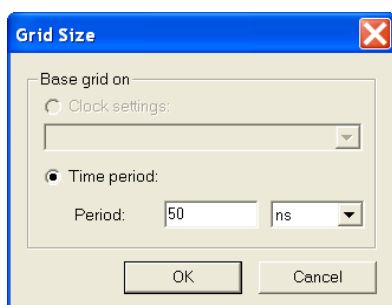
3. Ustawienie czasu symulacji i rozdzielczości czasowej

Przed przystąpieniem do nadawania wartości sygnałom wejściowym należy ustalić czas trwania symulacji oraz rozdzielczość czasową, czyli najkrótszy czas trwania zadanego stanu logicznego. Należy przy tym uwzględnić fakt, że czas całej symulacji można uzależnić od ilości wejść w projektowanym układzie (np. tak aby sprawdzić działanie wszystkich kombinacji wejść) lub od czasu potrzebnego do zamknięcia cyklu pracy układu (np. pełnego cyklu pracy licznika). Przy ustawianiu rozdzielczości czasowej należy pamiętać, że czas propagacji sygnałów może wynosić nawet 30 ns, czyli krótszy sygnał może nie zostać poprawnie zinterpretowany lub skutek jego zmiany może się przesunąć poza następną zmianę stanu.

Ustawienie rozdzielczości czasowej następuje po wybraniu polecenia *Edit → Grid Size...*,

natomiast **czas całej symulacji** ustawia się poleceniem *Edit → End Time...*

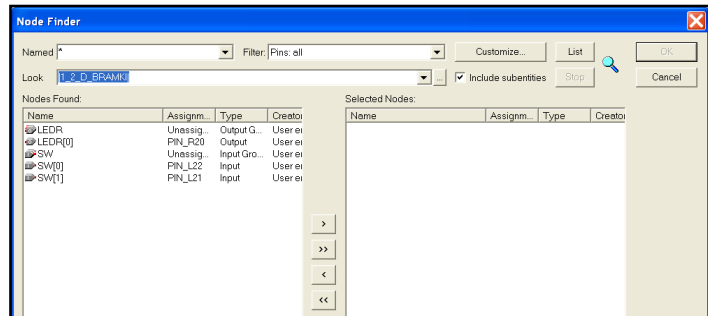
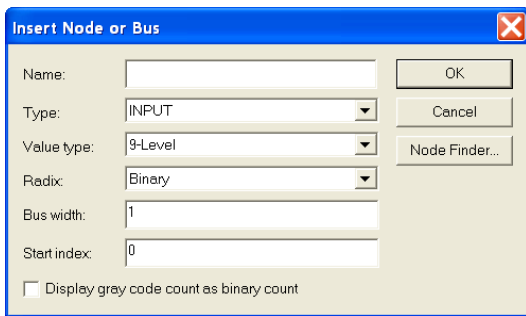
W podanym przykładzie są dwa sygnały wejściowe, zatem rozdzielczość *Grid Size* została ustawiona na **50 ns**, a czas całej symulacji *End Time* na **200 ns**.



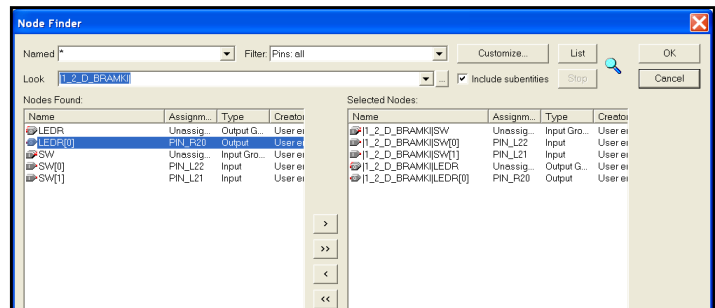
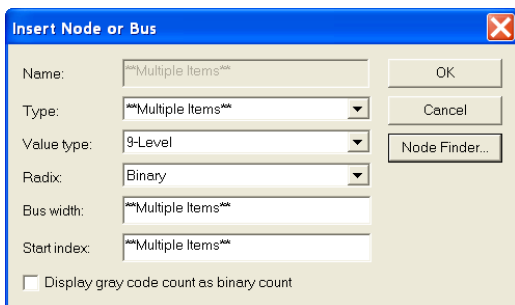
4. Chcąc obserwować przebieg symulacji dla ustawionego okresu symulacji wybieramy z menu **View > Fit In Window**

5. Wprowadzanie portów we/wy


Klikamy dwukrotnie w kolumnę *Name* (lewa strona edytora) otwierając okno *Insert Node or Bus*

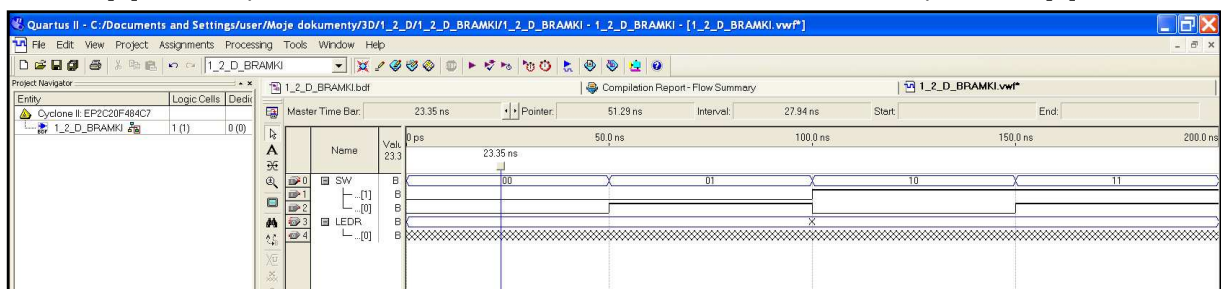


Następnie wybieramy *Node Finder* i w zakładce *Filter* wybieramy *Pins: all*. Klikamy na przycisk *List* i wybieramy z listy porty wejściowe SW[0] i SW[1] oraz port wyjściowy LEDR[0] klikając w przycisk **>**. Magistrale LEDR i SW możemy również przenieść na prawą stronę. Wówczas obserwowane sygnały będą jednocześnie opisane w wybranym kodzie.




6. Ustawianie stanów portów wejściowych

Stany portów wejściowych edytujemy zaznaczając myszką wymagany przedział czasu, a następnie klikamy prawym przyciskiem myszy i z menu *Value* wybieramy żądany stan / typ sygnału. Chcąc np. ustawić wartość jedynki logicznej należy wybrać *Value > Forcing High (1)* lub wybrać ikonę  na pasku narzędziowym edytora przebiegów czasowych. Należy np. ustawić wartość logiczną „1” dla przedziału czasu od 100 ns do 200 ns dla portu SW[1] oraz dla przedziałów od 50 ns do 100 ns i od 150 ns do 200 ns dla portu SW[0].



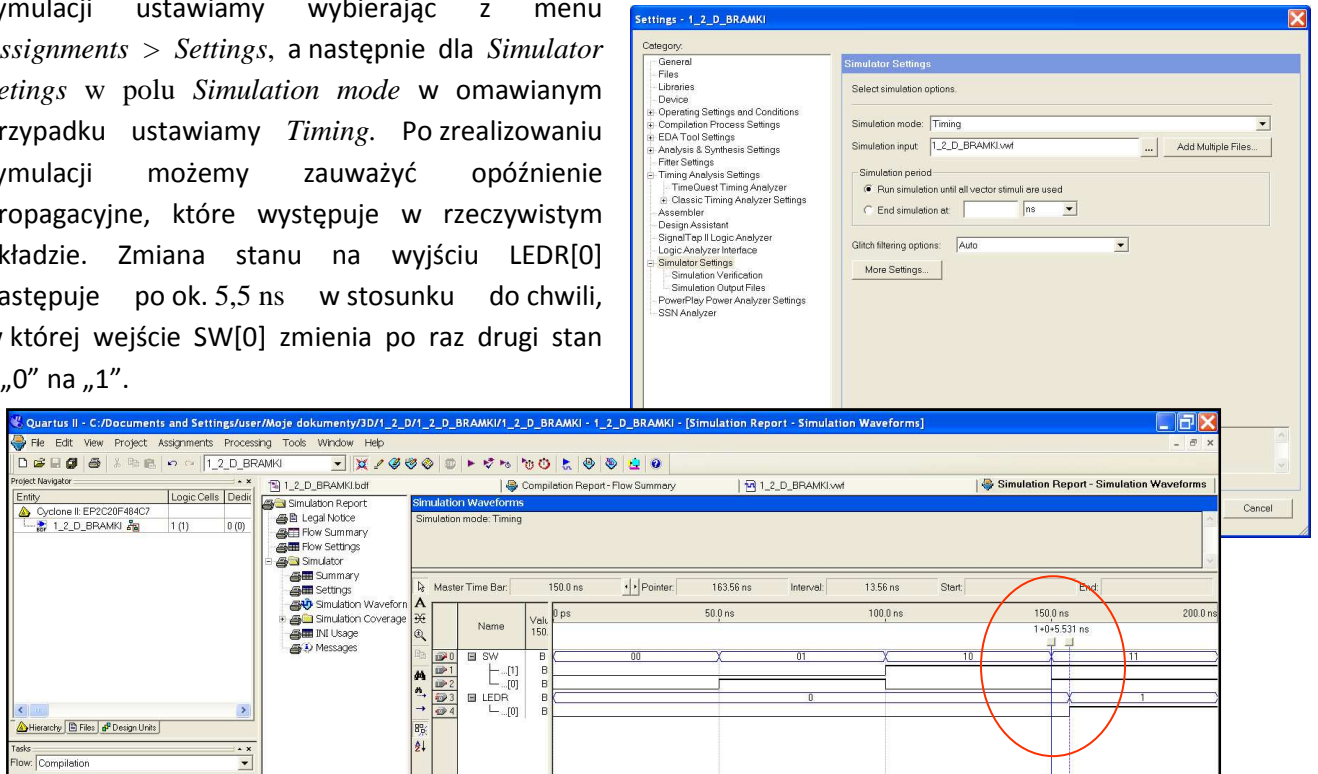
7. Symulacja

Aby wykonać symulację trzeba:

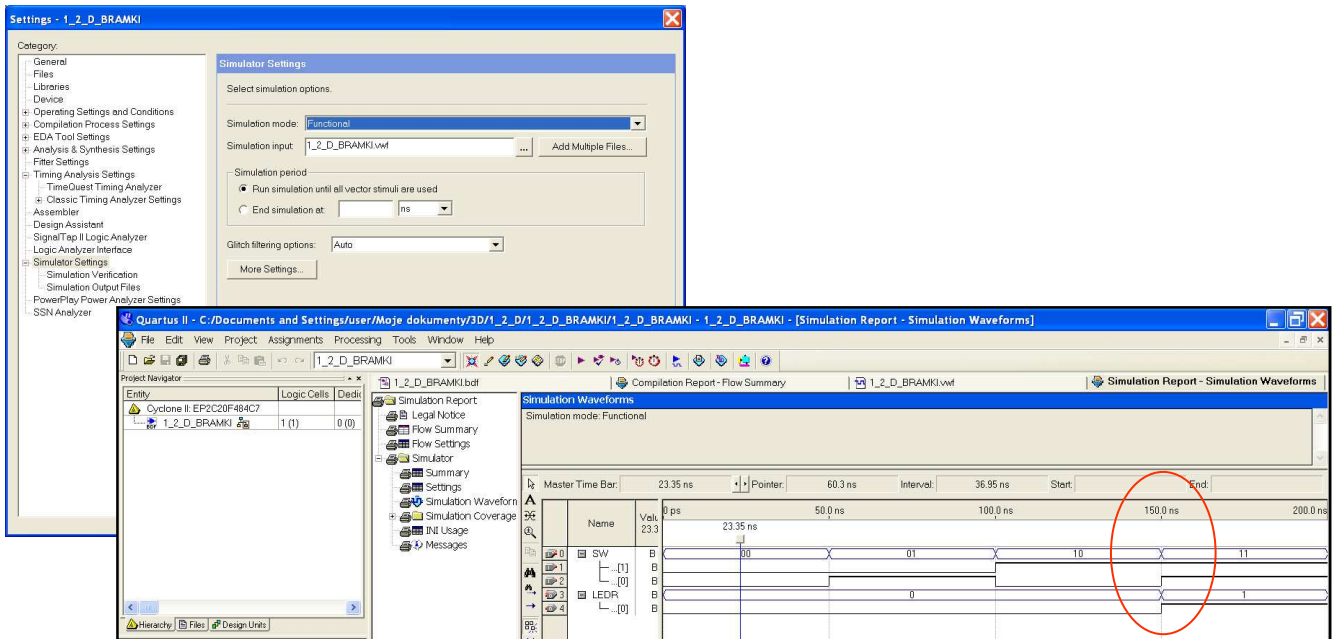
- wygenerować *Functional Simulation Netlist* wybierając z menu *Processing > Generate Functional Simulation Netlist*
- następnie wybierać z menu *Processing > Start Simulation* lub kliknąć na ikonę .

W środowisku Quartus możemy wykonać dwa typy symulacji:

- **czasową – *timing simulation***, która uwzględni opóźnienia propagacyjne występujące w układzie. Typ symulacji ustawiamy wybierając z menu *Assignments > Settings*, a następnie dla *Simulator Settings* w polu *Simulation mode* w omawianym przypadku ustawiamy *Timing*. Po zrealizowaniu symulacji możemy zauważyć opóźnienie propagacyjne, które występuje w rzeczywistym układzie. Zmiana stanu na wyjściu LEDR[0] następuje po ok. 5,5 ns w stosunku do chwili, w której wejście SW[0] zmienia po raz drugi stan z „0” na „1”.



- **funkcjonalną – *functional simulation***, która nie uwzględni opóźnień propagacyjnych. W tym przypadku w polu *Simulation mode* ustawiamy *Functional*



Konfiguracja układu FPGA


Aby umożliwić konfigurację układu FPGA znajdującego się na płycie **DE_1** należy:

- zestaw **DE_1** podłączyć do komputera poprzez złącze USB,
- zainstalować sterownik *USB-Blaster*. Podczas pierwszego podłączenia zasilanego zestawu do komputera system Windows zapyta o lokalizację sterownika, należy wówczas wybrać katalog *Quartus/ drivers/ usb_blaster*. Po zainstalowaniu sterownika można przystąpić do programowania układu FPGA.

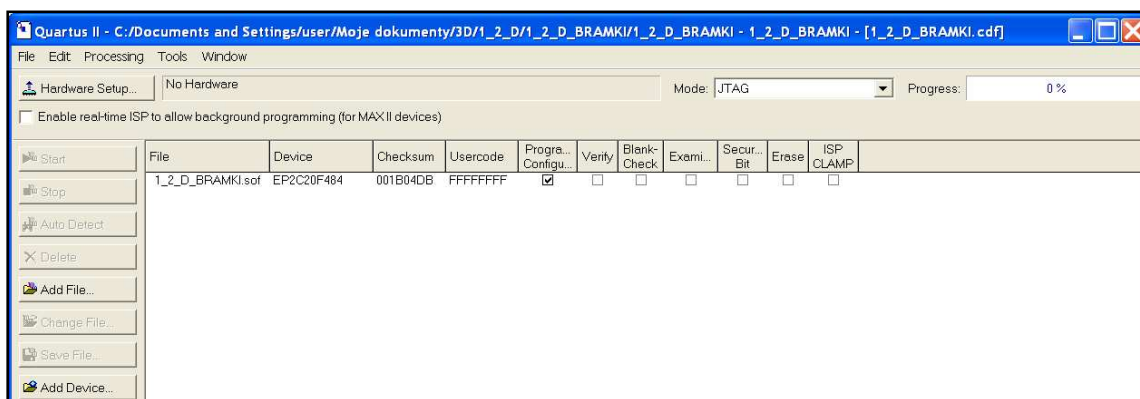
Konfiguracja układu FPGA znajdującego się na płycie **DE_1** może być realizowana w dwu trybach:

- **JTAG (Joint Test Action Group)** – w tym przypadku kod wynikowy kompilacji przesyłany jest bezpośrednio do układu FPGA – Cyclone II _ EP2C20F484C7. Układ FPGA zaprogramowany w tym trybie pamięta dane konfiguracyjne tak długo, jak długo włączone jest zasilanie. Zaletą tego rozwiązania jest możliwość przeprowadzenia nieskończonej liczby poprawnych przeprogramowań pamięci SRAM, natomiast wadą utrata programu po wyłączeniu zasilania.
- **AS (Active Serial)** – w tym przypadku układ FPGA wykorzystuje pamięć typu FLASH, do której wpisywane są informacje o konfiguracji. Dane te są pamiętane również po wyłączeniu zasilania. Nie ma potrzeby ponownego konfigurowania układu.

1. Programowanie w trybie JTAG

W celu zaprogramowania układu w trybie **JTAG** należy ustawić przełącznik **RUN/PROG** znajdujący się na płycie **Altera DE_1** w pozycję **RUN**. Moduł umożliwiający konfigurację wywołujemy wybierając z menu *Tools > Programmer* lub klikając ikonę . 

W module tym wybieramy tryb programowania **JTAG** oraz sprzęt docelowy *USB-Blaster* (za pomocą polecenia *Hardware Setup*).



W polu *File* możemy zauważyć plik o nazwie „*1_2_D_BRAMKI.sof*”, a w polu *Device* znajdujący się na płycie **DE_1** układ FPGA – **EP2C20F484C7**. W przypadku braku na liście wymienionego pliku należy go dodać wybierając *Add File*. W oknie dialogowym modułu *Programmer* należy zaznaczyć opcję w polu *Program/Configure*.

Konfigurację układu **EP2C20F484C7** rozpoczynamy klikając na *Start* .

Po przeprowadzeniu konfiguracji możemy przystąpić do fizycznego testowania przygotowanego projektu „**1_2_D_BRAMKI**”.

Materiały źródłowe

- [1] DE1 Development and Education Board User Manual (DE1_UserManual_v1018.pdf)
- [2] Artur Cichowski, Wojciech Śleszyński: Wprowadzenie do systemu projektowego Quartus II PG WEiA 2007 (cwiczenie_1_2_QUATRUS_II.pdf)
- [3] Quartus II Introduction Using Schematic Design (tut_quartus_intro_schem.pdf)
- [4] Krystyna M. Noga, Marcin Radwański: Projektowanie układów programowalnych w środowisku Quartus II z wykorzystaniem edytora tekstowego, AM Gdynia 2008
- [5] <http://www.altera.com>