

PLYTKA TESTOWA

Do praktycznego testowania realizowanych projektów laboratoryjnych przeznaczona jest płytki testowa. Na płytce znajdują się następujące komponenty:

- XC3S200FT256 – układ FPGA firmy Xilinx serii Spartan-3,
- XCF02S – programowalna pamięć konfiguracyjna 2Mbit,
- dwie pamięci SRAM 256K×16,
- porty VGA, RS-232 i PS/2,
- generator kwarcowy 50MHz,
- osiem diod LED,
- poczwórny wyświetlacz 7-segmentowy LED,
- osiem przełączników,
- cztery przyciski monostabilne,
- trzy listwy łączeniowe.

Diody LED świecą po podaniu poziomu H na odpowiednią końcówkę układu FPGA:

LED	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
Końcówka FPGA	P11	P12	N12	P13	N14	L12	P14	K12

Naciśnięcie przycisku monostabilnego ustala poziom H na odpowiedniej końcówce układu FPGA:

Przycisk	BTN3	BTN2	BTN1	BTN0
Końcówka FPGA	L14	L13	M14	M13

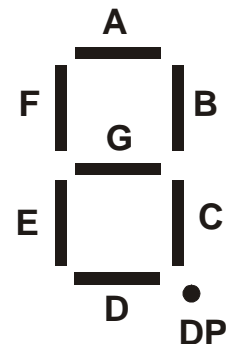
Osiem przełączników umożliwia ustalenie poziomu H (pozycja UP) lub poziomu L (pozycja DOWN) na końcówkach układu FPGA:

Przełącznik	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
Końcówka FPGA	K13	K14	J13	J14	H13	H14	G12	F12

Wyświetlanie informacji na wyświetlaczu 7-segmentowym odbywa się multipleksowo. Katody odpowiednich segmentów są podłączone wspólnie do końcówek układu FPGA i świecą po podaniu poziomu L. Natomiast anody poszczególnych cyfr sterowane są indywidualnie poziomem L.

Segment	A	B	C	D	E	F	G	DP
Końcówka FPGA	E14	G13	N15	P15	R16	F13	N16	P16

Anoda	AN3	AN2	AN1	AN0
Końcówka FPGA	E13	F14	G14	D14



Generator kwarcowy dostarcza sygnał o częstotliwości 50MHz do układu FPGA poprzez końcówkę T9.

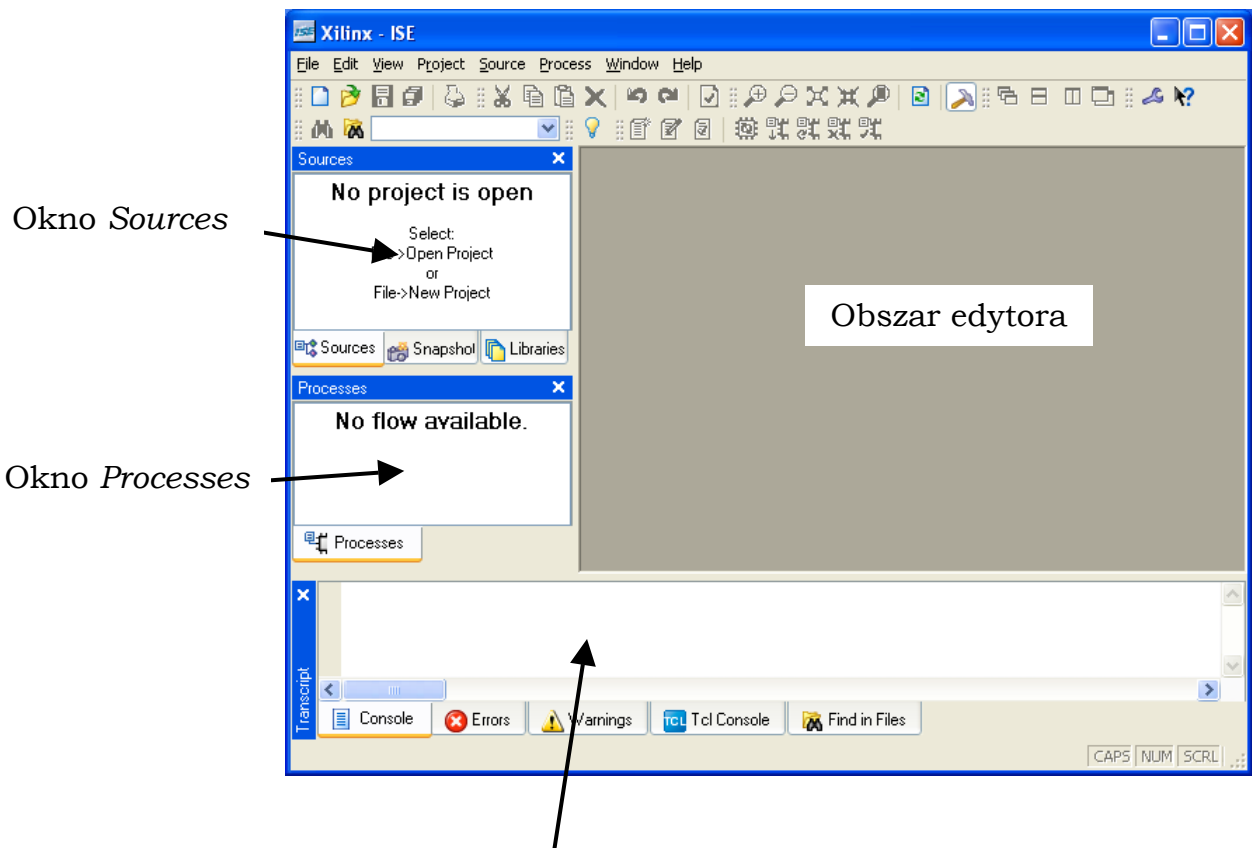
Programowanie układów XC3S200 i XCF02S wykonuje się za pomocą komputera PC poprzez złącze JTAG.

PROCES PROJEKTOWANIA

Możliwości systemu projektowego *WebPACK ISE 8.2i* ilustruje projekt licznika moduło 16. Opis działania licznika jest przygotowany w języku VHDL.

Uruchomienie systemu

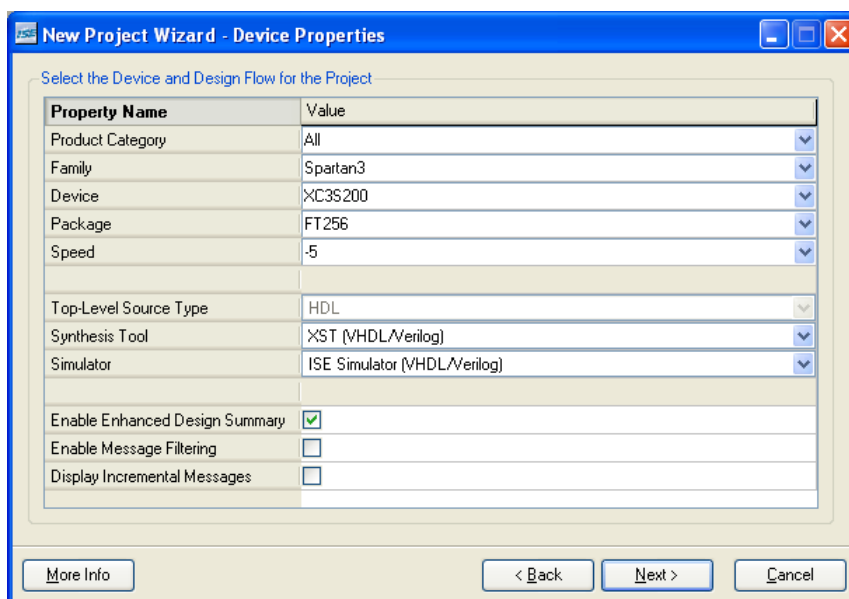
System uruchamia się dwukrotnym kliknięciem lewego klawisza myszki na ikonie *Xilinx ISE 8.2i*. Po uruchomieniu programu pojawia się nawigator projektu *Xilinx – ISE*.



Okno *Transcript* – wyświetlanie komunikatów

W celu utworzenia nowego projektu typu *HDL* należy:

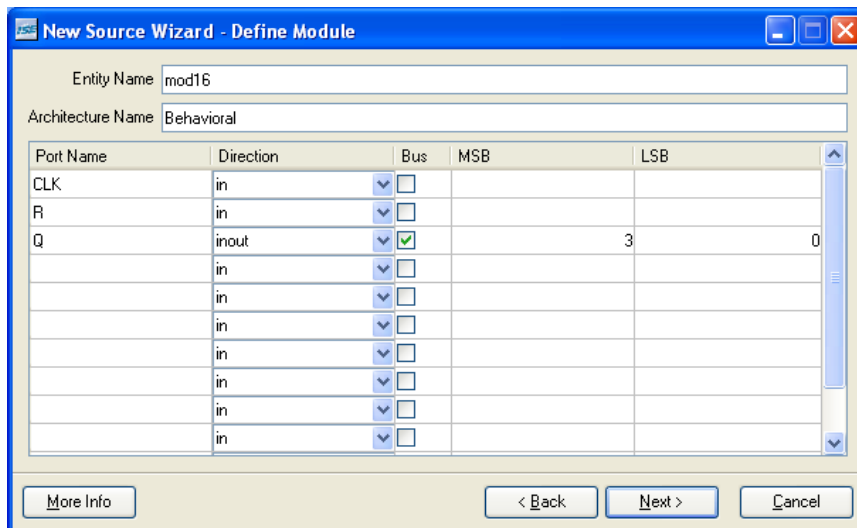
1. Wybrać *File* → *New Project...* uruchamiając *New Project Wizard*
2. Wpisać nazwę projektu *proj* w polu *Project Name*
3. W polu *Project Location* ustawić folder *...\Labuprog* dla lokalizacji projektu
4. Sprawdzić czy w polu *Top-Level Source Type* jest wybrana opcja *HDL*
5. Nacisnąć *Next*
6. Wypełnić pola okna *New Project Wizard – Device Properties* jak poniżej



7. Nacisnąć *Next*
8. W kolejnych dwóch oknach *Create New Source* i *Add Existing Sources* nacisnąć *Next*
9. W oknie *Project Summary* nacisnąć *Finish*

W celu opisanania licznika modulo 16 w języku VHDL należy:

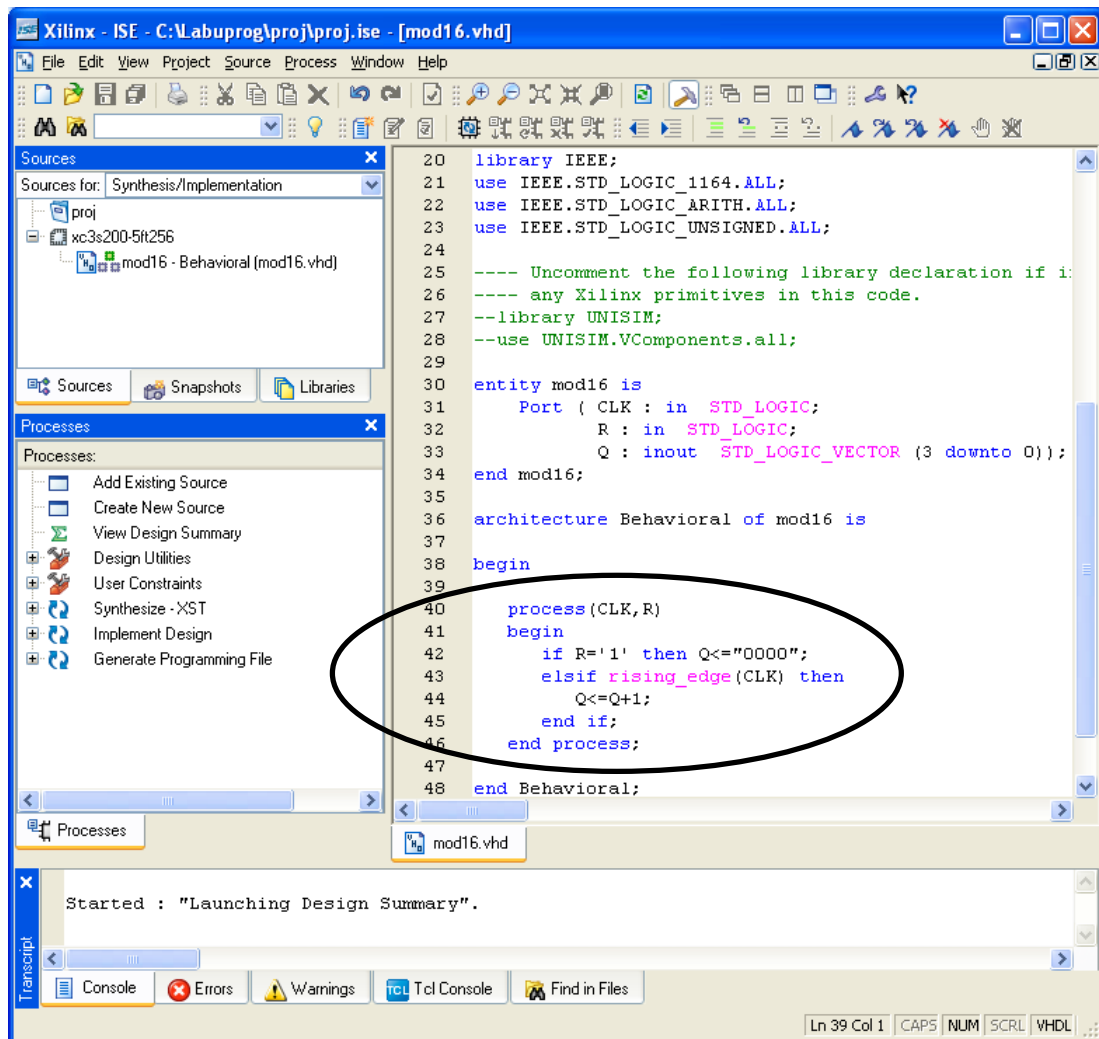
1. Wybrać *Project* → *New Source...* uruchamiając *New Source Wizard*
2. W oknie *Select Source Type* zaznaczyć typ *VHDL Module*
3. W polu *File Name* wpisać nazwę pliku *mod16*
4. W polu *Location* domyślnie powinien być wybrany folder *...\Labuprog\proj*
5. Nacisnąć *Next*
6. W oknie *Define Module* wypełnić odpowiednie kolumny nazwami sygnałów i ich parametrami jak poniżej



7. Nacisnąć *Next*
8. W oknie *Summary* nacisnąć *Finish*
9. W utworzonym pliku *mod16.vhd* „dopisać” działanie licznika w części „architecture”, zapisać plik *mod16.vhd*

```

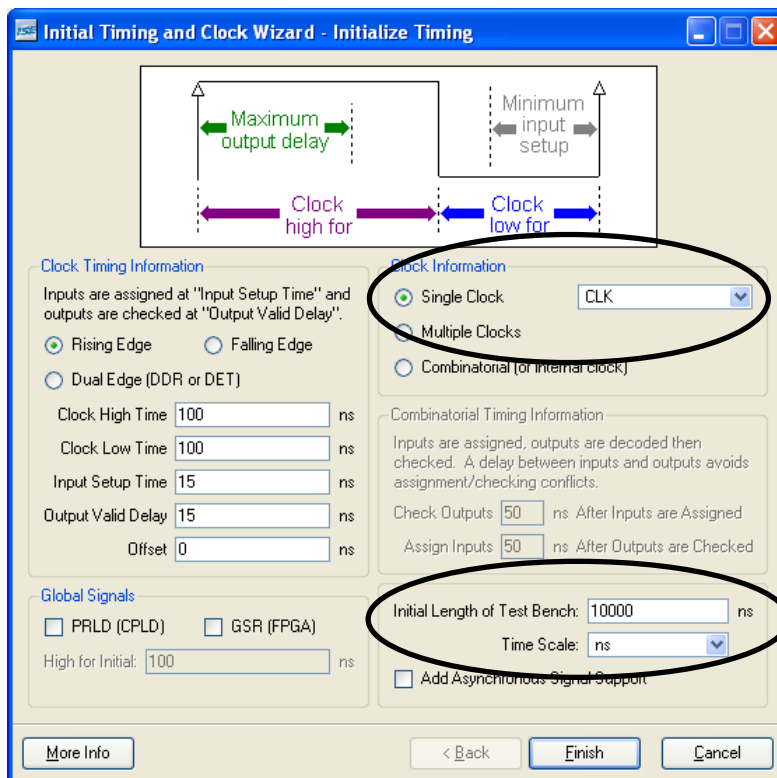
process ( CLK , R )
begin
    if R='1' then Q<="0000";
    elsif rising_edge(CLK) then
        Q<=Q+1;
    end if;
end process;
    
```



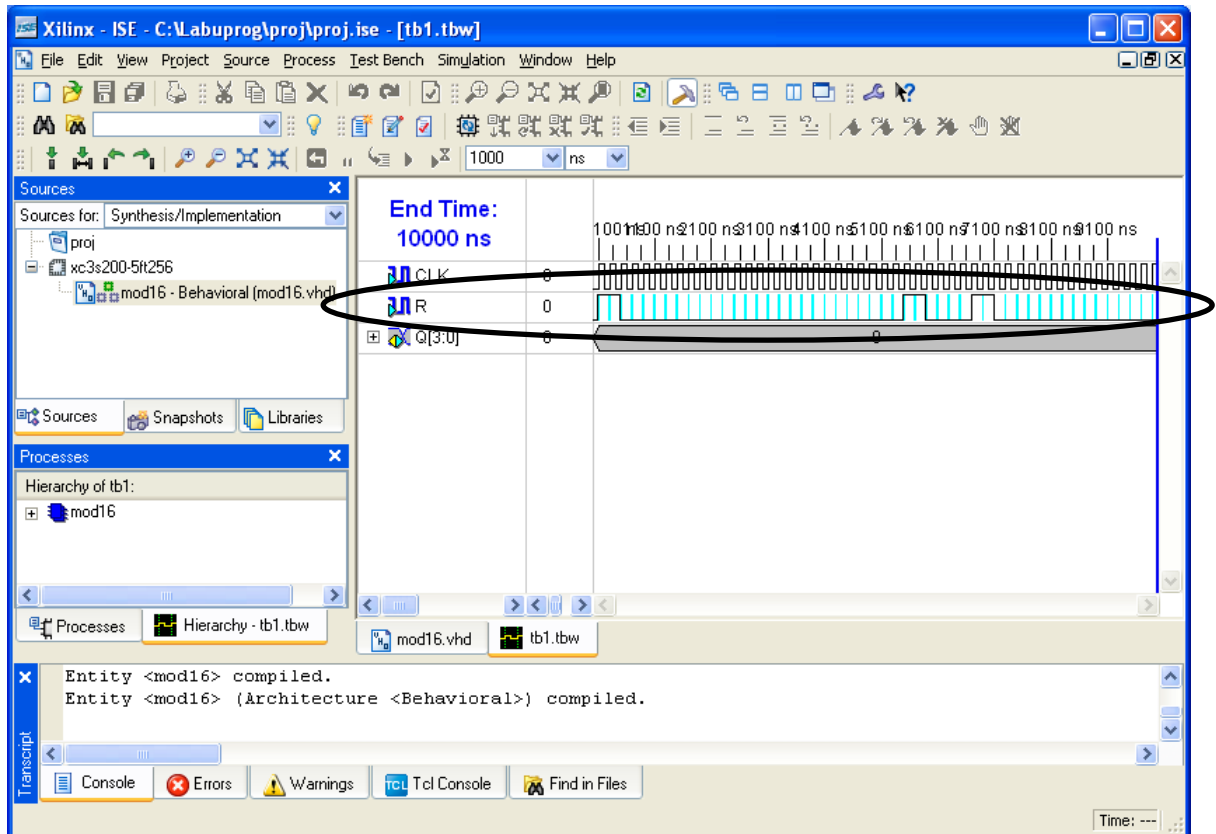
- W celu sprawdzenia poprawności składni języka VHDL w oknie *Processes* rozwinąć kolejno wiersze *Implement Design* oraz *Synthesize – XST*, następnie uruchomić *Check Syntax*

W celu przeprowadzenia symulacji funkcjonalnej działania licznika należy:

1. Wybrać *Project* → *New Source...* uruchamiając *New Source Wizard*
2. W oknie *Select Source Type* zaznaczyć typ *Test Bench WaveForm*
3. W polu *File Name* wpisać nazwę pliku *tb1*
4. W polu *Location* domyślnie powinien być wybrany folder *...\Labuprog\proj*
5. Nacisnąć *Next*
6. W oknie *Associate Source* powinien być zaznaczony *mod16*, nacisnąć *Next*
7. W oknie *Summary* nacisnąć *Finish*
8. W oknie *Initialize Timing* w obszarze *Clock Information* wybrać *Single Clock* oraz sygnał *CLK*, ponadto w polu *Initial Length of Test Bench* ustawić czas symulacji 10000 ns, nacisnąć *Finish*



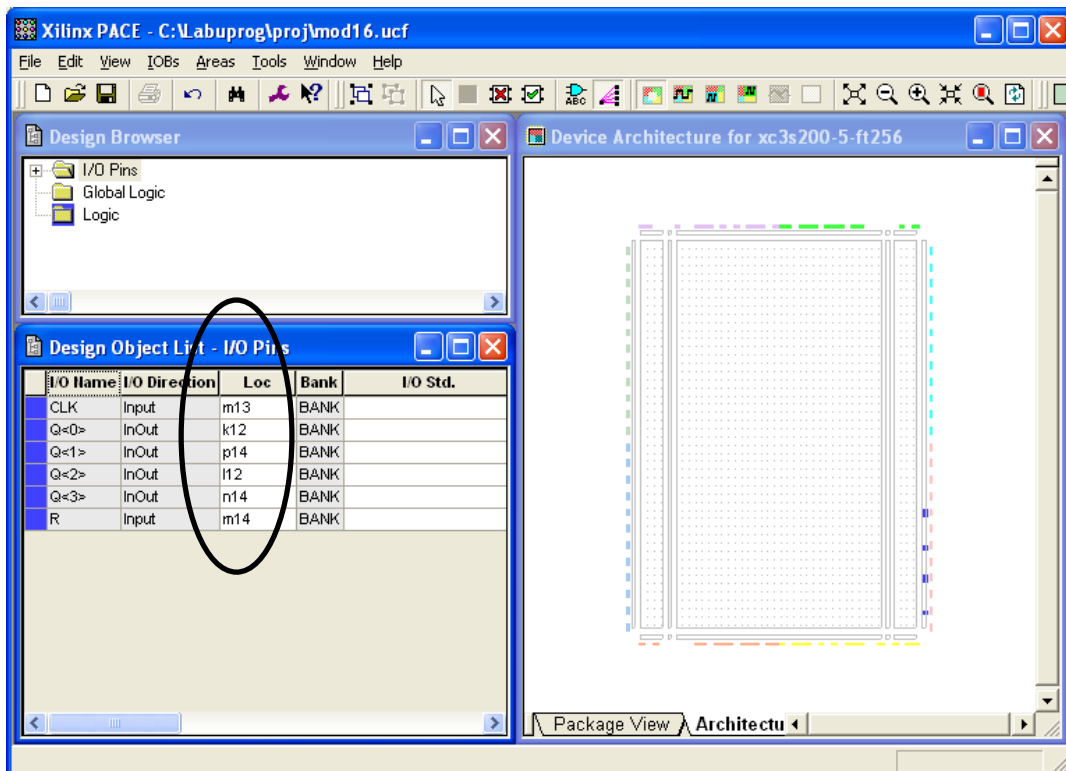
9. W oknie nawigatora projektu przy użyciu myszki „narysować” przebieg sygnału *R*, zapisać plik *tb1.tbw*



10. W oknie *Sources* wybrać *Behavioral Simulation* oraz zaznaczyć *tb1* (*tb1.tbw*)
11. W oknie *Processes* wybrać zakładkę *Processes*, rozwinąć wiersz *Xilinx ISE Simulator* i uruchomić *Simulate Behavioral Model*
12. Wynik symulacji w postaci przebiegów sygnału *Q[3:0]*

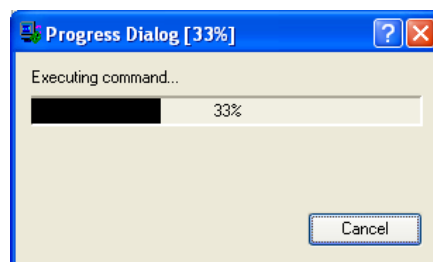
Implementacja projektu licznika w układzie XC95108

1. W oknie *Sources* wybrać *Synthesis/Implementation* oraz zaznaczyć *mod16*
2. W oknie *Processes* rozwinąć wiersz *User Constraints* i uruchomić *Assign Package Pins*
3. W kolejnym oknie informacyjnym nacisnąć *Yes*
4. W oknie *Xilinx PACE* (edytor pliku *mod16.ucf*) przypisać sygnałom końcówki układu scalonego jak poniżej (kolumna *Loc*)



5. Zapisać plik *mod16.ucf*, w oknie *Bus Delimiter* nacisnąć *OK*.
6. W oknie *Processes* uruchomić *Generate Programming File*
7. **Sprawdzić podłączenie płytki testowej!**
8. Następnie rozwinąć wiersz *Generate Programming File* i uruchomić *Configure Device (iMPACT)*
9. W oknie *Welcome to iMPACT* wybrać *Configure devices using Boundary-Scan (JTAG)* oraz *Automatically connect a cable and identify Boundary-Scan chain*, nacisnąć *Finish*

10. Pojawia się okno *Assign New Configuration File*, w którym wybieramy plik *mod16.bit*, nacisnąć *Open*, w oknie *Warning* nacisnąć *OK*.
11. W kolejnym oknie *Assign New Configuration File* nacisnąć *Cancel*
12. Nacisnąć prawym klawiszem myszki na symbolu układu XC3S200 i wybrać *Program...*
13. W oknie *Programming Properties* powinna być nie aktywna opcja *Verify*, następnie nacisnąć *OK*
14. Programowanie w toku...



W celu przeprowadzenia symulacji czasowej działania licznika należy:

1. W oknie *Sources* wybrać *Post – Route Simulation* oraz zaznaczyć *tb1 (tb1.tbw)*
2. W oknie *Processes* rozwinąć wiersz *Xilinx ISE Simulator* i uruchomić *Simulate Post – Place & Route Model*
3. Wynik symulacji w postaci przebiegów sygnału *Q[3:0]*