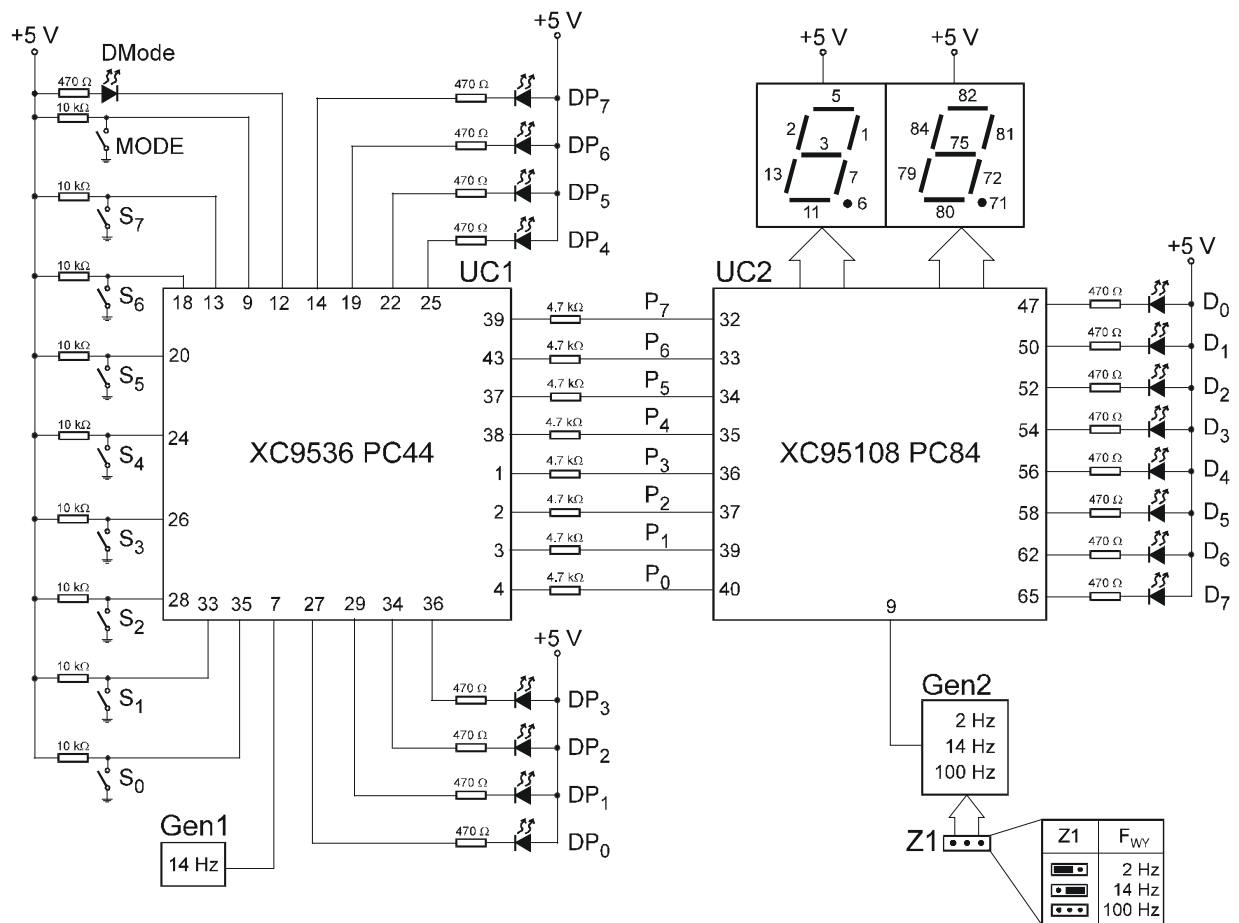


PLYTKA TESTOWA

Do praktycznego testowania realizowanych projektów laboratoryjnych przeznaczona jest płytki testowa. Na płytce znajdują się dwa układy programowalne CPLD:

- UC1 – XC9536 PC44,
- UC2 – XC95108 PC84.

Układ UC1 wytwarza sygnały $P_0 \div P_7$ na podstawie stanu przycisków $S_0 \div S_7$ eliminując wpływ drgań ich styków. Diody $DP_0 \div DP_7$ świecą, gdy odpowiedni sygnał $P_0 \div P_7$ ma poziom wysoki. Przycisk MODE służy do wyboru pracy sygnałów $P_0 \div P_7$: generacja pojedynczego impulsu, ustawianie poziomu niskiego lub

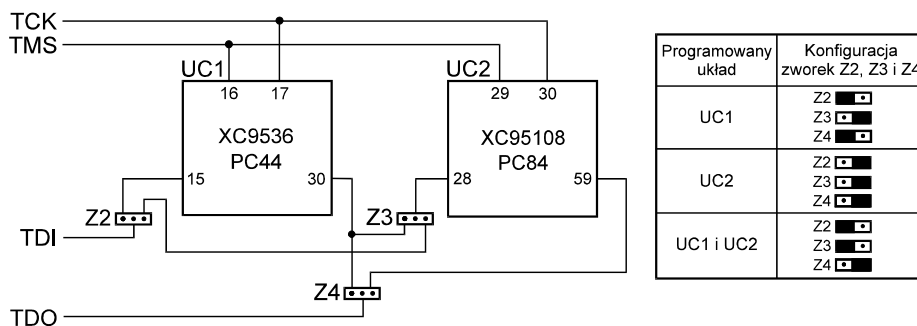


Uproszczony schemat płytki testowej

wysokiego. Generator Gen1 (NE555) wytwarza sygnał prostokątny o częstotliwości 14 Hz.

Układ UC2 przeznaczony jest do testowania realizowanych projektów. Sygnały $P_0 \div P_7$ doprowadzone są do wybranych końcówek układu UC2 poprzez rezystory ograniczające prąd (w przypadku zaprogramowania końcówek jako wyjść). Do układu przyłączono osiem diod świecących $D_0 \div D_7$ oraz dwa wyświetlacze 7-segmentowe LED, które świecą po podaniu poziomu L na odpowiednie końcówki. Ponadto do układu UC2 doprowadzono z generatora Gen2 (NE555) sygnał zegarowy o częstotliwości wybieranej za pomocą zworki Z1 (2 Hz, 14 Hz, 100 Hz).

Programowanie układów wykonuje się za pomocą komputera PC poprzez złącze JTAG. Sposób połączenia sygnałów złącza i wyboru programowanego układu za pomocą zworek Z2, Z3 i Z4 przedstawia poniższy rysunek.



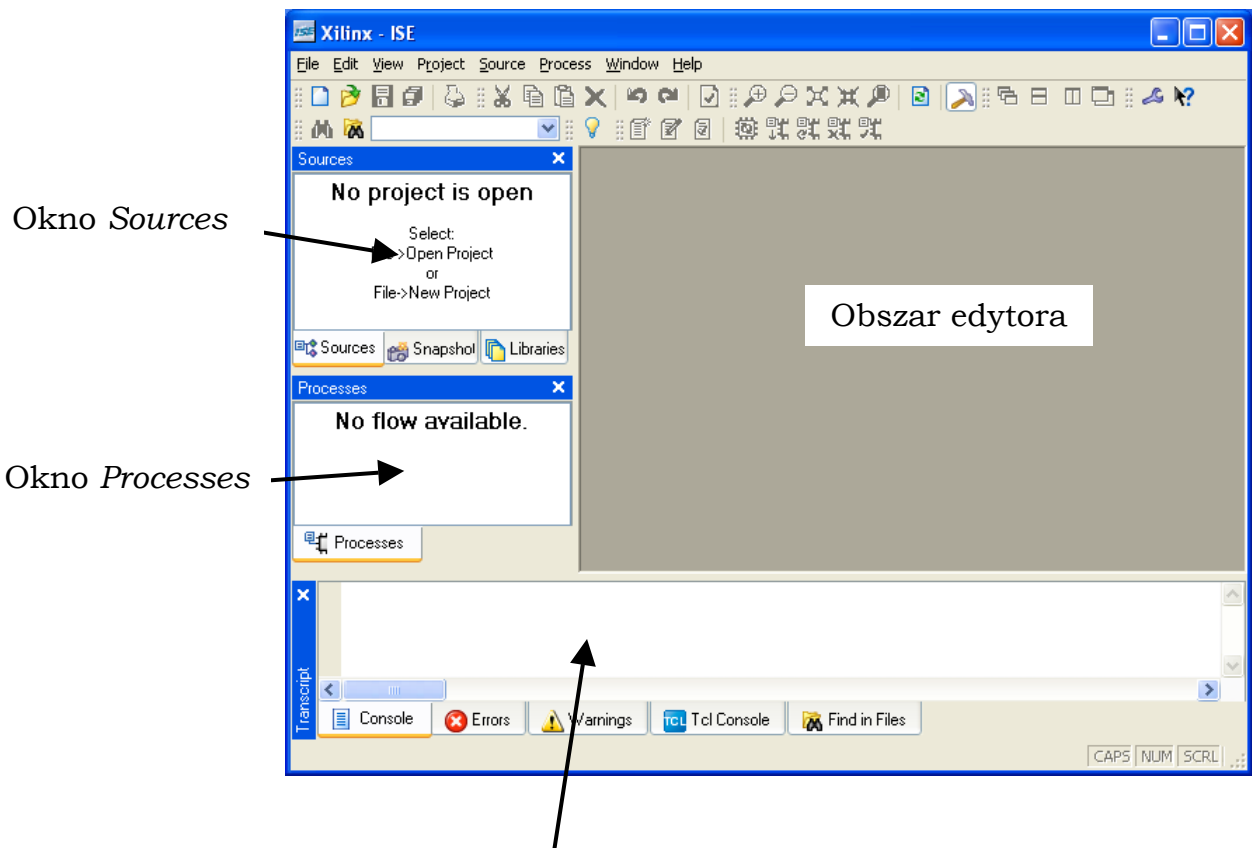
Konfigurowanie interfejsu JTAG na płycie testowej

PROCES PROJEKTOWANIA

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

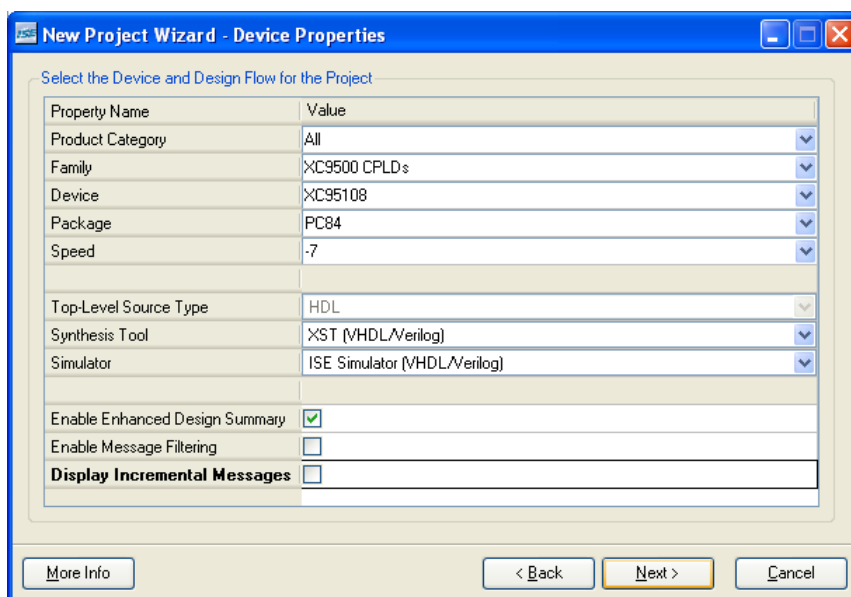
Uruchomienie systemu

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



W celu utworzenia nowego projektu typu HDL należy:

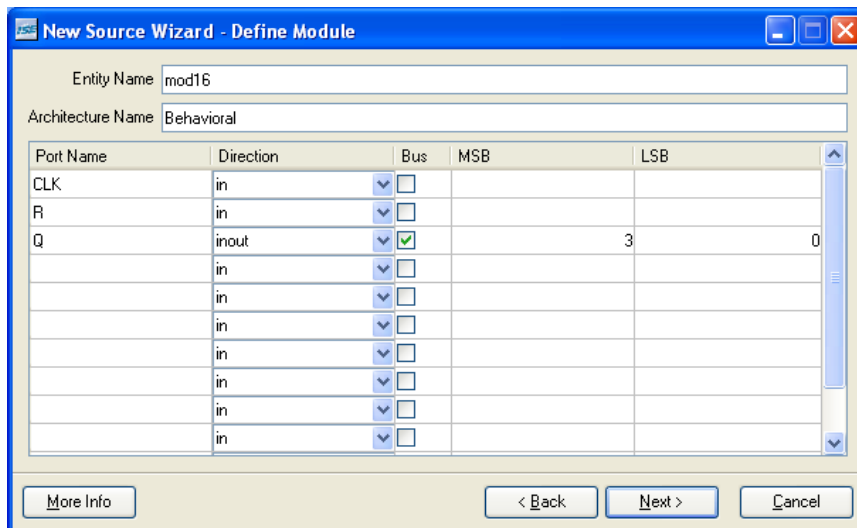
1. Wybrać *File* → *New Project...* uruchamiając *New Project Wizard*
2. Wpisać nazwę projektu *proj* w polu *Project Name*
3. W polu *Project Location* ustawić folder *...\Labuprog* dla lokalizacji projektu
4. Sprawdzić czy w polu *Top-Level Source Type* jest wybrana opcja *HDL*
5. Nacisnąć *Next*
6. 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 opisanego licznika modułu 16 w języku VHDL należy:

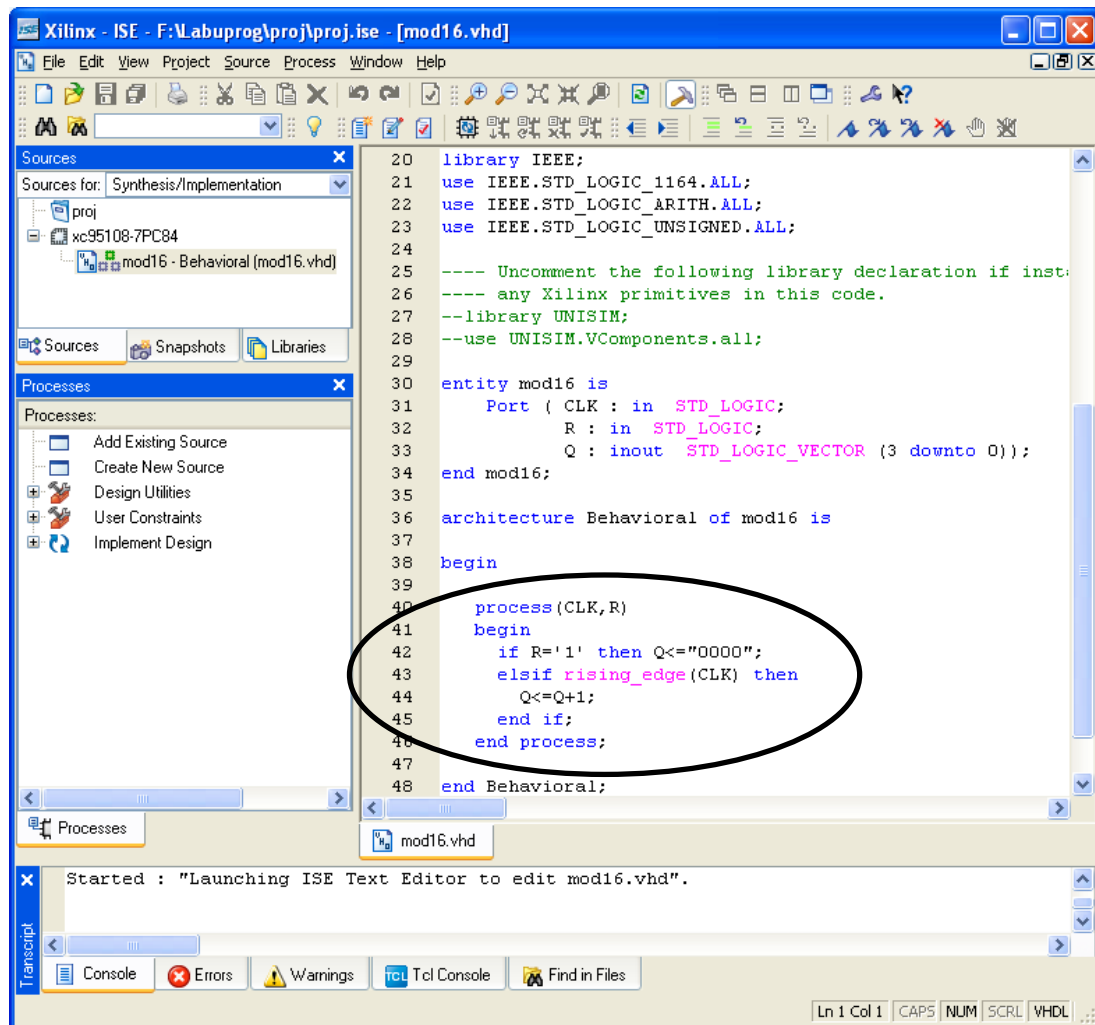
1. Wybrać *Project* → *New Source...* uruchamiając *New Source Wizard*
2. W oknie *Select Source Type* zaznaczyć typ *VHDL Module*
3. W polu *File Name* wpisać nazwę pliku *mod16*
4. W polu *Location* domyślnie powinien być wybrany folder *...\Labuprog\proj*
5. Nacisnąć *Next*
6. W oknie *Define Module*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

```

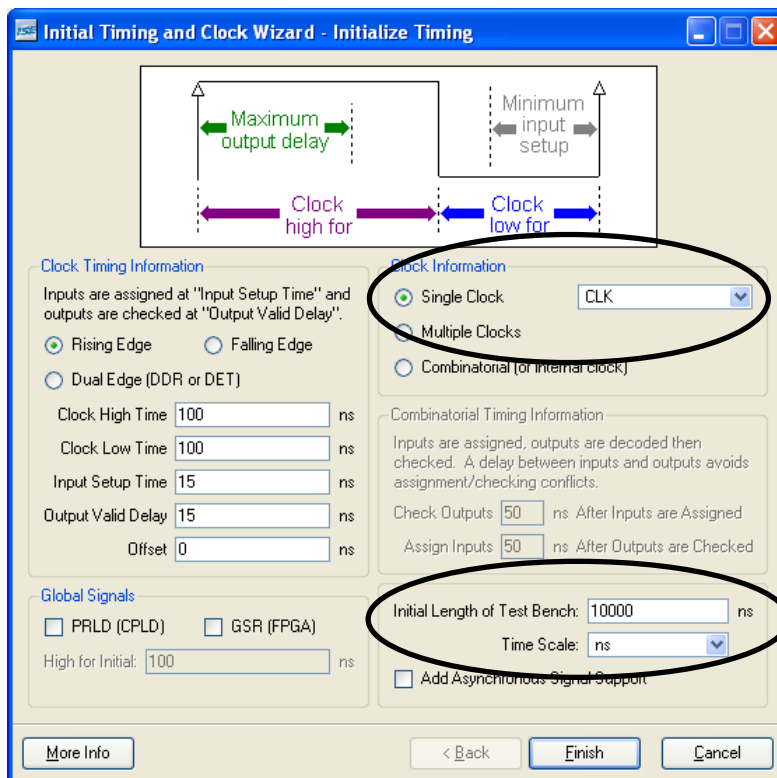
process ( CLK , R )
begin
    if R='1' then Q<="0000";
    elsif rising_edge(CLK) then
        Q<=Q+1;
    end if;
end process;
    
```



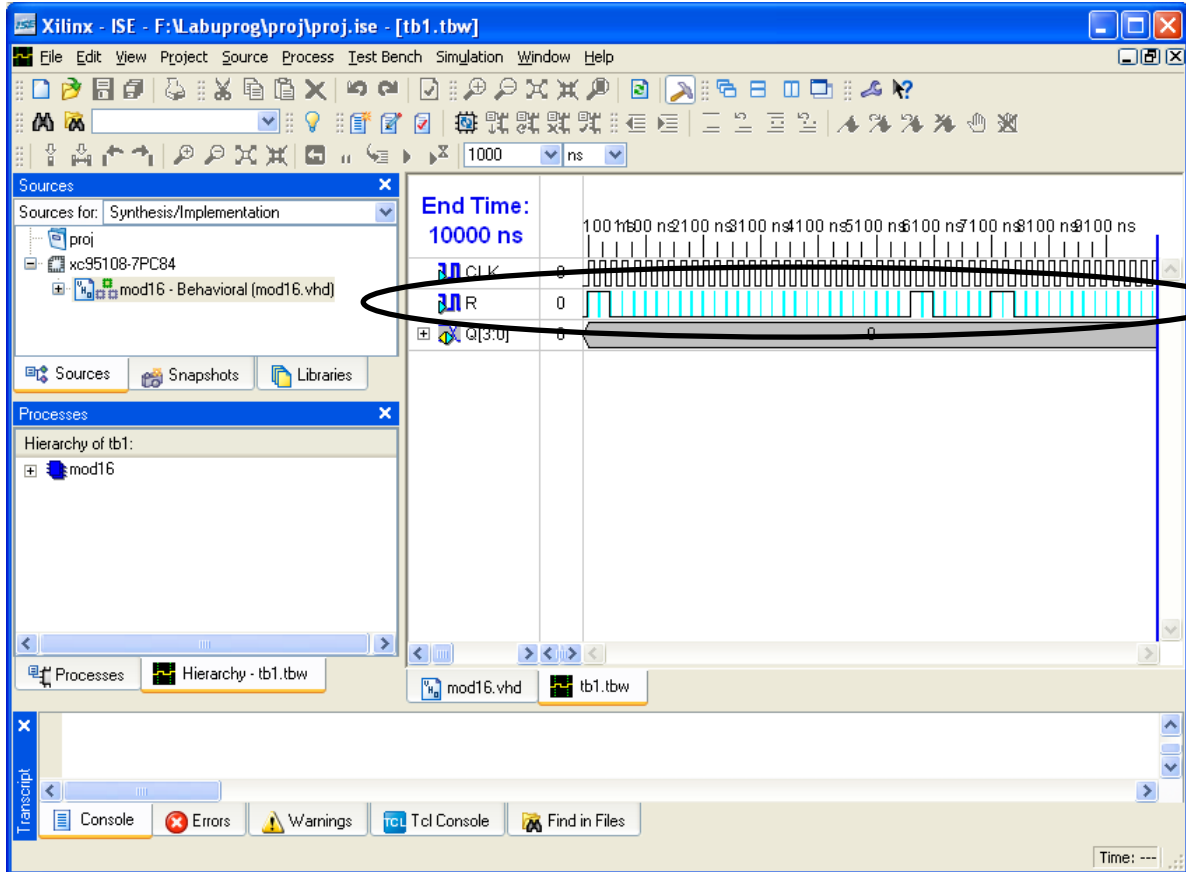
10. W celu sprawdzenia poprawności składni języka VHDL w oknie *Processes* rozwinąć kolejno wiersze *Implement Design* oraz *Synthesize – XST*, następnie uruchomić *Check Syntax*

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*



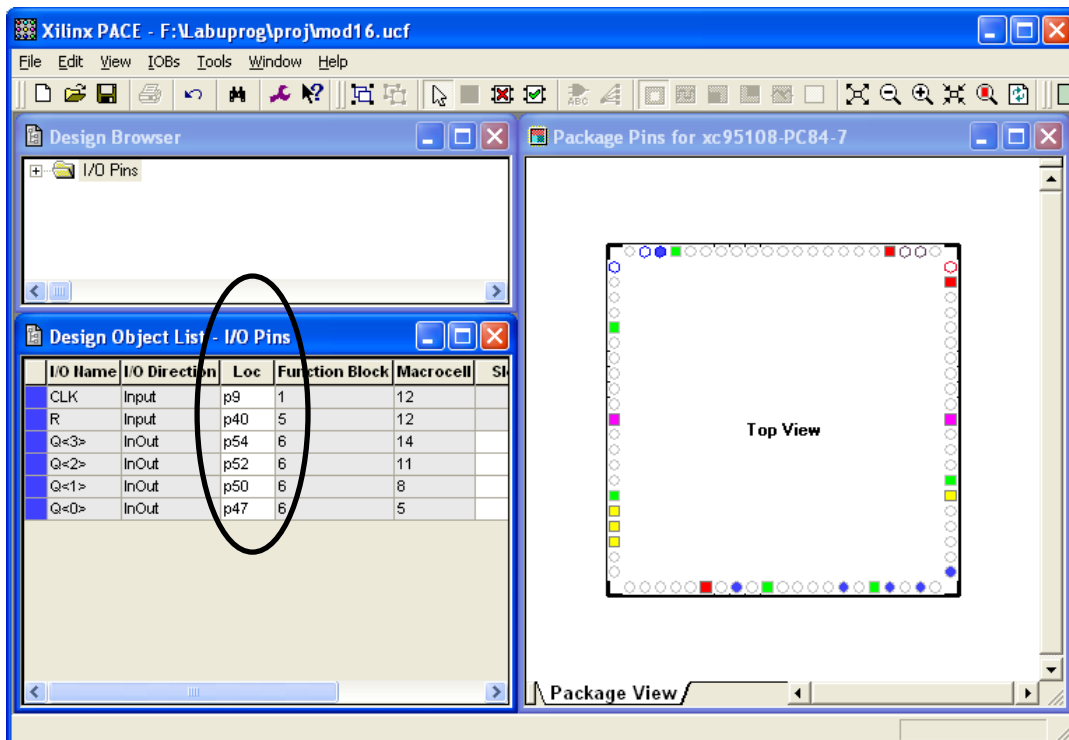
9. W oknie nawigatora projektu przy użyciu myszki „narysować” przebieg sygnału *R*, zapisać plik *tb1.tbw*



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]*

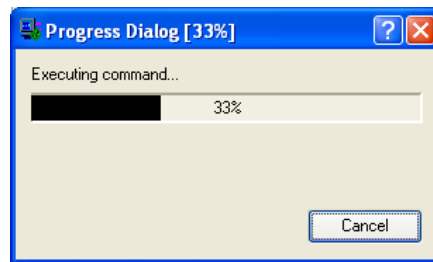
Implementacja projektu licznika w układzie XC95108

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* rozwinąć wiersz *Implement Design* i uruchomić *Generate Programming File*, w oknie edytora pojawi się raport
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.jed*, naciskamy *Open*
11. Nacisnąć prawym klawiszem myszki na symbolu układu XC95108 i wybrać *Program...*
12. Nacisnąć *OK* w oknie *Programming Properties*
13. Programowanie w toku...



W celu przeprowadzenia symulacji czasowej działania licznika należy:

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