

# WebPACK ISE5.1i Manual

Insight Korea Xilinx FAE Team

2003. 3. 10

WebPACK ISE 5.1i( 이하 WebPACK ) 은 Xilinx FPGA나 CPLD를 쉽게 디자인할 수 있게 하는 Free Design Software로서 Design Entry, Synthesis, 그리고 Verification , Simulation을 할 수 있는 통합환경을 제공합니다.

주의) ISE 5.1은 Windows XP and Windows 2000 이상의 OS에서만 지원합니다.

(Windows98에서는 ISE5.1i 을 이용하실 수 없습니다. Xilinx Tools 이전 버전을 이용하셔야 합니다.)

참고) 1.WebPACK ISE 5.1i에서는 CORE Generator를 이용하실 수 없습니다.

2.WebPACK ISE에서5.1i는 XC4000 계열의 디바이스를 지원하지 않습니다.

( XC4000, Spartan/ SpartanXL )

자세한 S/W의 지원사항에 대한 내용은 아래의 파일을 참조하시기 바랍니다.

<http://www.xilinx.com/publications/matrix/softmatrix.pdf>

# 목 차

1. Webpack ISE5.1i 설치하기
2. WebPACK ISE5.1i 실행하기
3. 프로젝트 작성
4. Synthesize 하기
5. Implementation 하기
6. User Constraints ( Pin Assignment )
7. Modelsim Simulation
8. Downloading
9. Revision Table

# 1.Webpack ISE5.1i 설치하기

Webpack을 아래의 Xilinx 홈페이지에서 내려 받을 수 있습니다. 처음 방문하시는 분은 간단한 등록 절차를 요구합니다.

[http://www.xilinx.com/xlnx/xil\\_prodcats/landingpage.jsp?title=ISE+ WebPack](http://www.xilinx.com/xlnx/xil_prodcats/landingpage.jsp?title=ISE+ WebPack)

등록 후 Log-in을 하시면 Download ISE WebPACK을 클릭하면 아래와 같은 창이 뜹니다.

ISE WebPACK Download Module	Download Size	CPLD Design Environment	FPGA Design Environment	CPLD Programming	FPGA Programming	ChipViewer, XPower HTML Reporting
<a href="#">Complete ISE WebPACK Software</a>	171 Mb*	✓	✓	✓	✓	✓
<a href="#">Complete CPLD Tool Set</a>	108 Mb*	✓		✓		✓
<a href="#">CPLD Design Environment</a>	87 Mb*	✓				
<a href="#">CPLD Programming Tools</a>	27 Mb*			✓		
<a href="#">CPLD Optional Tools</a>	49 Mb*					✓
<a href="#">Complete Programming Tools</a>	49 Mb*			✓	✓	

\* Customers installing the above WebPACK modules also need to download and install the latest Service Pack. At this time, [Service Pack 3](#) is available. Service pack installation instruction and update contents are located in the [README](#).

< 그림 1 > WebPack Home

이 Tutorial에서는 “Complete ISE Webpack Software”를 선택합니다.

주의) “WebPACK\_51\_fcfull\_i.exe”와 Service Pack3도 함께 설치합니다.

Simulation Tools은 기본적으로 Webpack에 제공되지 않습니다. 아래의 창에서와 같이 Modelsim MXE version “MXE\_5.6a\_Full\_installer.exe”을 내려 받아 추가 설치합니다.

ModelSim XE Download Module	Download Size	MXE Simulator	CPLD VHDL Library	CPLD Verilog Library	FPGA VHDL Library	FPGA Verilog Library
<a href="#">CPLD MXE Simulator</a>	18 Mb	✓	✓	✓		
<a href="#">Complete MXE Simulator</a>	65 Mb	✓	✓	✓	✓	✓

< 그림 2 > MXE downloading window

Tools의 설치 순서는 WebPACK\_51\_fcfull.i.exe 와 Servicepack3를 설치한 후, MXE\_5.6a\_Full\_installer.exe 의 순으로 설치를 합니다.

## 2. WebPACK ISE5.1i 실행하기

Webpack을 실행하기 위하여 아래와 같이 바탕화면에 있는 Project Navigator를 클릭합니다. .

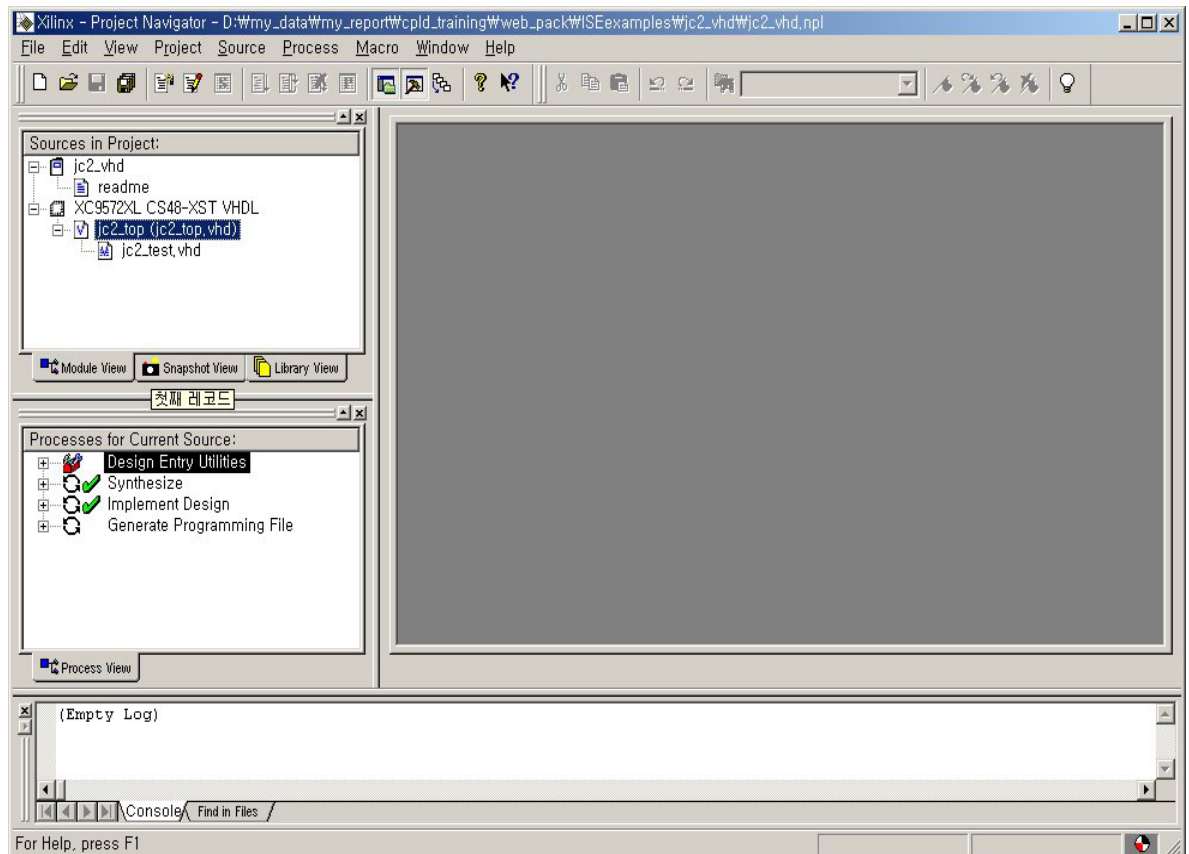


< 그림 3 > 바탕 화면의 Project Navigator

## 3. 프로젝트 작성

WebPACK을 실행하기 위하여 Project Navigator를 클릭하게 되면 아래와 같은 Project Navigator라는 window가 나타납니다. .

이 통합환경은 주로 네 가지 창으로 구성되어있다. 프로젝트 관리 및 각종소스데이터를 관리 하는 **Source in Project** 와 그때 사용된 소스의 Process의 상태를 나타내는 **Process for Current Source** 와 각종 명령 시 진행사항을 나타내는 **Console**창과 각종 소스 코드를 display해주는 **Editor**창으로 이루어져 있습니다.



< 그림 4 > Project Navigator window

## 3.1 Making New project

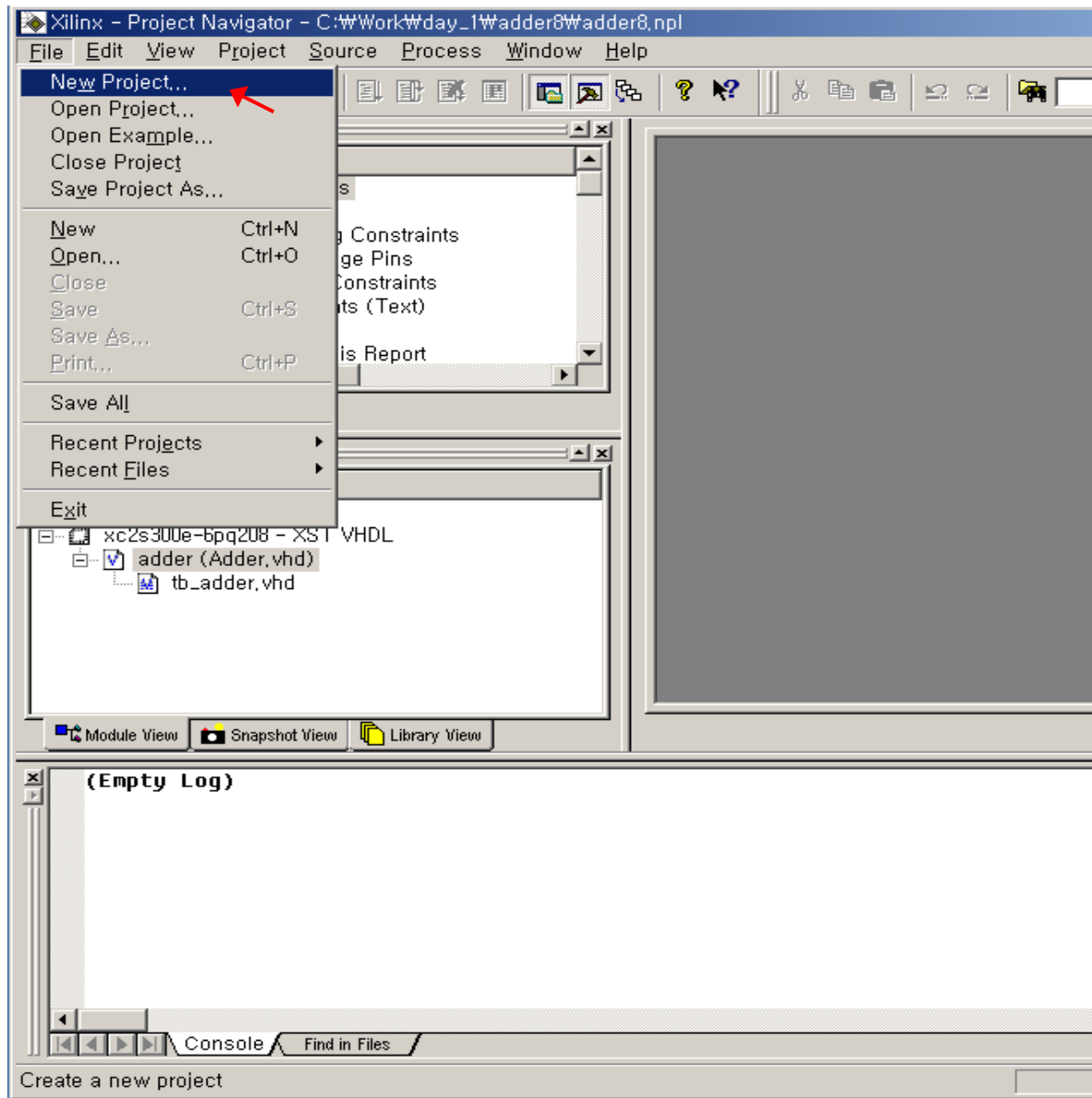
4-bit Johnson counter 예제를 이용하여 새로운 프로젝트를 만들어 봅니다.

I/O PINS:

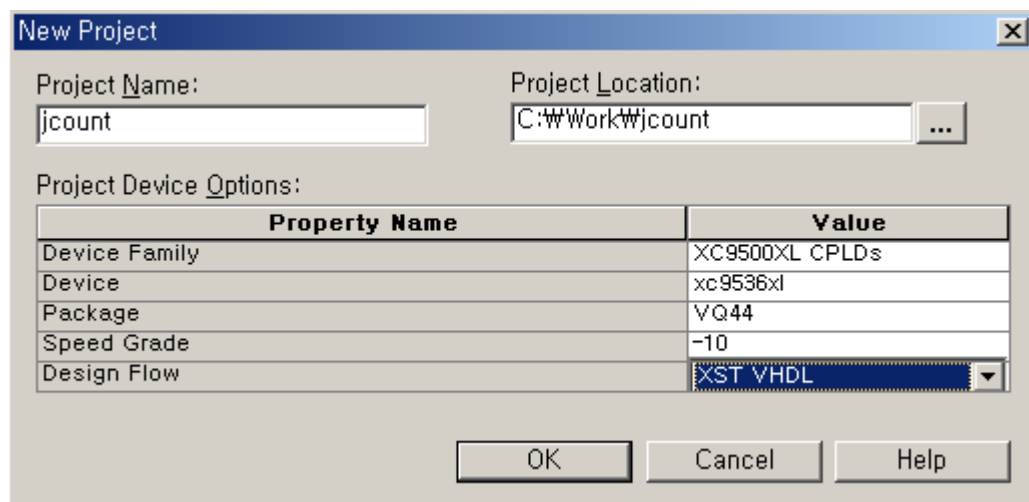
CLK : input free-running clock

Q<3>-Q<0> : counter outputs

새로운 프로젝트를 만들려면 Project Navigator 에서 **File > New Project**를 클릭하면 아래와 같은 <그림 5> New Project 창이 나타납니다.



< 그림 5 > Project Navigator 에서 New Project 생성



< 그림 6 > New Project 창

구성을 살펴보면

- Project Name : 작성할 프로젝트 이름
- Project Location : 작성할 프로젝트의 Working Directory 를 지정
- Device Family : 사용할 Device Family 선택
- Device : 사용될 Device Package 선택
- Design Flow : 사용될 Synthesis tool 선택

위와 같이 각 항목에 맞는 내용을 선택합니다.

<참조> XST 는 Xilinx Synthesis Tool 입니다.

New Project를 작성하고 OK버튼을 누르면 새로운 Project 의 구성이 완성됩니다.

앞으로 작성되는 모든 파일들은 Working Directory인 jc2\_vhd Folder내에서 생성됩니다.

탐색기를 열어, C:WORK 내의 파일들을 살펴봅니다.



< 그림 7 > 작업 디렉토리 내용

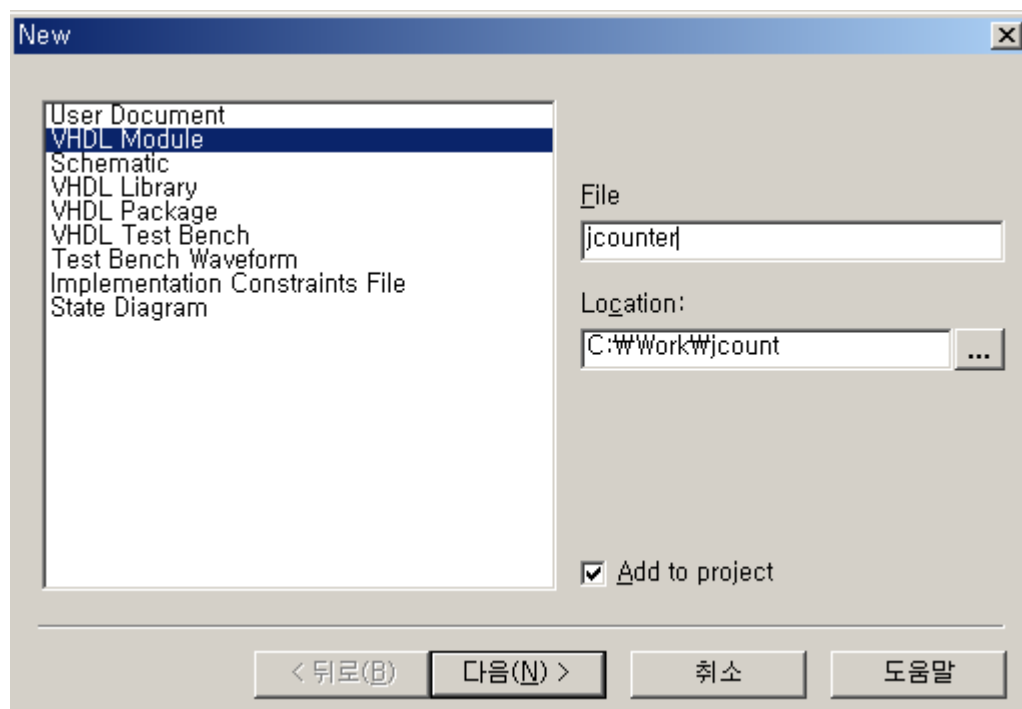
<참조> jc2\_vhd.npl 파일에는 아래와 같이 New Project 생성하면서 선택한 내용들이 들어있다.

```
// Created by Project Navigator ver 1.0
PROJECT demo_ise
DESIGN demo_ise Normal
DEVFAM xc9500xl
DEVFAMTIME 0
DEVICE xc9536xl
DEVICETIME 315558000
DEVPKG PC44
DEVPKGTIME 315558000
DEVSPEED -10
DEVSPEEDTIME 1034062411
FLOW XST VHDL
FLOWTIME 315558000
STIMULUS jc_tb.tbw Normal
```

MODULE JCOUNTER.VHD  
MODSTYLE jcounter Normal  
[STRATEGY-LIST]  
Normal=True

## 3.2 Making Program file

Project Navigator의 Menu에서 **Project > New Source**를 Click 하거나 Source in Project 창에서 마우스 오른쪽을 클릭하여 New source를 클릭하면 아래 같은 창이 뜹니다.



< 그림 8 > New Source 삽입 창

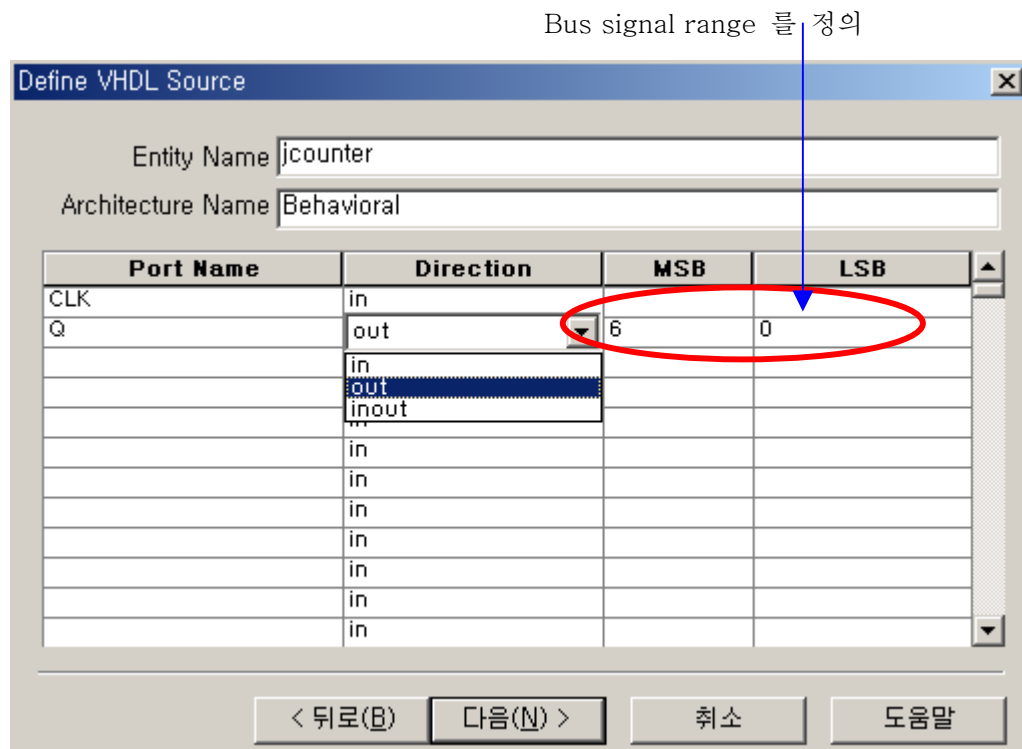
위와 같이 Johnson Counter를 VHDL코드로 작성을 하기위해 VHDL Module을 선택하고 File란에 jcounter 라고 기입합니다.

< 참조 > Source 의 종류를 간단히 살펴보면

- User Document : 사용자가 프로젝트에 대한 전반적인 설명이 가능하게 함.
- VHDL Module : 작성할 Source 로직이 VHDL 코드일 경우 사용.
- Schematic : 작성할 Source 로직을 schematic으로 작성할 경우 사용.
- VHDL Test Bench : ModelSim을 이용하여 시뮬레이션을 하기위해 Test Bench를 작성할 경우 사용.
- Implementation Constraints File : UCF 파일을 작성할 경우 사용.

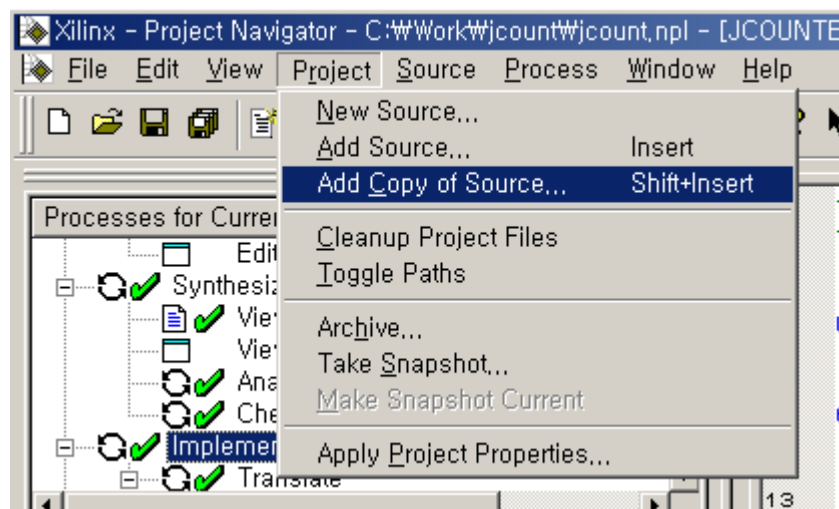


[다음]을 클릭하여 다음 단계로 넘어가면 아래와 같은 Define VHDL source 화면이 뜹니다.  
Port Name에 들어갈 이름과 Direction을 정하고 마침을 하면 Jcounter.vhd 란 VHDL파일이 생성됩니다.



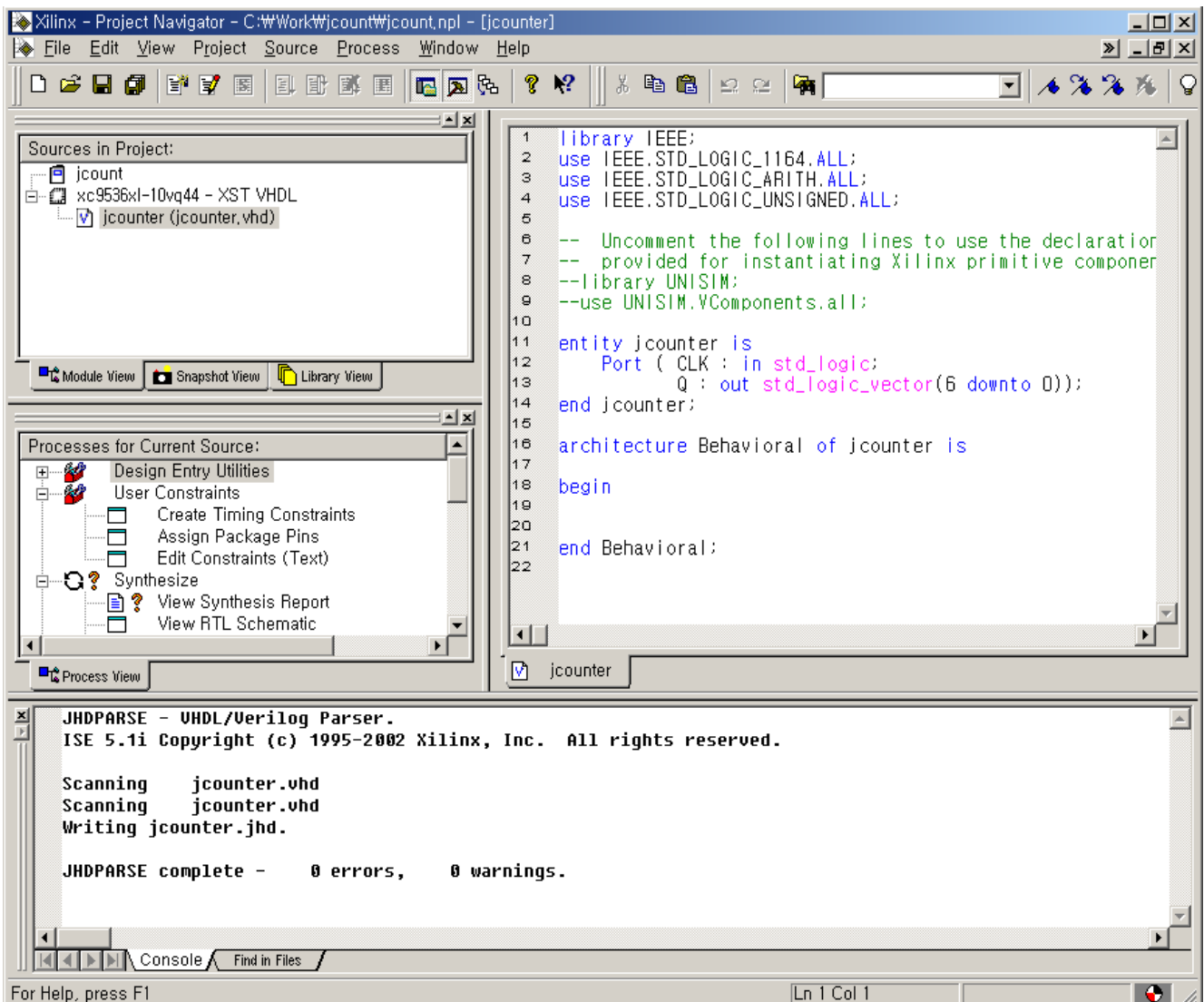
< 그림 9 > Define VHDL Source

만약 이전에 작성된 HDL Source를 이용할 경우에는 다음과 같은 메뉴를 이용합니다.  
**Project Navigator : Project -> Add copy of Source...**를 이용합니다.



< 그림 10 > Add Copy of source

“New Source...” 를 이용한 Source의 작성을 위한 모든 준비 작업이 끝나면 아래의 그림처럼 Project Navigator가 준비됩니다.



< 그림 11 > New Source로 VHDL file 입력된 후 Project Navigator

오른쪽 부분의 HDL source 창에 있는 jcounter.vhd 파일의 코드를 완성하여야 합니다. 파일의 Architecture 부분에 User application 로직을 설계합니다. .

< jcounter.vhd 완성 코드>

```

library IEEE;
use IEEE.std_logic_1164.all;

entity jcounter is
    port (
        clk      : in STD_LOGIC;
        Q        : out STD_LOGIC_VECTOR (7 downto 0)
    );

```

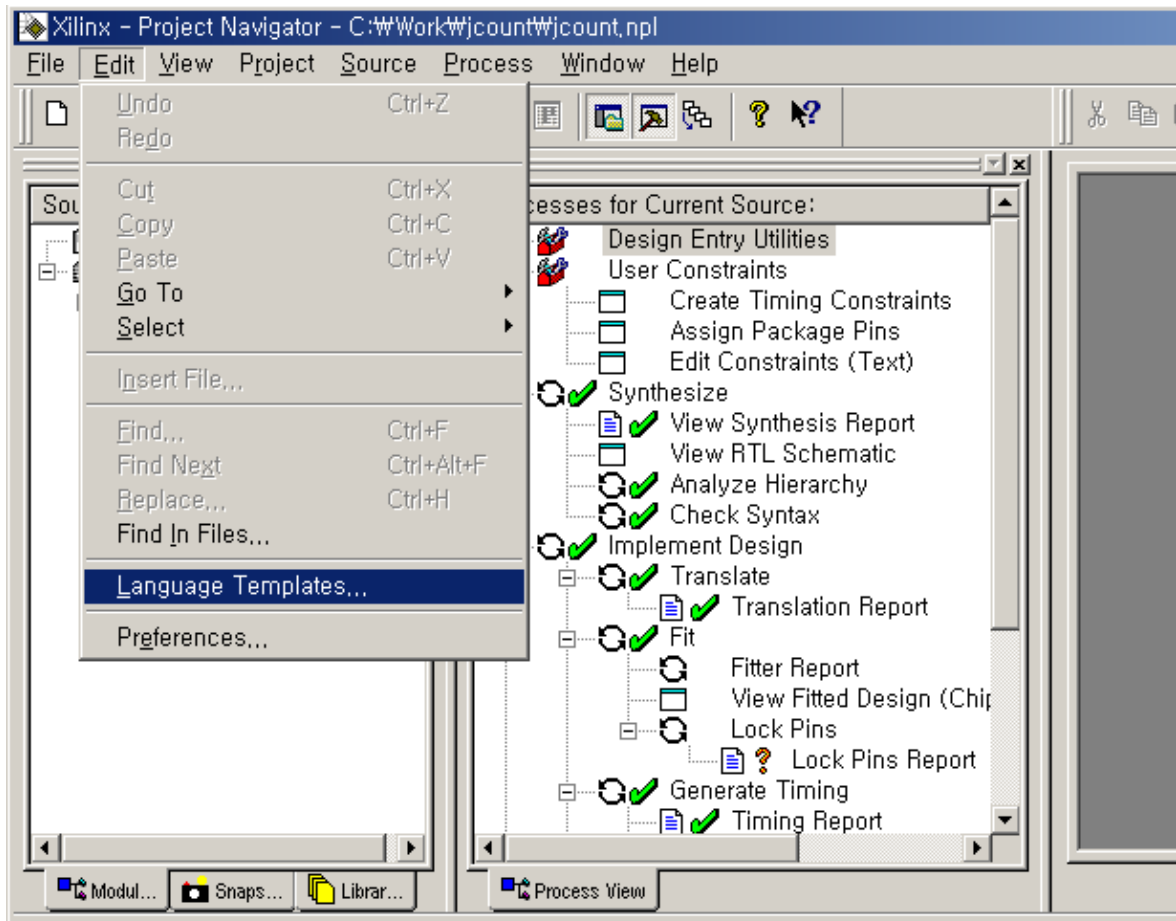
```
end jcounter;
```

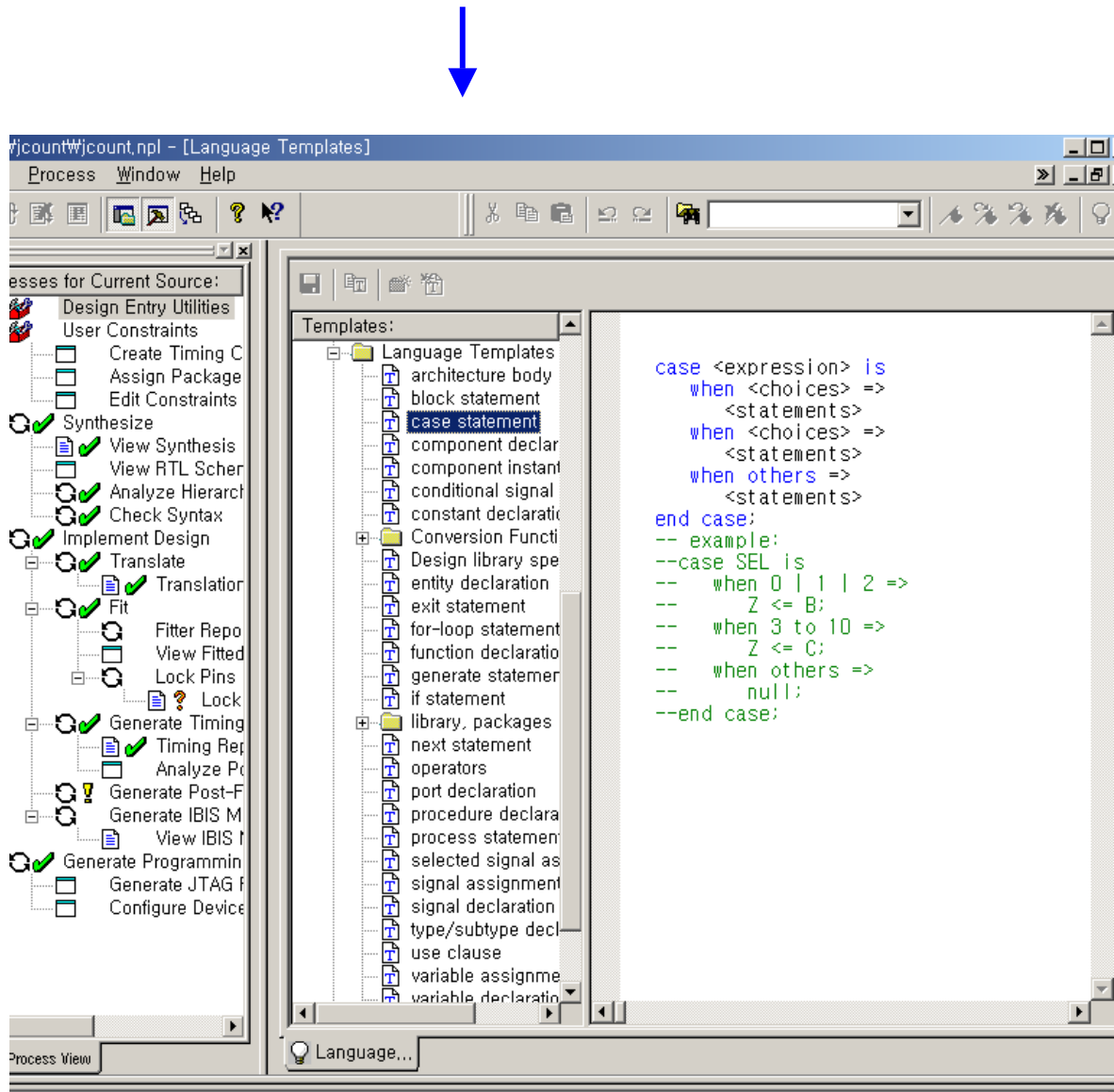
```
architecture jcounter_arch of jcounter is  
  signal Dout : std_logic_vector( 7 downto 0);  
begin
```

```
  process (CLK)  
  begin  
    if CLK'event and CLK='1' then  
      Dout(7 downto 1) <= Dout(6 downto 0);  
      Dout (0) <= not Dout(7);  
    end if;  
  end process;  
  Q <= Dout ;  
end jcounter_arch;
```

### 3.3 Language Template 사용하기

Project Navigator는 VHDL ( 혹은 Verilog HDL) 사용자를 위한 기본적인 Language Templates를 제공합니다.



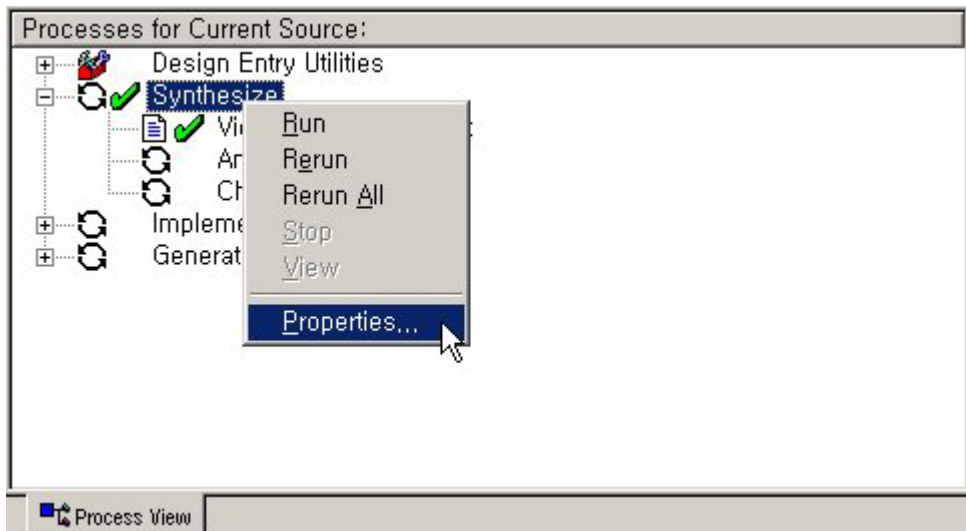


기본적인 HLD Standard 문법을 Language Templates를 이용하여 쉽게 설계를 할 수 있습니다.

## 4. Synthesize 하기

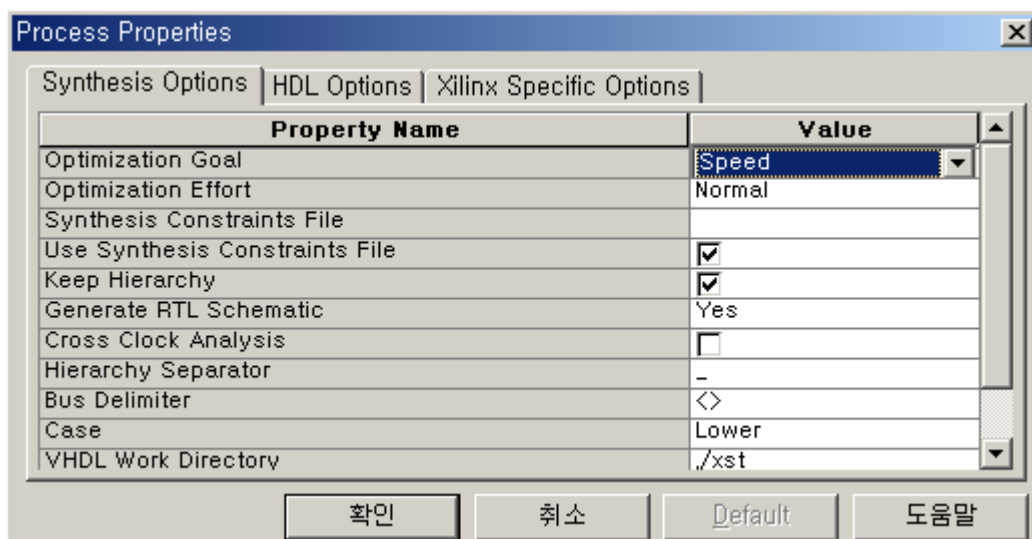
Webpack ISE에서 사용되는 Synthesis Tool은 XST(Xilinx Synthesis Technology) 입니다. XST 는 HDL design을 synthesize하고 .NGC 확장자를 가진 Netlist 파일을 생성합니다.

아래의 그림과 같이 Synthesize를 하기 위하여 Processes for Current Source 창의 Synthesize를 더블 클릭합니다. Synthesize Default Option을 변경하기 위한 방법은 다음과 같습니다. Synthesize에 마우스의 **오른쪽 버튼을 클릭**하고 Properties..를 선택하면....



< 그림 12 > Processes for Current Source창에서 Properties 선택

Process Properties가 뜹니다.



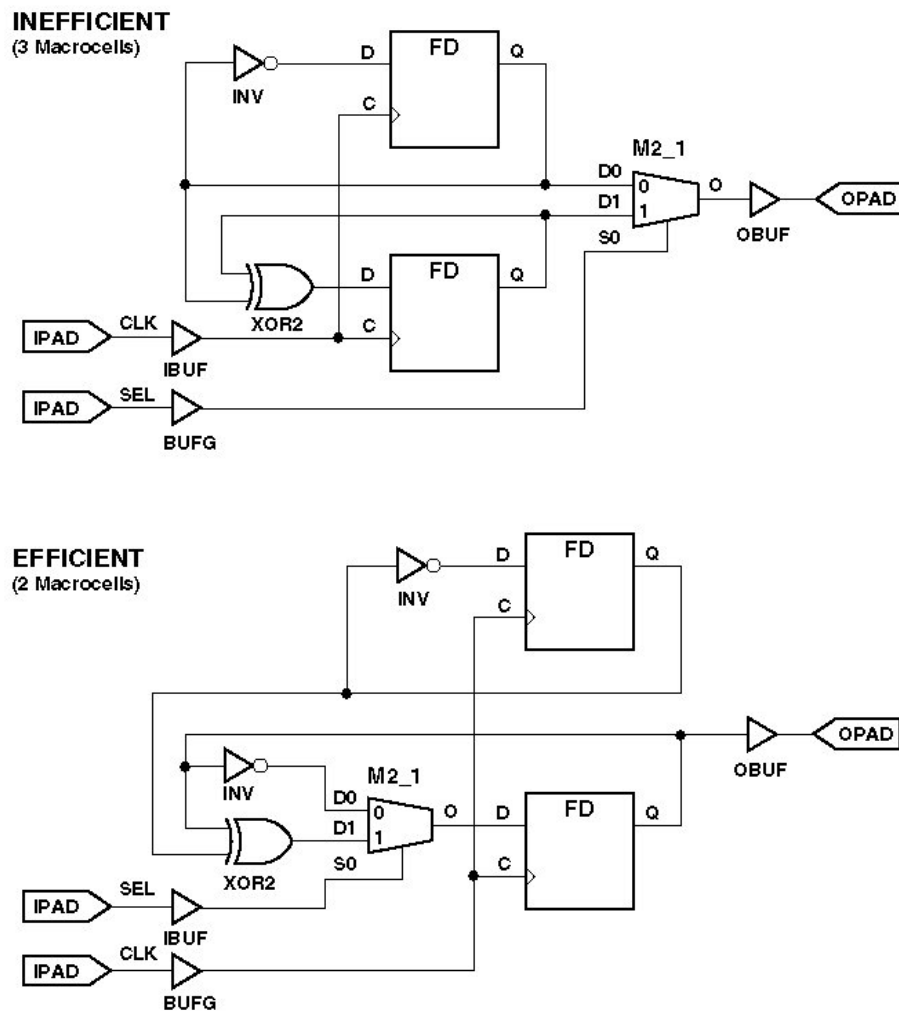
< 그림 13 > Synthesis Option Properties

Synthesis Options의 내용은...

- Optimization Goal 은 두 가지 value ( Speed, Area)가 있는데 로직레벨을 축소하면서 speed위주로 최적화하는 방법으로서 예를 들어 CPLD의 경우 3개의 macrocell사용하는 것을 2개만 사용하여 signal이 거치는 단계를 줄이는 방법이 있고 또 다른 방법으로 Area위주로 최적화하는 경우도 있습니다. .

<참조> 로직레벨을 축소한다는 의미는 아래의 그림을 참조하시면 됩니다.

Figure 3.8 Reducing Levels of Logic



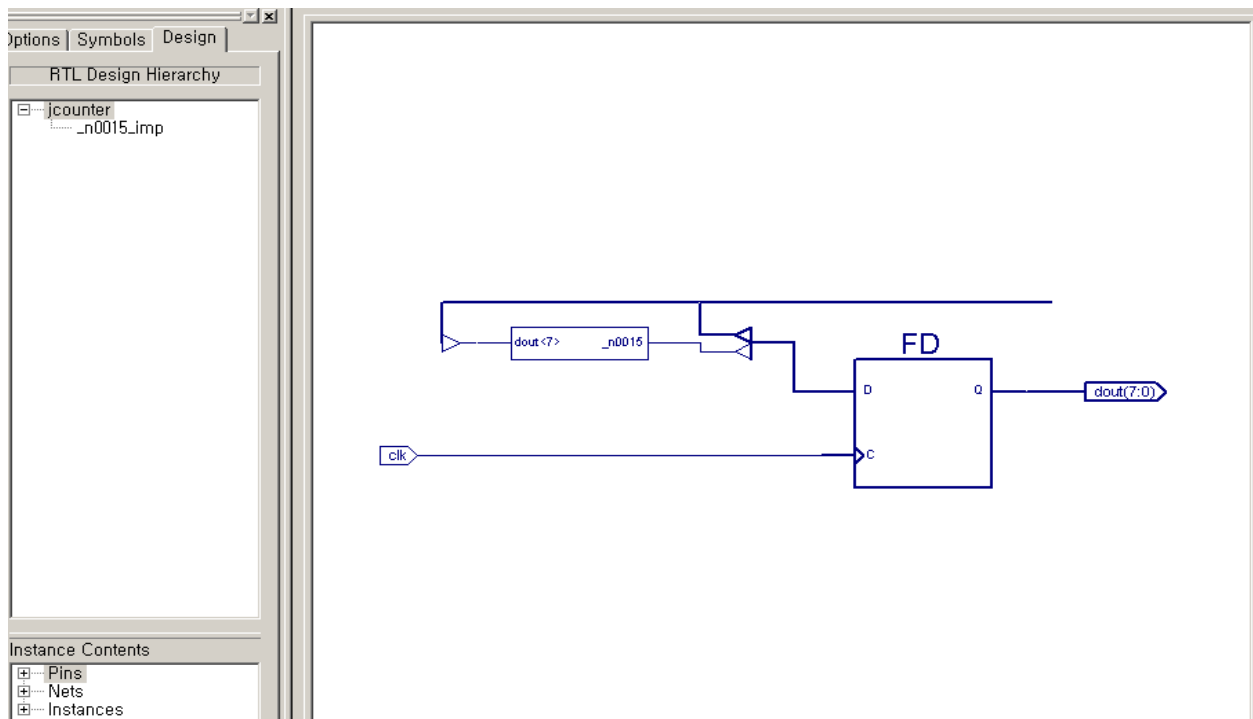
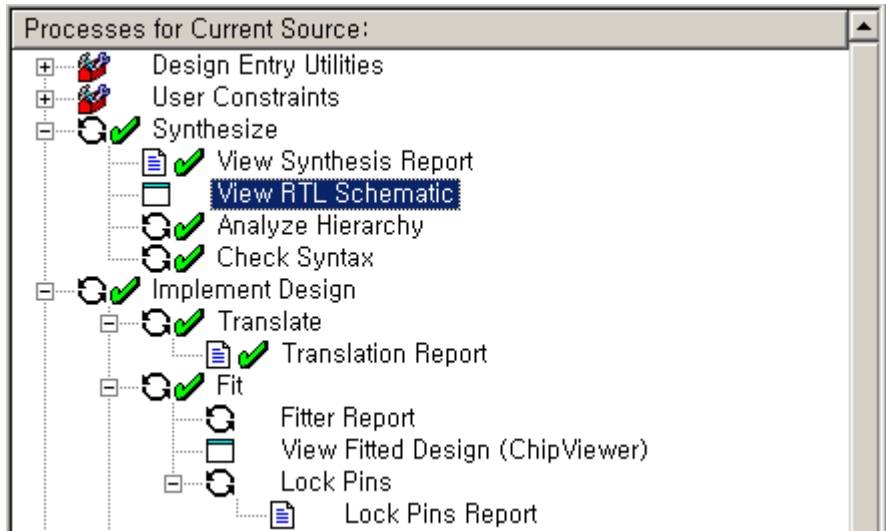
- Optimization Effort 는 synthesis optimization의 수준을 나타내며 default 로 Normal 이지만 high로 놓게 되면 multiple optimization 알고리즘을 이용하여 최적하지만 많은 Compile 시간이 걸립니다.

그 밖의 Properties 들은 Process Properties 의 도움말을 참조하여 대개의 경우는 Default 값으로 사용하는 것이 바람직합니다.

## 4.1 View RTL Schematic

HDL로 작성된 디자인을 쉽게 이해하기 위해 “View RTL Schematic”을 이용합니다.

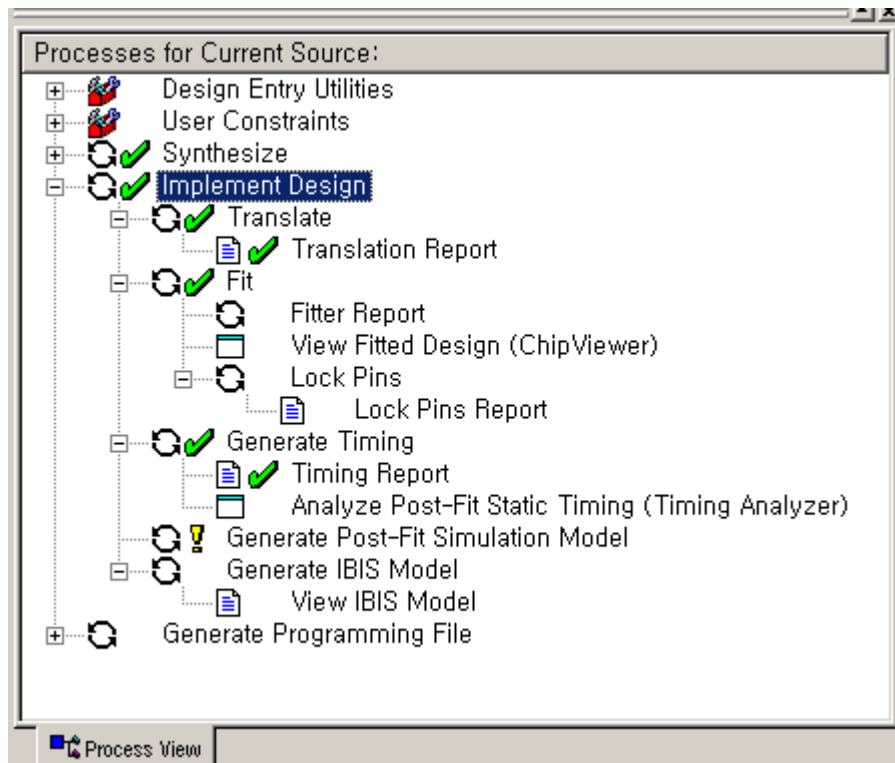
전체 Top 디자인의 Block-diagram과 내부의 세부적인 Sub-diagram을 더블클릭 함으로서 자세히 볼 수 있습니다.



< 그림 14 > View RTL Schematic

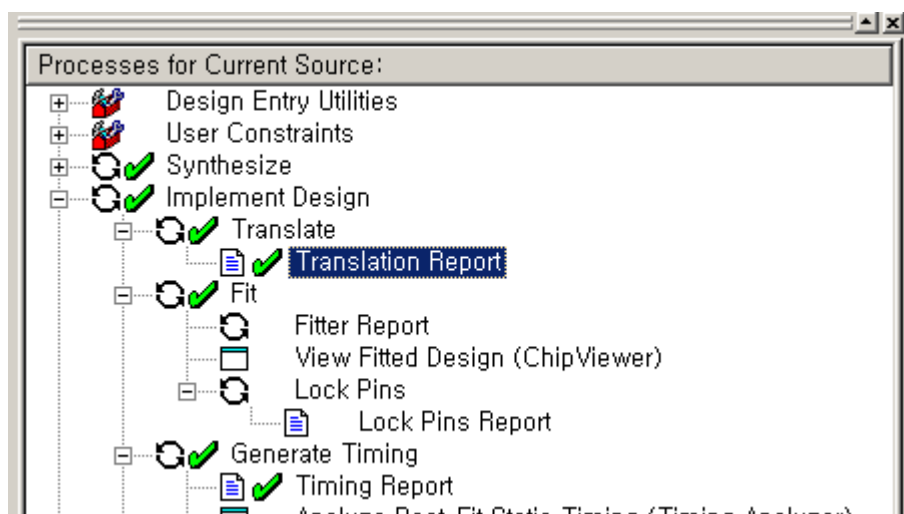
## 5. Implementation 하기

CPLD 과 FPGA은 약간 다른 Implementation을 수행합니다. 이 장에서는 CPLD의 경우만 설명합니다. CPLD의 Implementation은 Translate와 Fit, Generate Timing으로 나눌 수 있습니다. Translate 는 XST로 synthesis 해서 생성된 Netlist 인 .NGC파일과 User가 작성한 UCF파일을 가지고 .NGD 파일을 생성



< 그림 15 > Implement Design 선택

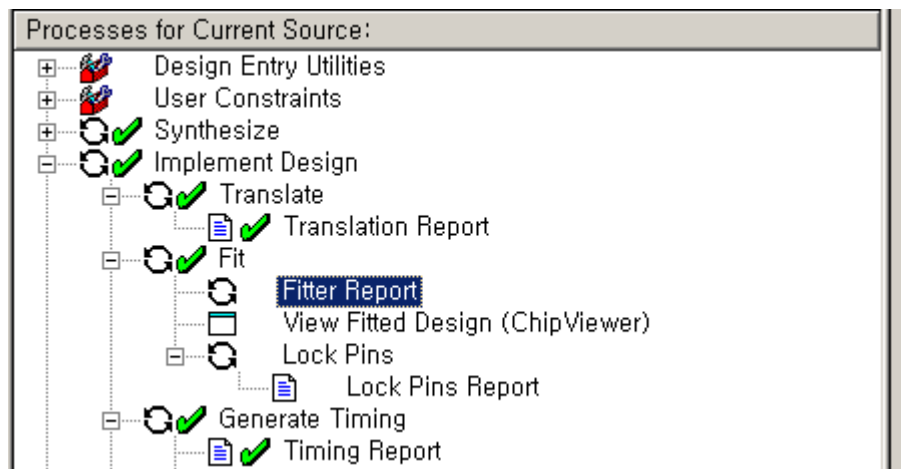
이 과정을 수행하면서 발생한 Error 및 Warning 등의 Report를 해주는 파일이 NGDbuild Log파일인 .bld파일 안에 들어있고 이 .ngd파일은 Translation Report를 더블클릭하면 아래와 같이 볼 수 있습니다.



<그림 16> Translation Report 선택



위와 같이 Translate를 한 후에 선택한 CPLD device구조에 맞게 Fitting작업합니다.  
 이러한 작업을 하는 것이 Fitter인데 그 결과도 Fitter Report를 더블 클릭함으로써 내용을 확인할 수 있습니다.



<그림 16> Fitter Report 선택

Fitter Report에는 사용된 Device 정보를 포함해서 사용된 Macro cell의 수 , 사용된 i/o ,실제로 Signal이 mapping되어서 fitting등에 관련된 내용들이 들어있습니다. 이 Report는 설계자의 의도대로 설계가 제대로 되었는지 확인할 수 있는 중요한 Report입니다.

```

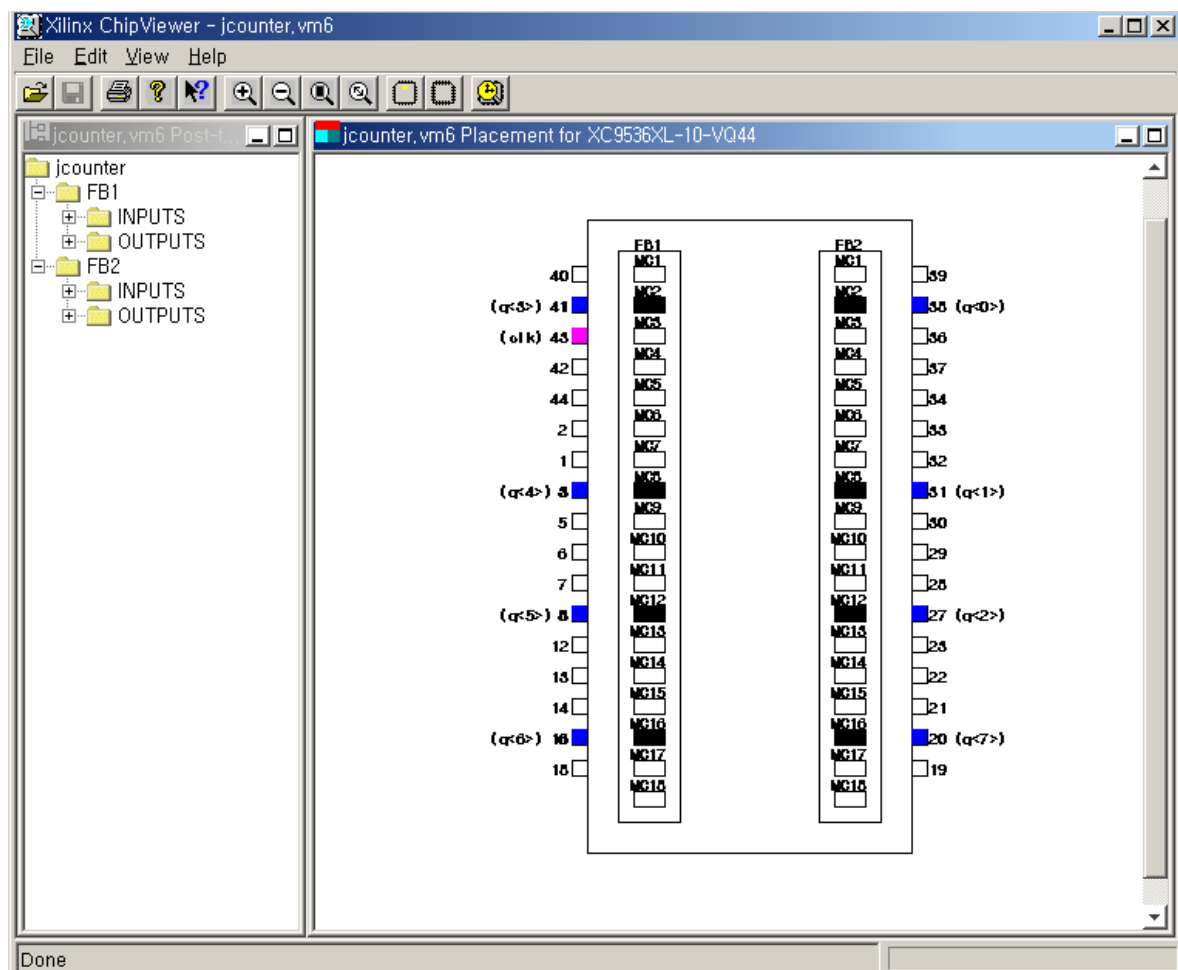
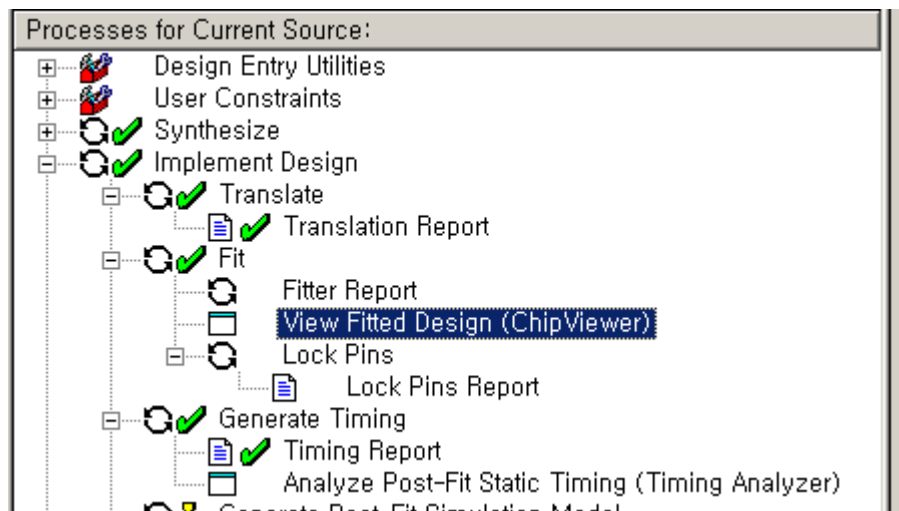
1 cpldfit: version F.26
2
3
4 Design Name: jcounter
5 Device Used: XC9536XL-10-VQ44
6 Fitting Status: Successful
7
8 ***** Resource Summary *****
9
10 Macrocells      Product Terms    Registers      Pins          Function Block
11 Used           Used             Used           Used           Inputs Used
12 8 /36 ( 22%)  8 /180 ( 4%)  8 /36 ( 22%)  9 /34 ( 26%)  8 /108 ( 7%)
13
14 PIN RESOURCES:
15
16 Signal Type      Required    Mapped    | Pin Type          Used    Remaining
17 -----
18 Input           : 0         0         | I/O                : 8      20
19 Output          : 8         8         | GCK/I/O            : 1      2
20 Bidirectional   : 0         0         | GTS/I/O            : 0      2
21 GCK              : 1         1         | GSR/I/O            : 0      1
22 GTS              : 0         0
23 GSR              : 0         0
24
25 Total           9          9
26

```

< 그림 17> Fitter Report의 내용

## 5.1 ChipViewer

실제로 Fitting된 signal들의 연결정보를 graphic하게 보는 것이 Chipviewer인데 ChipViewer를 더블클릭하면 아래와 같은 ChipViewer가 뜹니다.

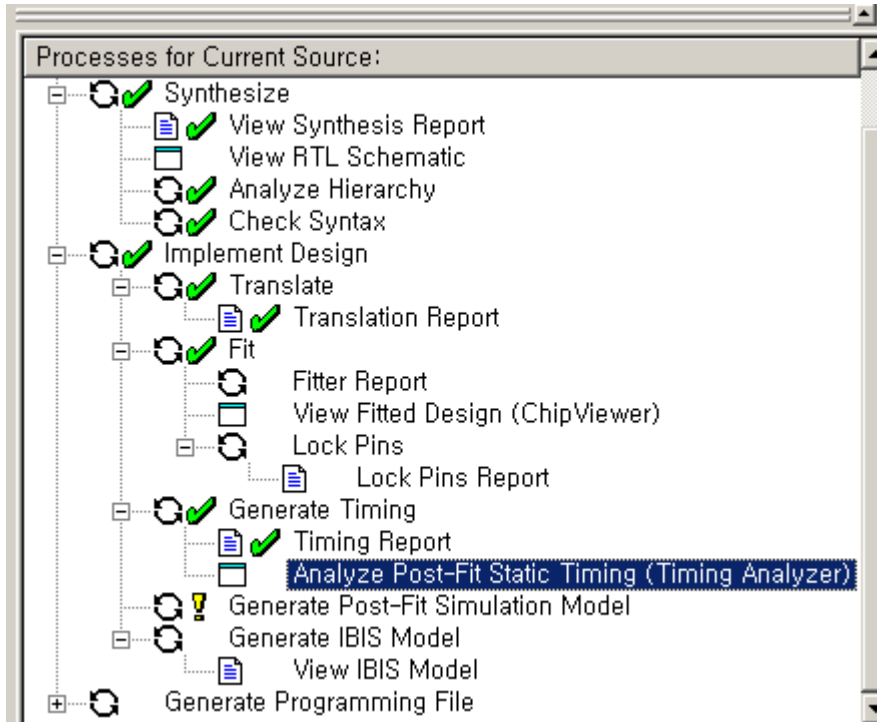


< 그림 18 > ChipViewer 선택

원하는 신호의 fitting Path를 보고 싶으면 원하는 PAD나 MC를 클릭하면 됩니다.

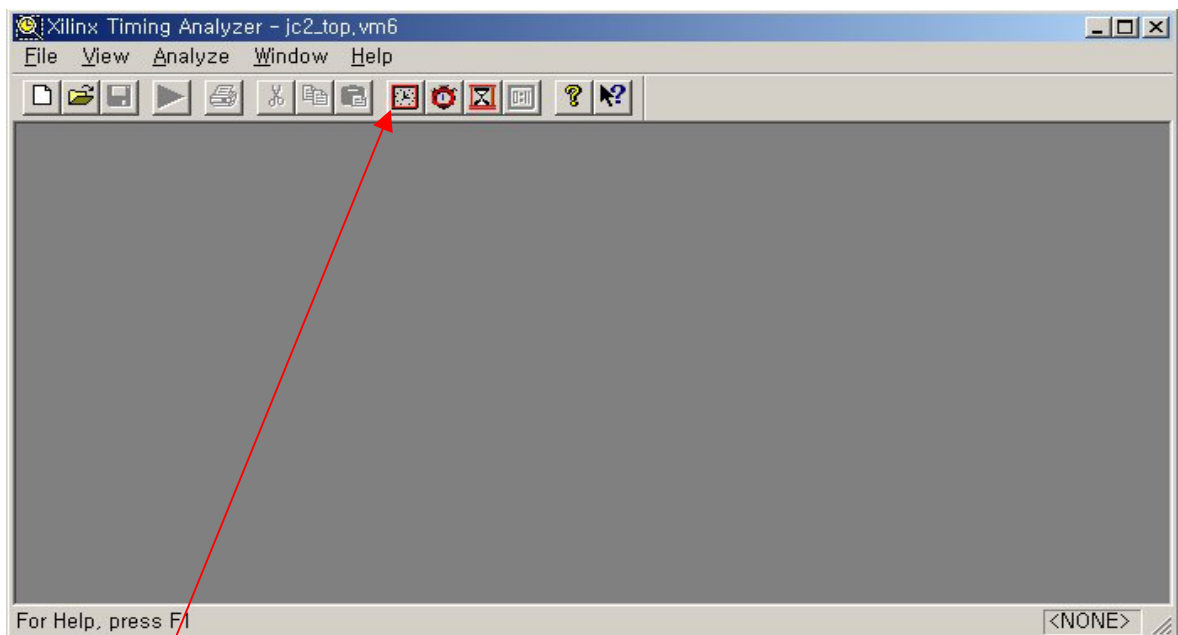
## 5.2 Generate Timing

지금까지 설계한 디자인의 Timing 결과를 확인하는 단계입니다. 이것은 아래와 같이 Timing report를 이용합니다. 선택된 CPLD device에 맞게 Fitting된 Signal들의 Timing정보를 보는 것입니다.



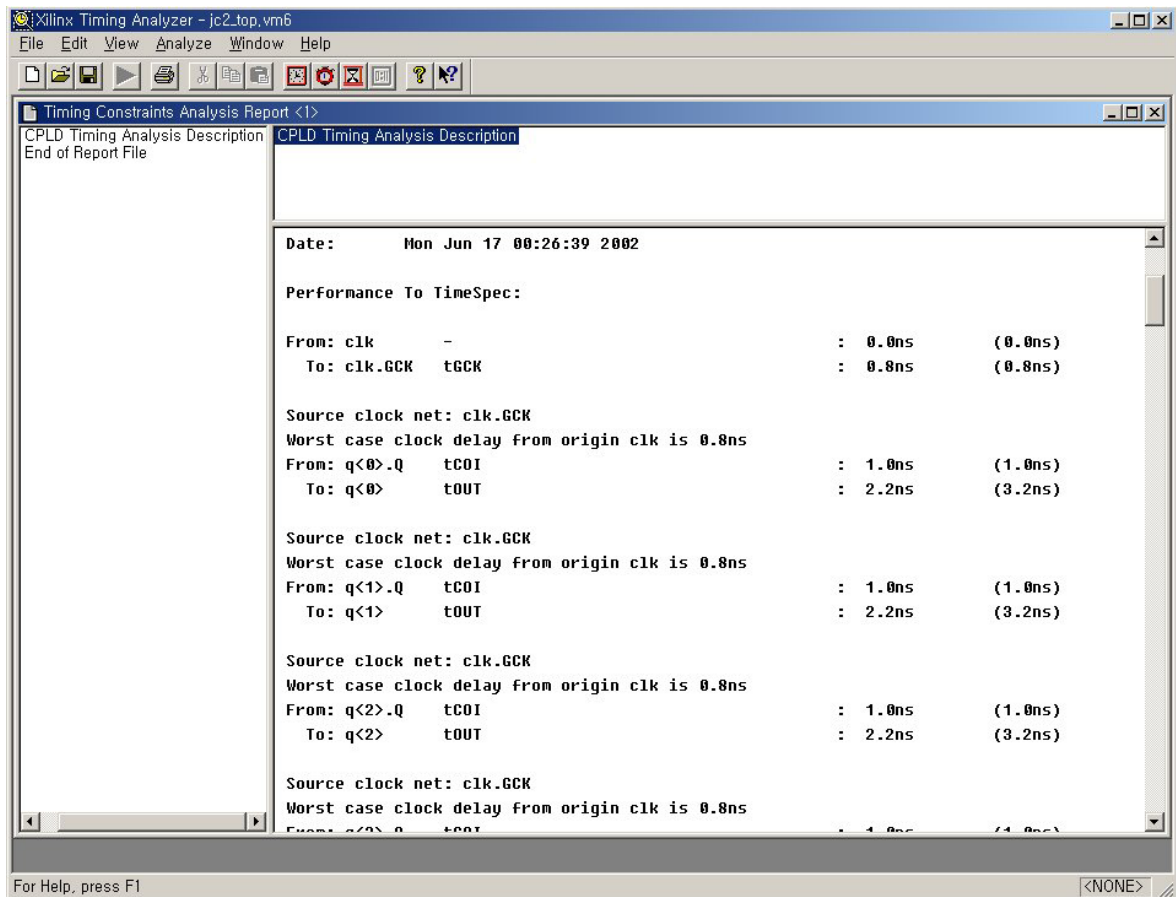
<그림 19> Timing Analyzer

Timing Analyzer를 통해 좀 더 면밀한 Timing정보로 디자인을 분석할 수 있습니다.



<그림 20> Timing Analyzer 창

“Analyze against Timing Constraints”아이콘을 클릭함으로 아래와 같이 Timing Constraint에 대해 실제로 fitting신호들의 시간정보를 볼 수 있습니다.



< 그림 21 > Timing Analyzer 창에서 Timing Constraints Analysis Report

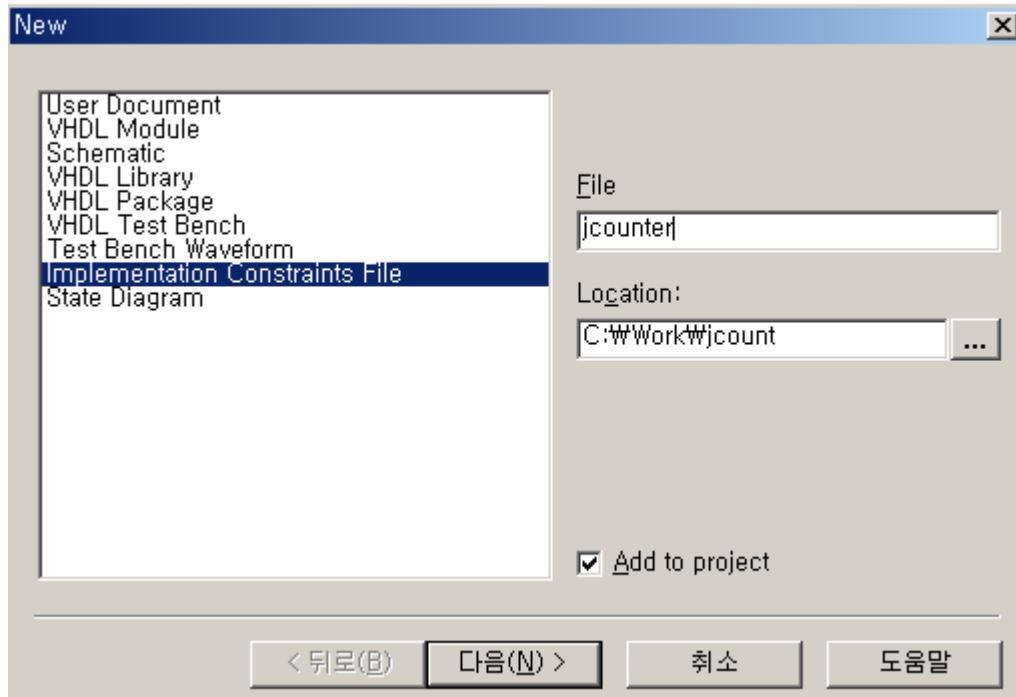
이렇게 Implement를 마치고 정상적이면 녹색의 체크표시가 되고 설계자가 원하는 Timing Option에 만족하지 못하여 Error가 발생하면 빨간색으로 표시됩니다.

## 6. User Constraints ( Pin Assignment )

이 장에서는 지금까지 Auto Assignment 된 project의 여러 In/Out pin들을 설계자가 의도하는 Device의 Pin에 직접 설정하는 방법을 설명합니다. 그 방법을 여러 가지가 있습니다. 먼저 HDL source내에 Attribute 명령문을 이용하는 방법과, UCF 파일 내에 설정하는 방법이 있습니다. 아래에서는 UCF 파일을 이용하는 방법을 설명합니다.

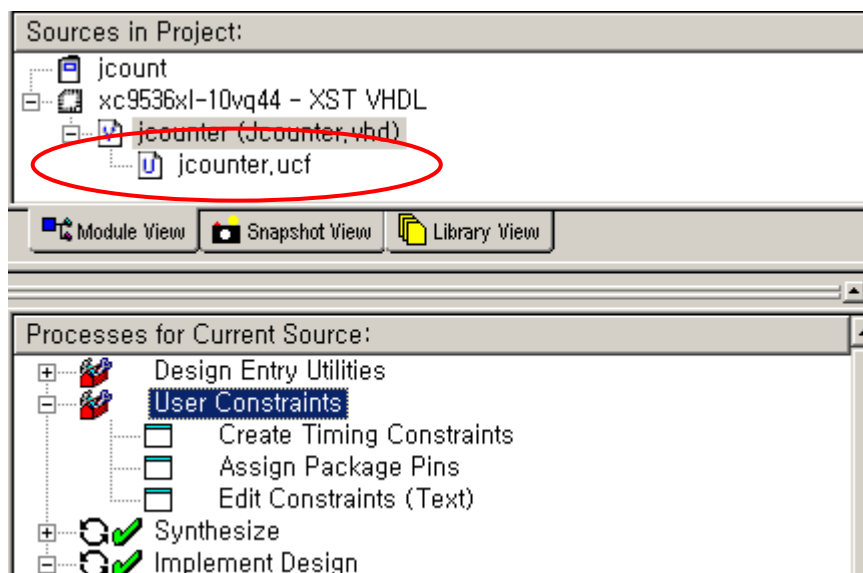
## 6.1 UCF 파일 만들기

먼저 **Project Navigator -> Project -> New Source...** 를 선택하고 아래와 같이 작성한 하면,



< 그림 22 > Implementation Constraints File 선택

아래와 같이 jcounter.ucf 파일이 **TOP 파일 밑에** 생성됩니다. 이 UCF 파일 안에는 아무런 정보가 없는 빈 파일입니다. 이 파일에 Pin Assignment 정보를 포함한 각종 Timing Option을 작성합니다. 이번 장에서는 Pin Assignment를 하는 방법만 설명하겠습니다.

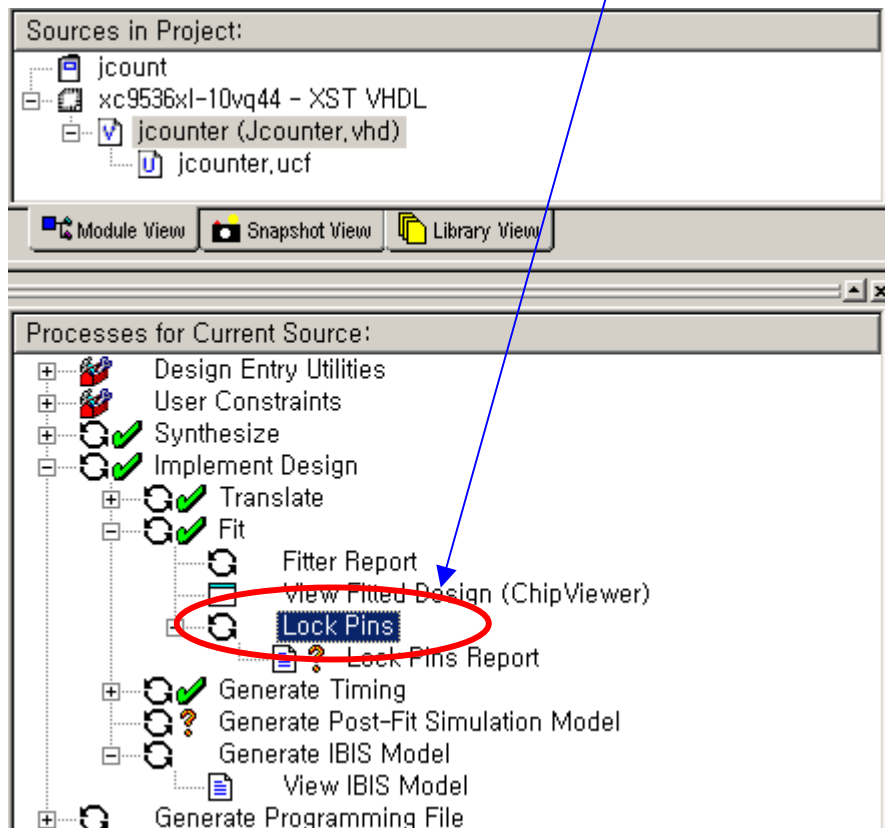


< 그림 23 > User Constraints

생성된 UCF 파일에 Pin위의 그림과 같이 Pin Assignment를 하는 방법은 여러 가지가 있다. “Edit Constraints(Text), Assign Package Pins, Create Constraints Editor를 이용하는 방법이 있습니다.

## 6.2 “Lock Pins” 이용하기

이 방법은 Implementation시 자동으로 만들어지는 Fitter Report의 Pin Assign 정보를 이용하는 것입니다. 아래의 그림과 같이 Lock Pins을 실행시키면,



<그림 24> Lock Pins 선택

Working Directory에 다음과 같은 내용을 포함한 “Jcounter\_lock.ucf”라는 파일이 생깁니다. 그 파일을 열어보면,

```
#PINLOCK_BEGIN
#Mon Mar 03 14:52:37 2003
NET "clk"          LOC = "S:PIN43";
NET "q<0>"         LOC = "S:PIN38";
NET "q<1>"         LOC = "S:PIN31";
NET "q<2>"         LOC = "S:PIN27";
NET "q<3>"         LOC = "S:PIN41";
NET "q<4>"         LOC = "S:PIN3";
NET "q<5>"         LOC = "S:PIN8";
NET "q<6>"         LOC = "S:PIN16";
NET "q<7>"         LOC = "S:PIN20";
#PINLOCK_END
```

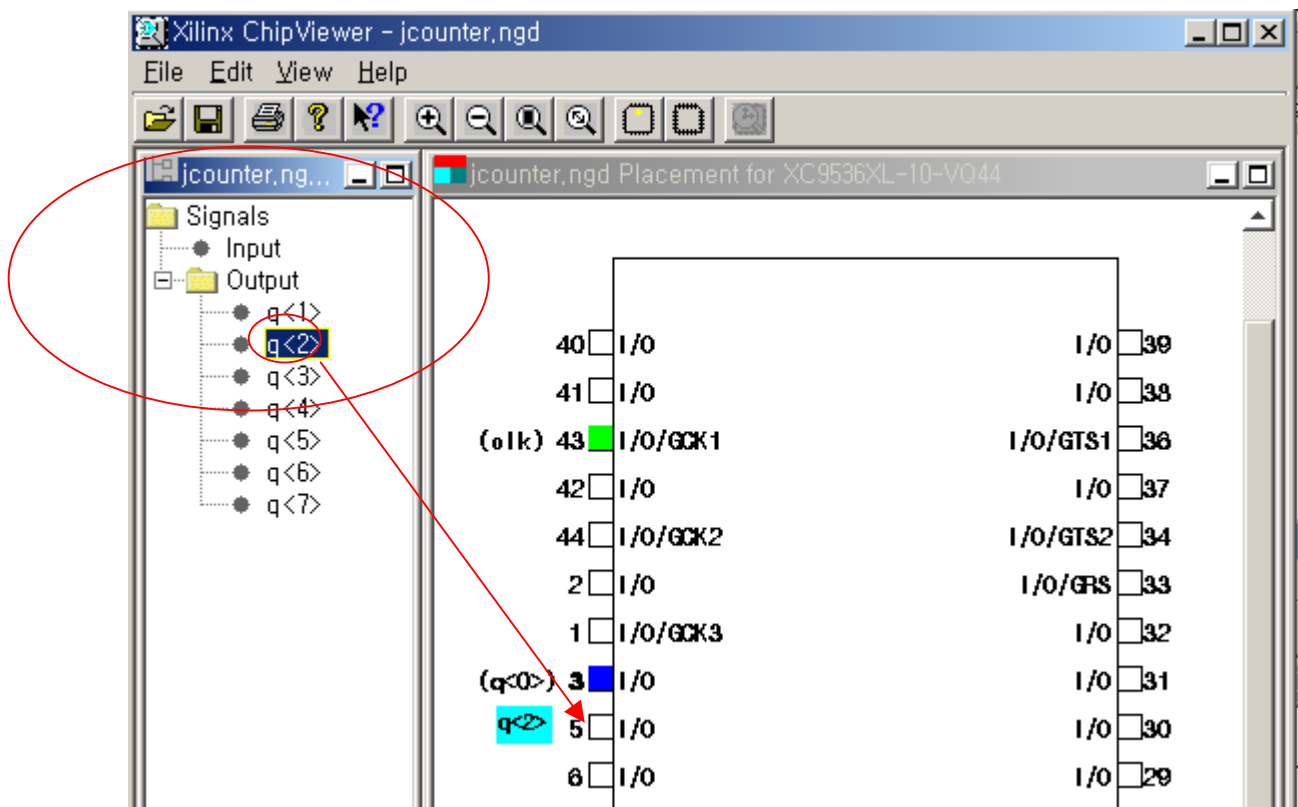
이 Pin Assign 정보를 변경하고자 하는 핀 번호만 수정한 후 저장합니다.

## 6.3 Edit Constraints ( Text )

Edit Constraints를 실행하면 jcounter.ucf 파일의 Text editor가 뜨는데 그곳에 규정에 맞는 내용을 직접 작성합니다.

## 6.4 Assign Package Pins

ChipViewer를 이용하는 방법으로서 아래의 그림과 같이 In/Out Pin들 선택하고 원하는 위치의 Pad에 Drag and Drop 하면 Pin Assign을 마치게 됩니다.

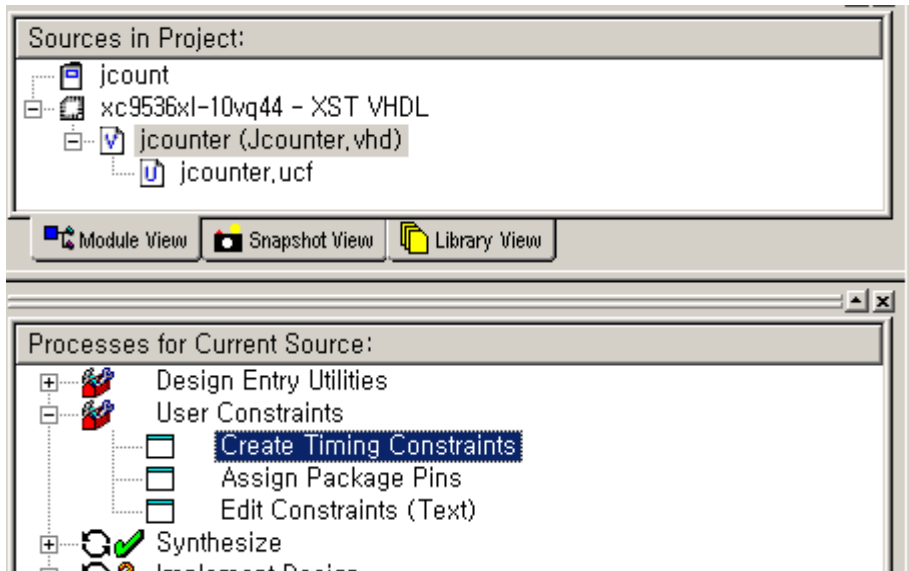


< 그림 25 > ChipViewer for Pin Assign

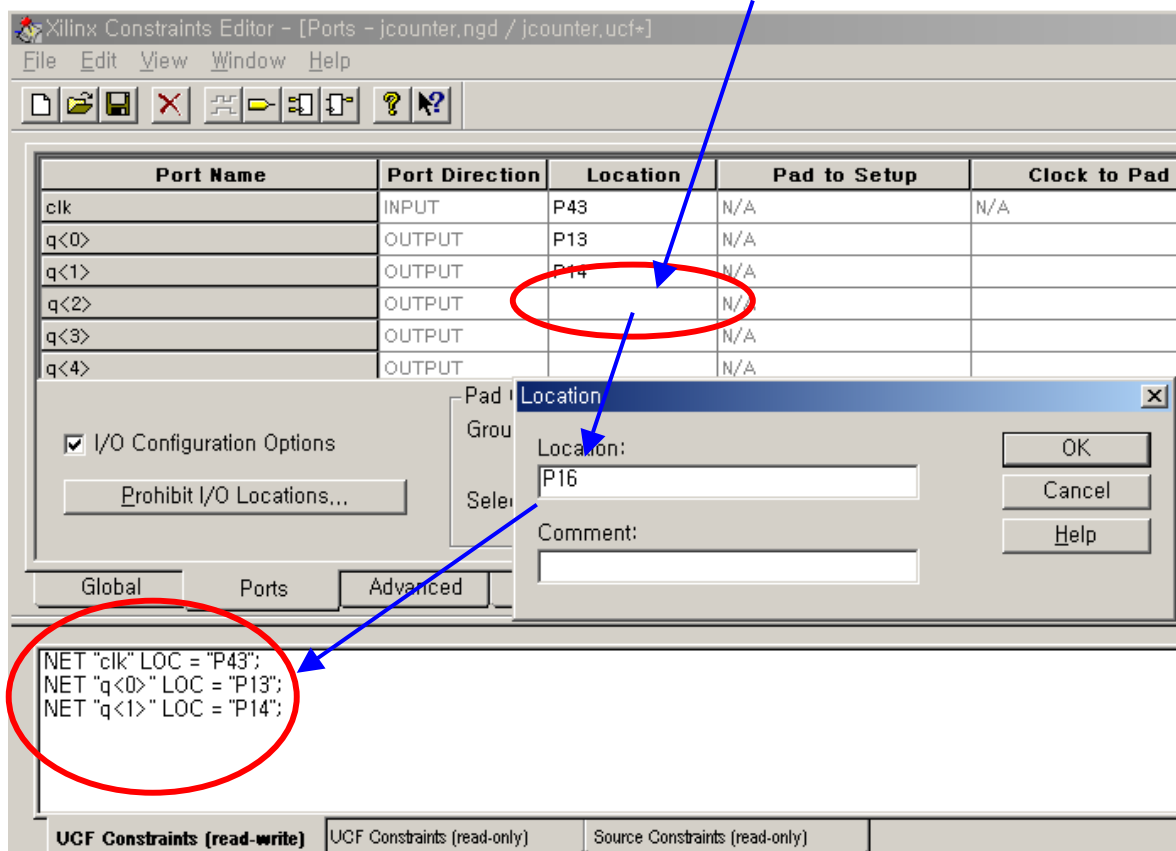
모든 In/Out pin의 설정이 끝난 후, 저장을 하여 Pin Assign 정보를 Ucf 파일에 저장한다.

## 6.5 Create Timing Constraints

Constraints Editor 이용하여 Pin Assignment를 하는 방법입니다. 아래와 같이 Create Timing Constraints를 실행합니다.



해당 핀의 란을 선택 & 핀 정보 입력

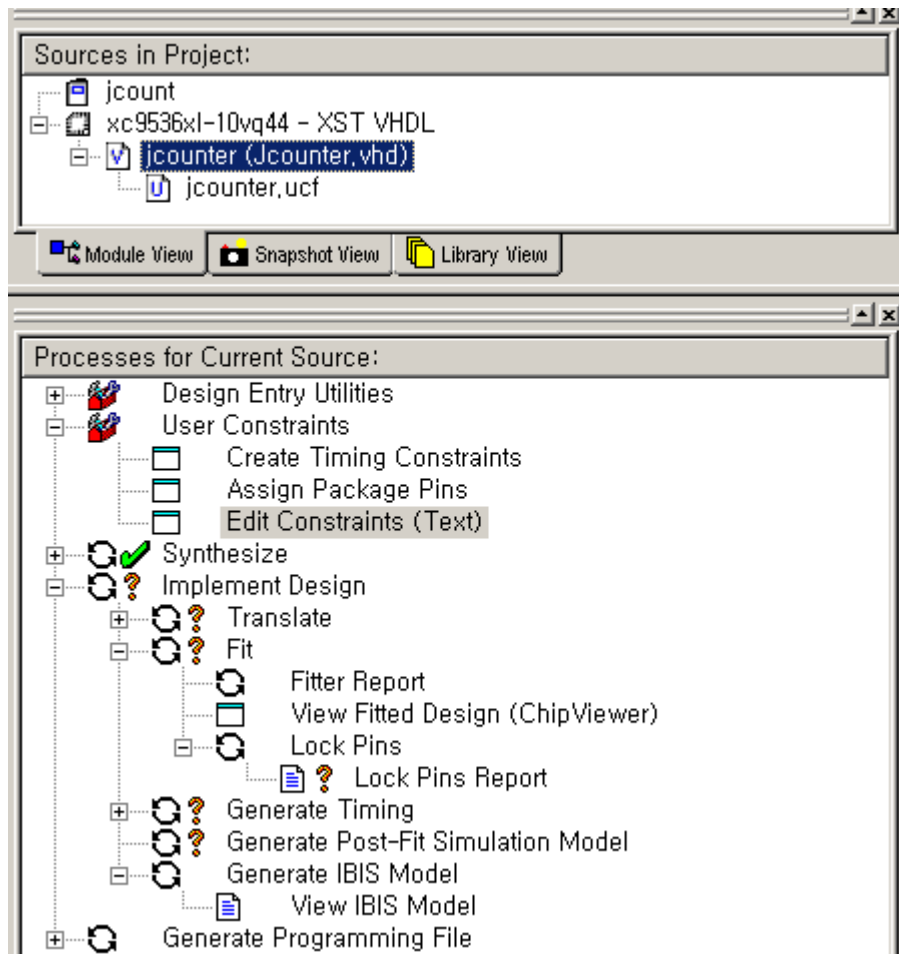


< 그림 26 > Constraints Editor

위의 그림과 같이 직접 핀 정보를 해당 란에 직접 입력하고 저장하면, UCF 파일에 자동으로 저장됩니다.



Pin Assign 정보나 Timing Option을 수정한 후, Project Navigator는 아래와 같이 Implementation 상태는 물음표로 표시됩니다. 이것은 UCF 정보가 변경되어 Re-implementation을 해야 한다는 내용입니다.



< 그림 27> Re-Implementation

<참고> Pin Assign 은 직접 top-level의 코드에 직접 아래와 같이 삽입해도 된다.

이 방법은 jcounter.vhd 코드 안에서 아래와 같은 Attribute 명령문을 이용하여 pin\_assign을 할 수 있다.

```
attribute pin_assign : string;
attribute pin_assign of clk : signal is "13";
attribute pin_assign of q : signal is "14,16,18,19,20,21,22";
```

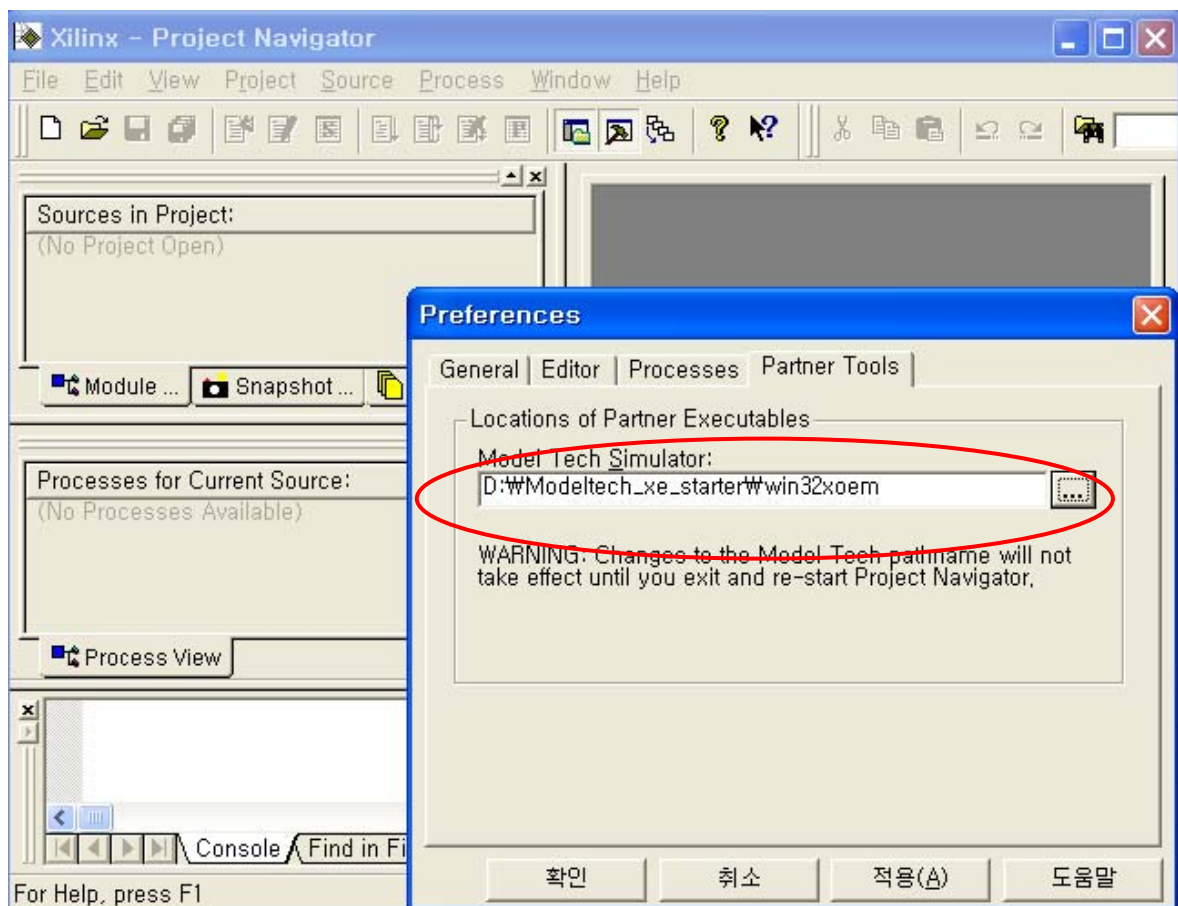
## 7. Modelsim Simulation

Xilinx에서 무료로 제공하는 Modelsim MXE starter version을 사용하기 위해서는 [MXE\\_5.6a\\_Full\\_installer.exe](#)를 설치합니다. Modelsim 사용에 필요한 License는 프로그램 설치 후 자동으로 Xilinx Home page에서 등록하여 License 파일을 받거나, Windows 시작메뉴 -> 프로그램 -> Modelsim XE v5.6a -> Submit License Request를 선택하여 받을 수 있습니다. License 파일은 메일을 통하여 즉시 받을 수 있습니다.

### 7.1 Modelsim MXE license

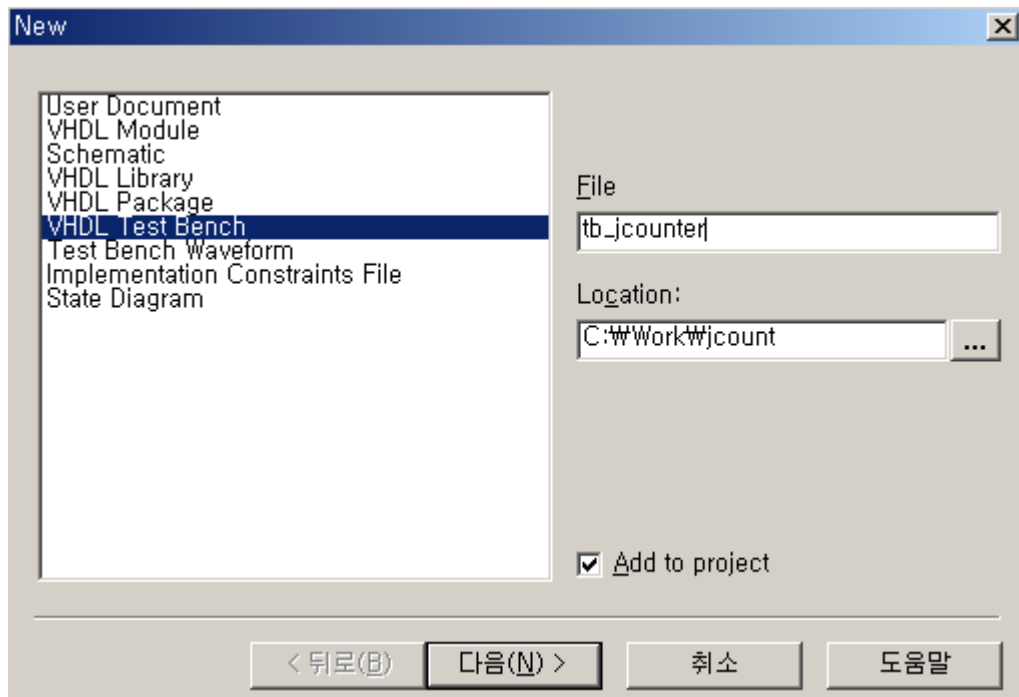
Project Navigator에서 Modelsim을 연결하기 위해서는 메뉴의 **Edit -> Preference**를 선택하고 Partner Tools에서 아래와 같이 Modelsim Xilinx Edition version 의 실행 파일의 디렉토리 경로가 정확히 설정되어있는지를 확인.

-> 아래의 그림은 Modelsim MX\_Starter version 을 사용할 경우를 설명함.

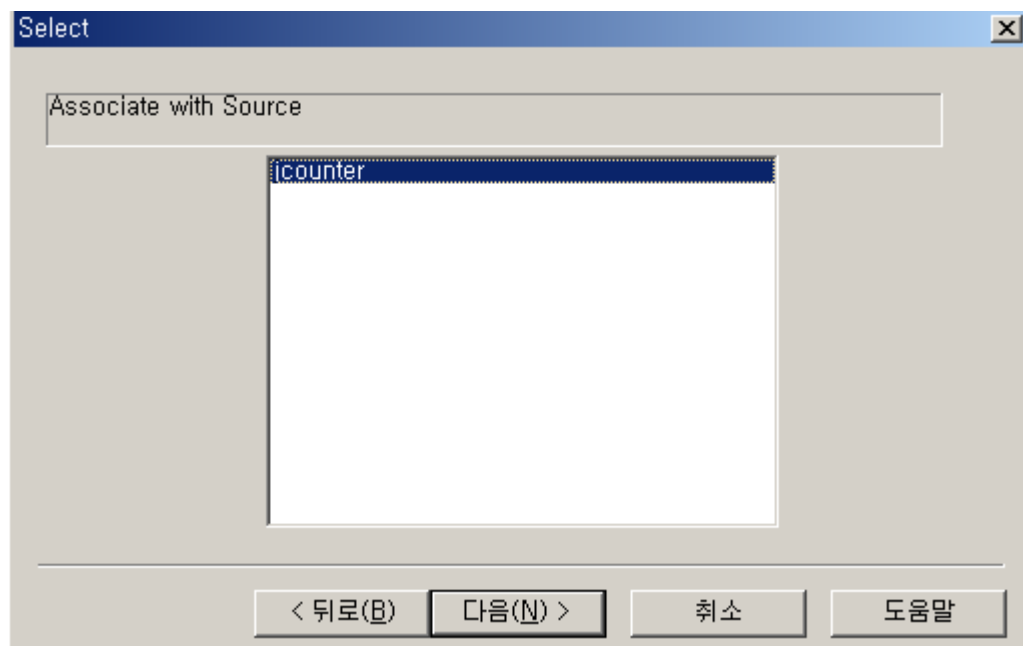


## 7.2 Making Test Bench for Modelsim

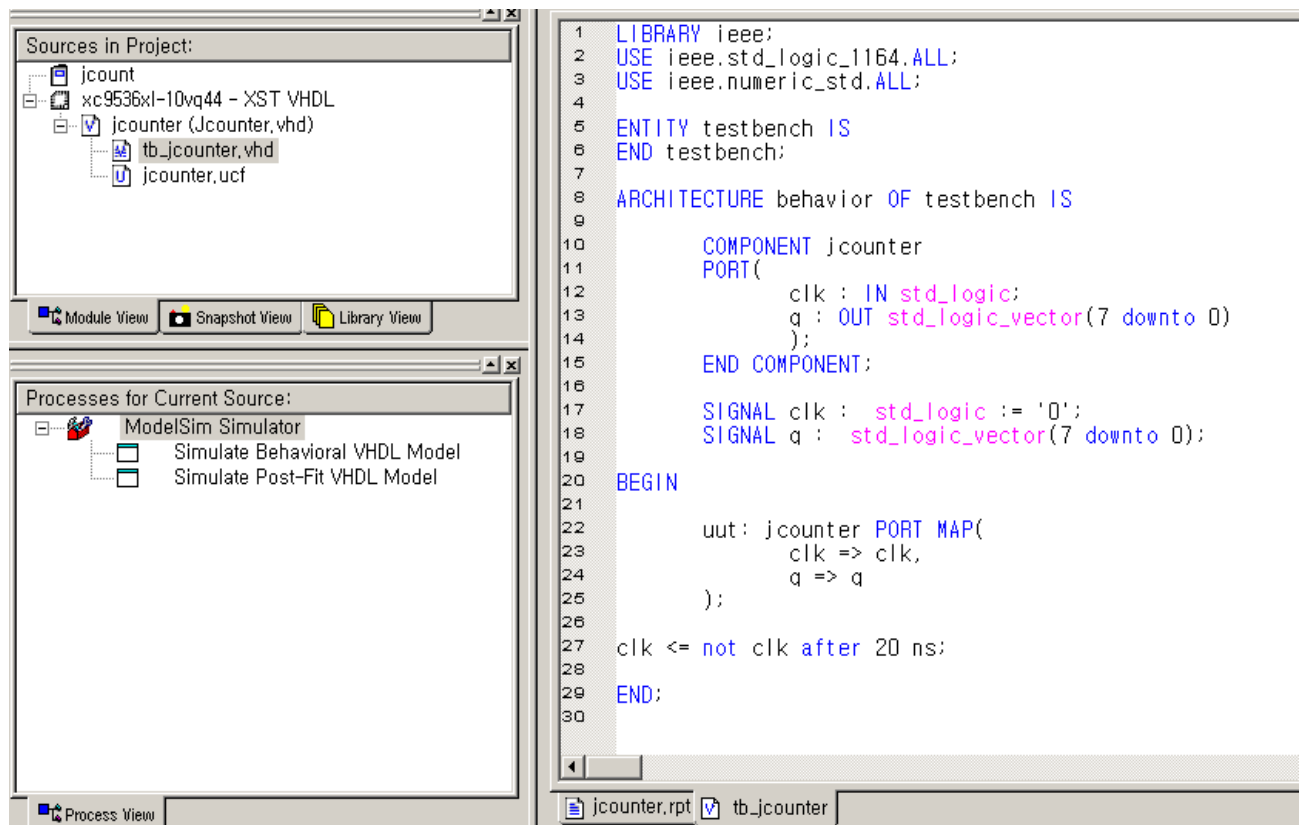
Project Navigator에서 **Project -> New source...** 를 선택한 후, 아래와 같이 “VHDL Test Bench”를 선택합니다. 보통 “tb\_top file name “로 설정합니다.



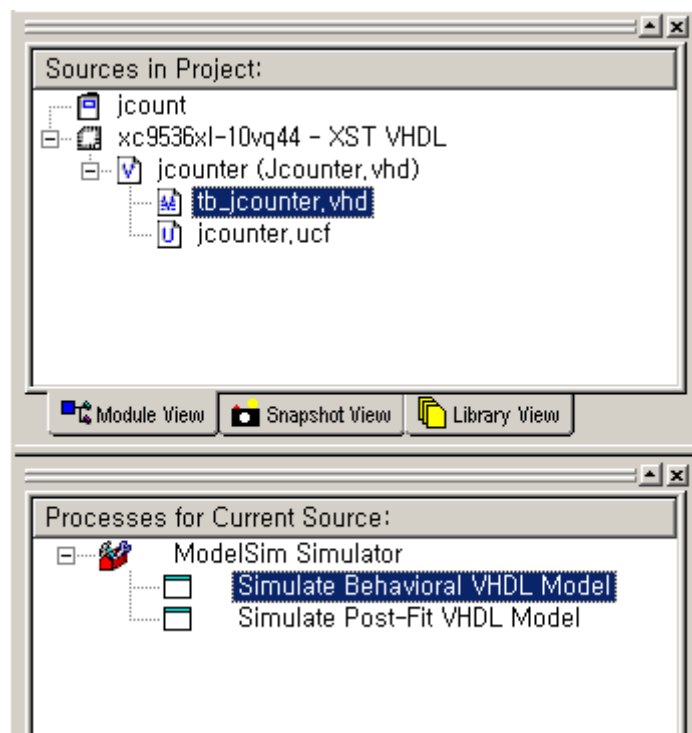
다음과 같은 Window가 나타나면, 마침 button을 Click합니다.  
만약 여러 개의 entity가 존재할 경우 TOP\_entity를 선택합니다.



Test Bench 파일의 초기 작성이 끝났으면, 아래와 같이 “tb\_jcounter.vhd” 파일을 완성합니다.

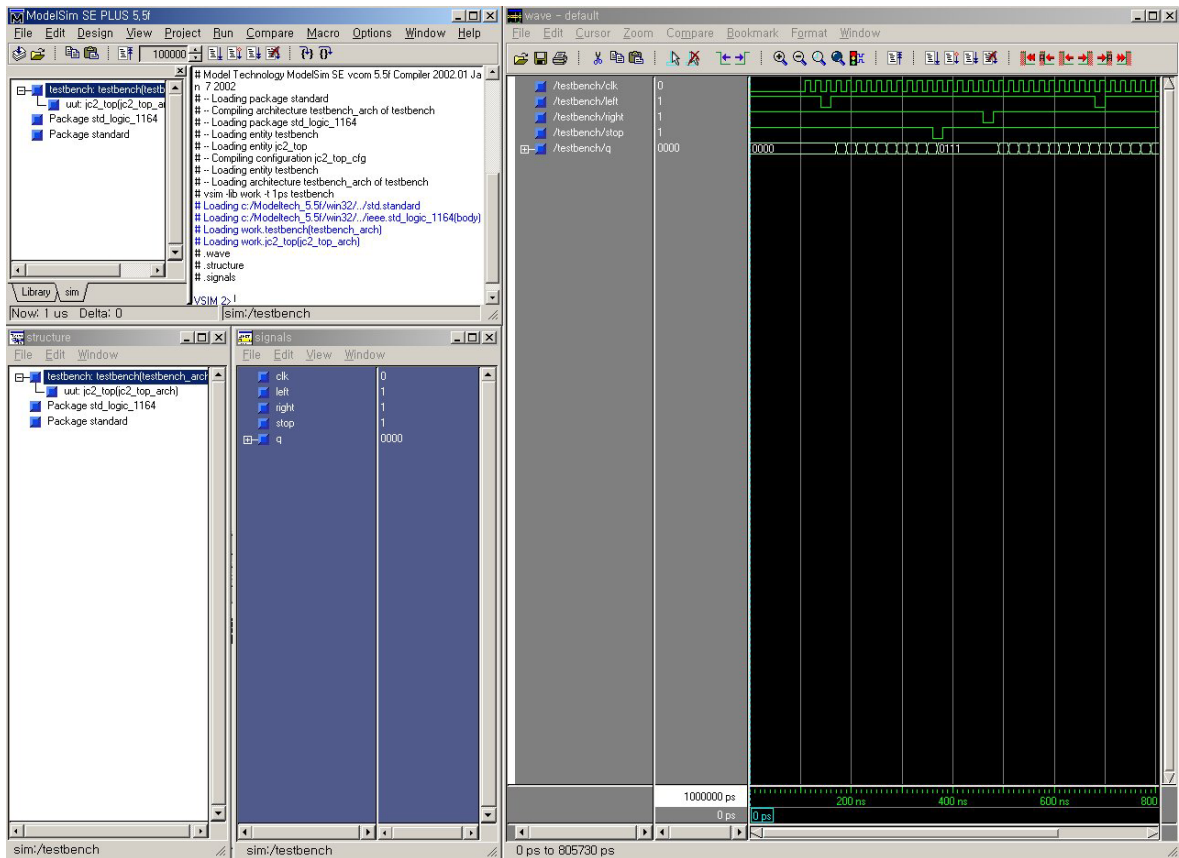


Project Navigator창에서 tb\_jcounter.vhd에 커서를 위치 시키면 Processes for Current Source창에 Modelsim simulator가 나타납니다.

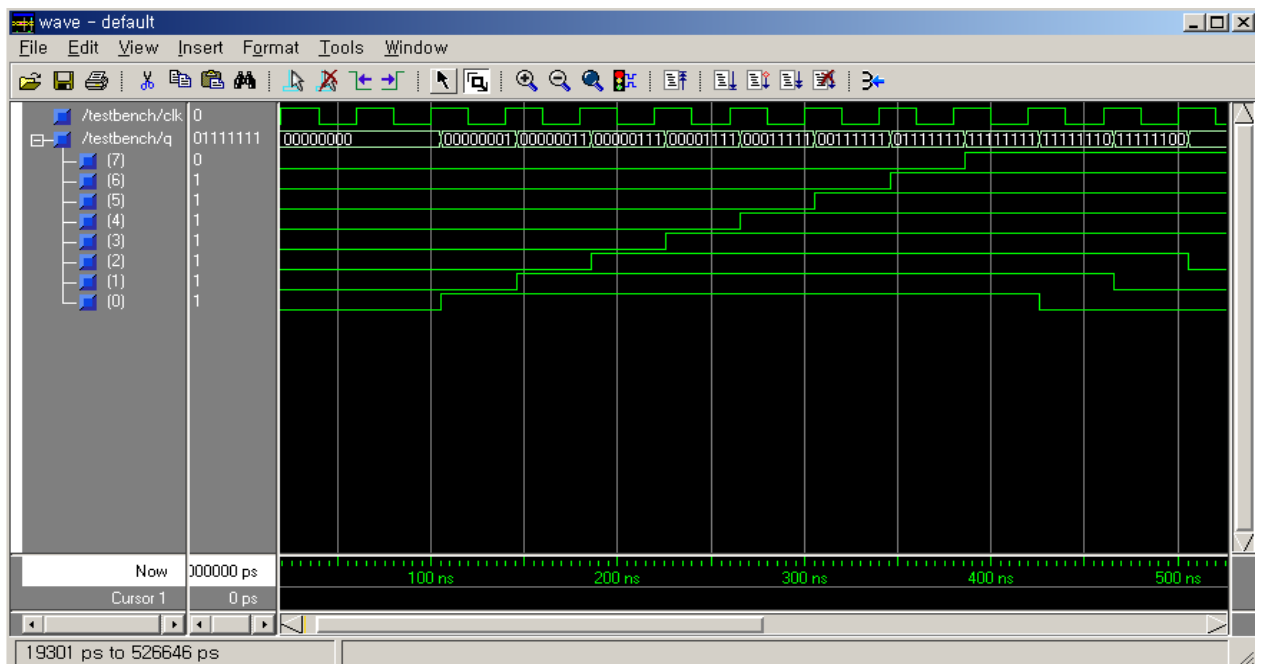


function 시뮬레이션을 하려면 Simulation Behavioral VHDL Model을 클릭합니다.

Timing 시뮬레이션을 하려면 Simulate Post-Fit VHDL Model를 클릭하면 아래와 같은 Modelsim 창이 뜹니다.

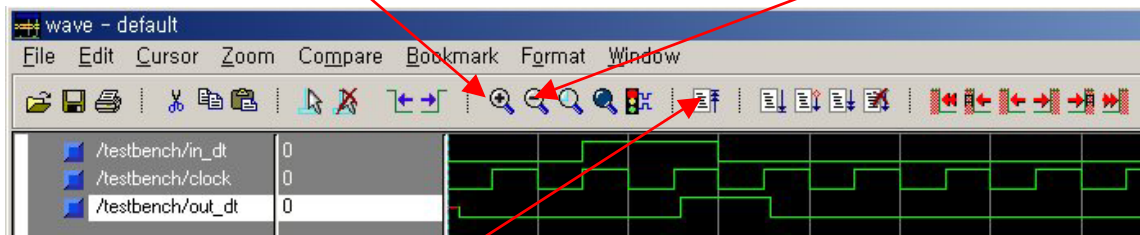


시뮬레이션 파형들은 wave창에서 볼 수 있다.



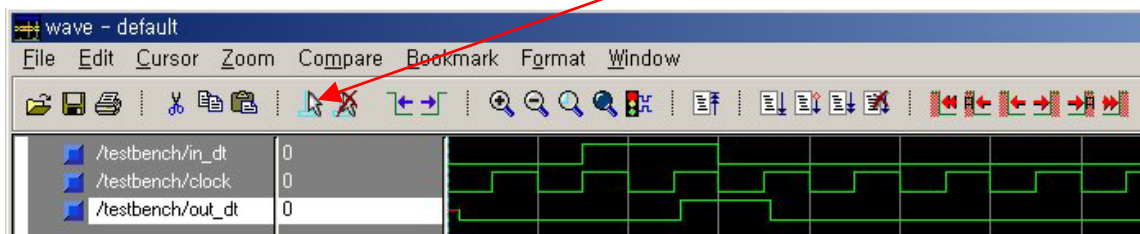
- 확대(Zoom in 2X)하려면 이곳을 클릭

- 축소(Zoom out 2X)하려면 이곳을 클릭

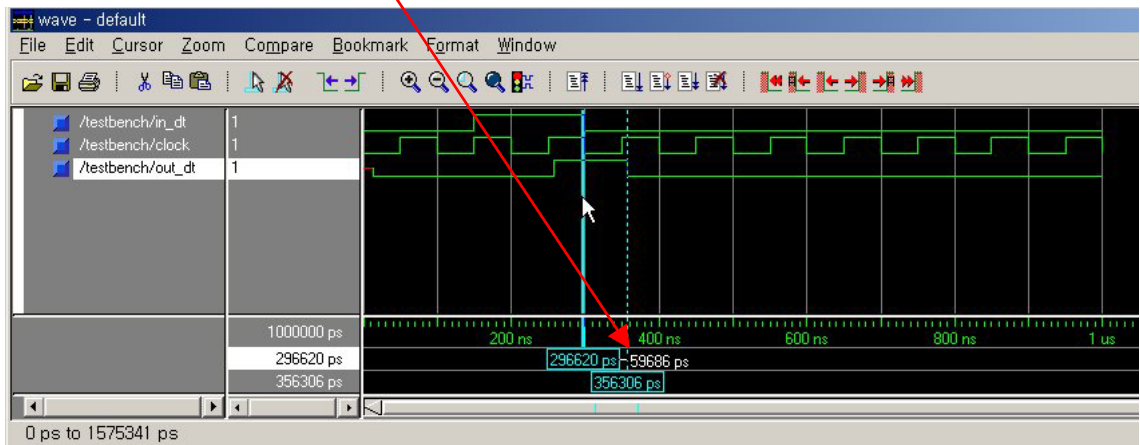


- 다시 Resimulation하려면 이곳을 클릭

- Delay를 확인하기 위해 또 하나의 지시자가 필요합니다. Add Cursor Icon을 Click합니다.



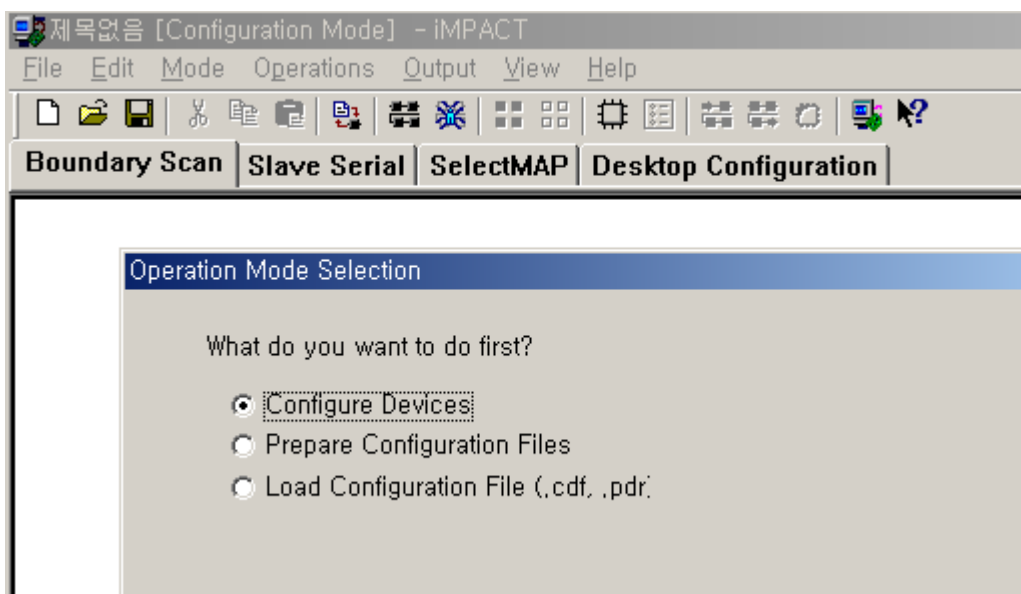
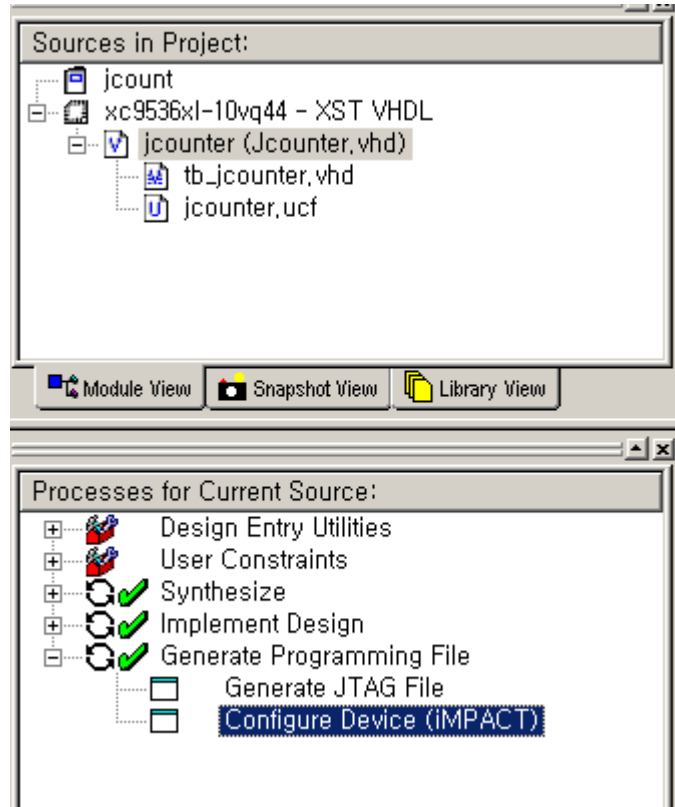
- cursor를 통해서 delay를 확인한다.



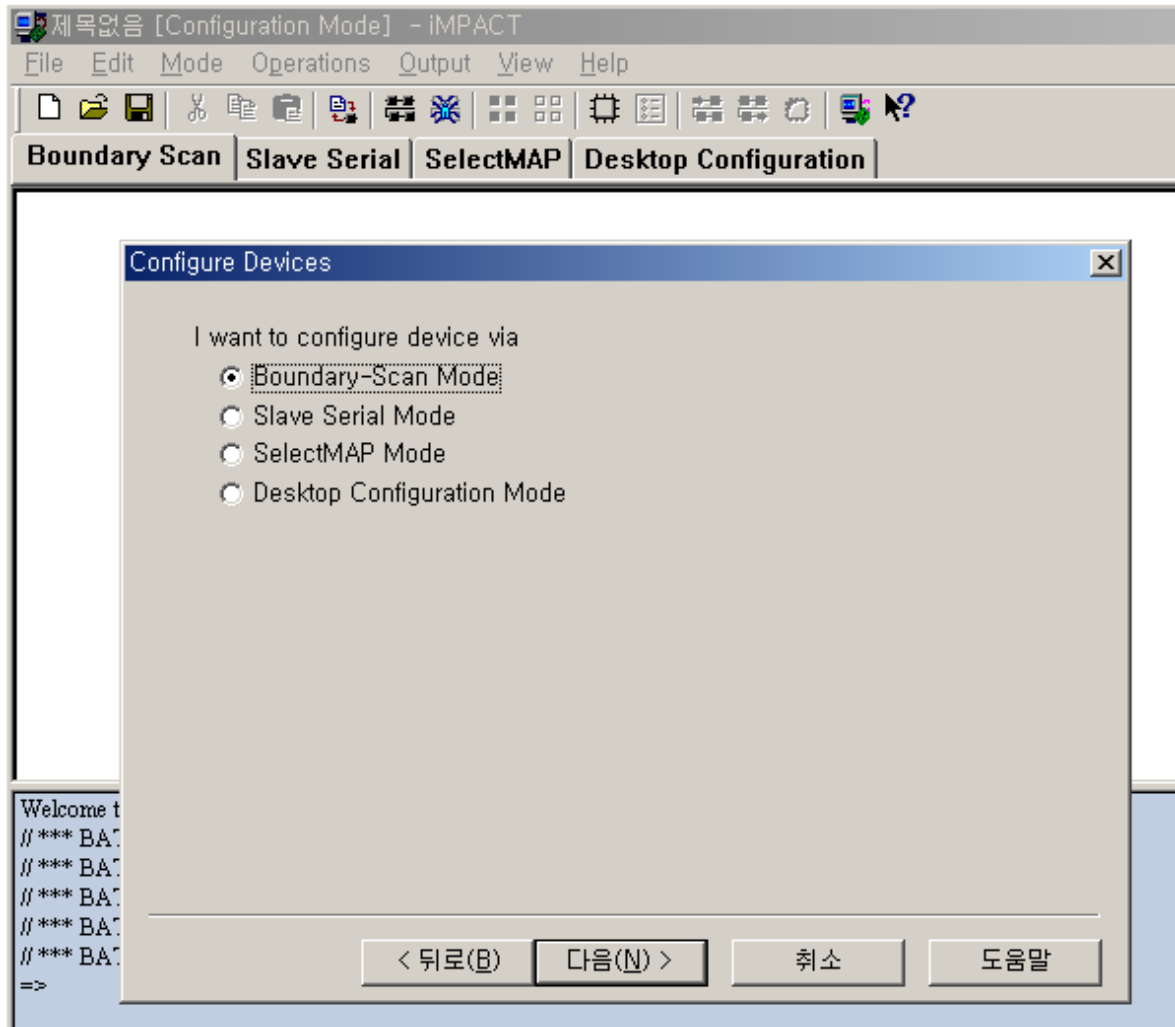
## 8. Downloading

이 장에서는 설계한 디자인의 모든 검증이 끝난 후, Downloading file을 만들어 Device에 직접 downloading 하는 과정을 설명합니다.

Project Navigator에서 아래와 같이 Configure Device ( iMPACT )를 실행합니다.



“ Prepare Configuration Files”은 FPGA로 개발 시 PROM 파일을 만들 경우 사용합니다. 이  
장에서는 CPDL를 사용함으로 이 기능은 설명하지 않습니다.



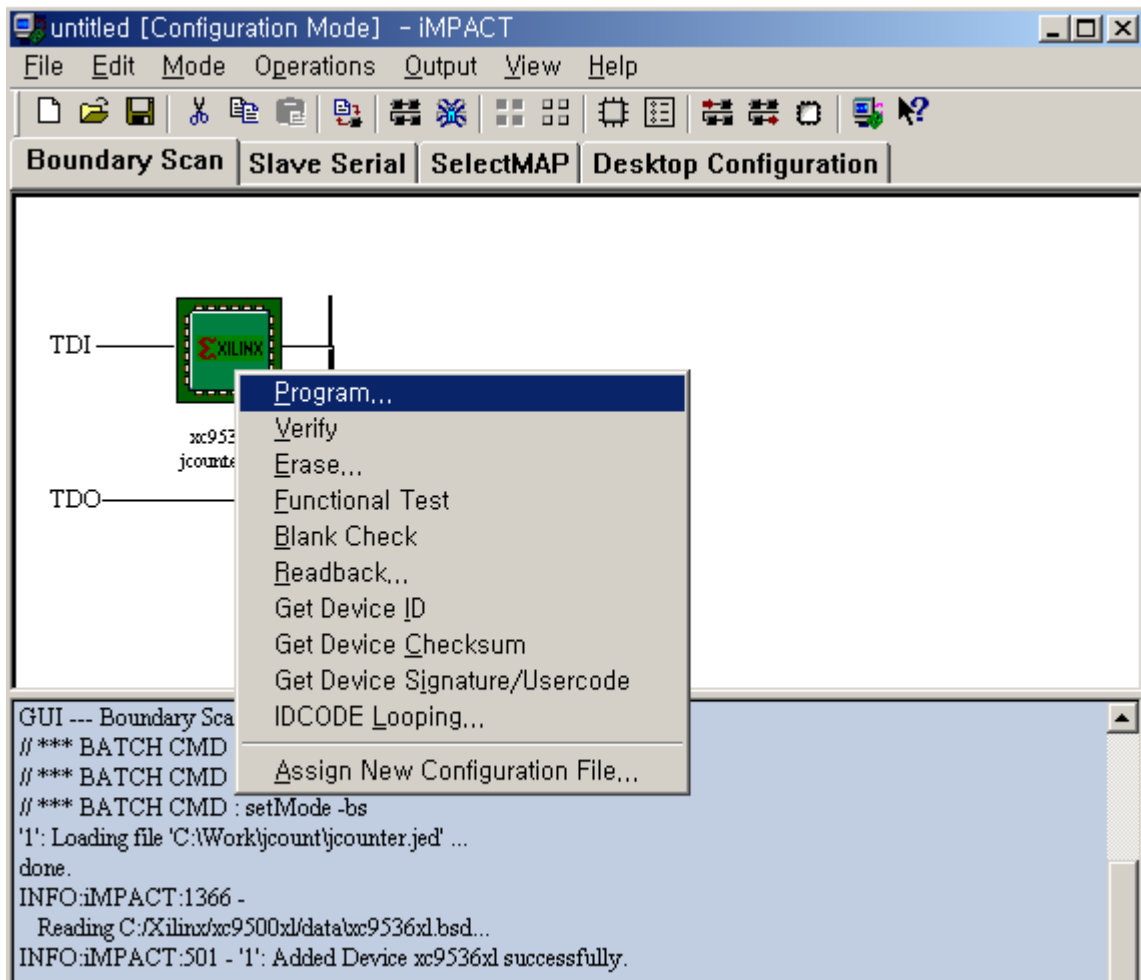
참고 : Boundary-Scan Mode : Cable을 이용하여 모든 CPLD를 Download할 경우와  
FPGA의 JTAG port를 이용하여 Download를 할 경우에 사용.

Slave Serial Mode : Cable을 이용하여 FPGA의 CCLK mode를 이용하여  
Configuration 할 경우

SelectMap Mode : FPGA의 Parallel data port를 이용한 configuration 방법







Downloading 할 준비가 모두 끝났으면, 위의 그림과 같이 Device 아이콘에 마우스 오른쪽 버튼으로 선택하여 Pop-up 메뉴를 띄우고, “Program....” 을 선택하여 Device로의 downloading 을 실행합니다.

Cable을 통하여 .JED 파일이 디바이스로 Downloading 됩니다.

## 9. Revision Table

2003. 3. 6      WebPACK ISE 5.1i      ver1.0   Release