

메모 (Technical)

Document No. : MJL-LD-AN-6

Author: 하태욱 [twha@mjl.com]

Version: 1. 1

Date: 2000 년 8 월 6 일

Subject: **Modelsim 5.3d ALTERA Edition 을 사용한 Simulation 방법**

Mentor Graphics 의 자회사인 Model Technology 에서 나온 VHDL/Verilog-HDL simulator Software 인 Modelsim 을 사용하여 VHDL 또는 Verilog-HDL 로 디자인 Source Code 를 작성한 다음 Functional, Timing Simulation 을 수행하는 방법을 기술하고자 한다.

그중 여기서는 VHDL 에 대해서만 국한하여 설명을 한다.

1. Source Code 만들기.

일반적인 Editor software 을 사용하여 작업을 한다. 여기서는 MAX+plus II 에서 HDL editor 를 사용하여 작업을 하기로 한다. 먼저 MAX+plus II 를 기동한다.

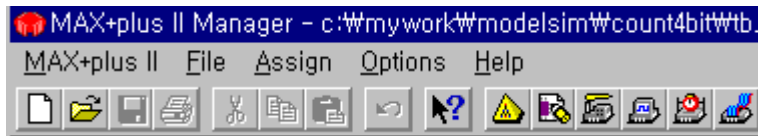


Figure 1 MAX+plus II 의 상단 Toolbar Icon

상기 그림에서 가장 좌측에 있는 아이콘을 마우스로 클릭하면 아래 그림과 같은 대화 상자가 나타난다.

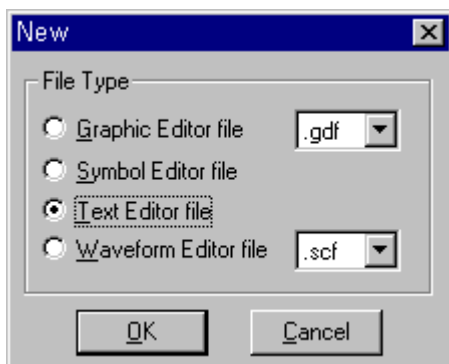


Figure 2 File Type Selector Window

위의 그림처럼 File Type 을 Text Editor file 을 선택하고 OK button 을 클릭한다.

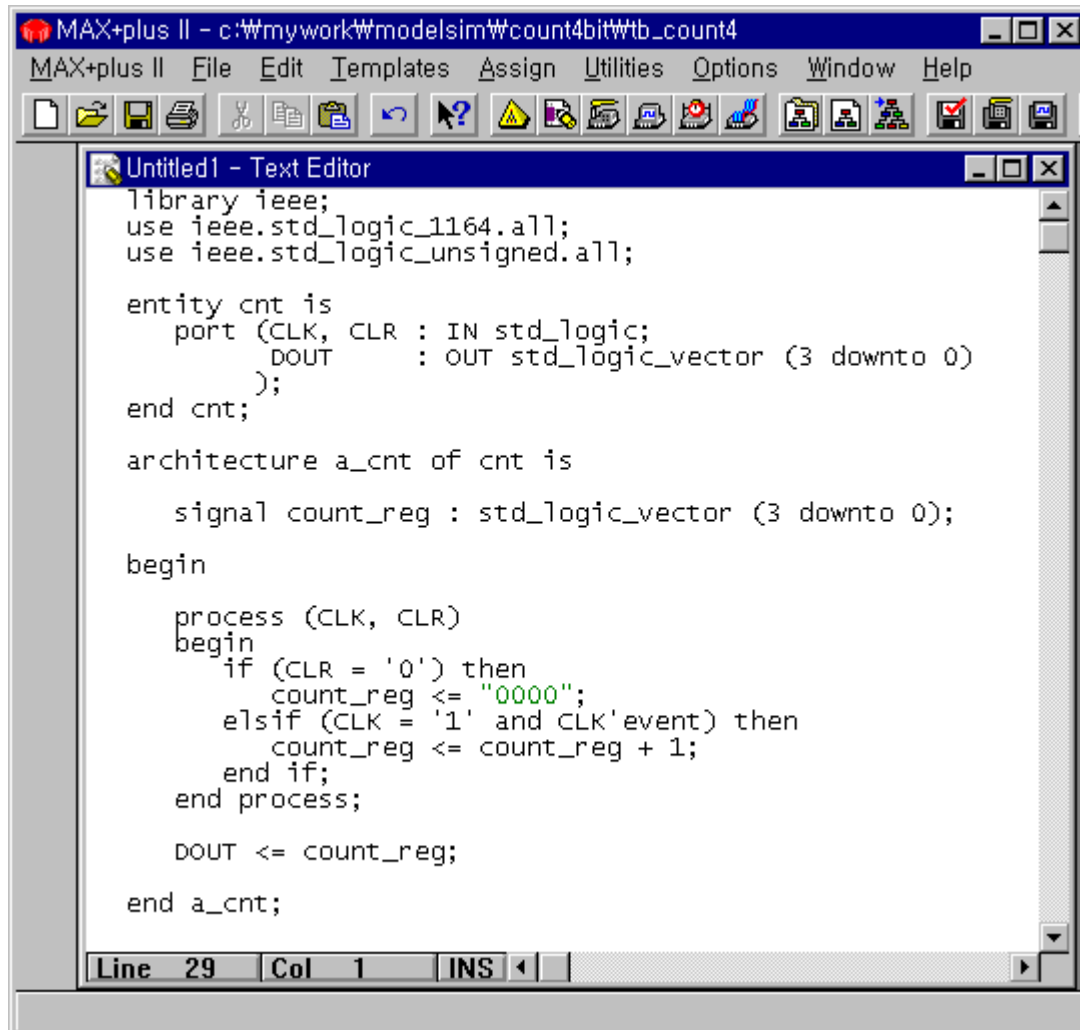


Figure 3 Untitled1 - Text Editor 창에 작성한 Source Code

상기 그림과 같이 "Untitled1 - Text Editor"라는 창이 나타나면 상기 그림과 같이 Source code 를 작성한다. 위와 같이 작성이 되면 그림 1 에서 왼쪽에서 세 번째에 위치한 Save icon 을 클릭하면 아래 그림 4 와 같이 대화창이 나타날 것이다.

그림 4 와 같이 File 이 저장될 위치 (path)를 선택한 후 File name 을 VHDL Source Code 에서 정의한 Entity name 과 동일하게 준다.

Entity name 과 같게 File name 을 주는 것은 MAX+plus II VHDL Synthesis 가 이렇게 하지 않으면 Error 를 내기 때문이다. 만일 사용자가 MAX+plus II 에 내장된 VHDL Synthesis 를 사용하지 않고 Third Vendor 용 VHDL Synthesis Tool 을 사용하면 이런 제약 사항은 없어 진다.

여기서는 C:\Wmywork\Wmodelsim\Wcounter 폴더에 cnt 라는 이름으로 파일을 저장한다.

파일 확장자를 주지 않아도 되는 것은 그림 4 에서 보는 것과 같이 Automatic Extension 이 .vhd 로 선택이 되었기 때문이다. 그러므로 사용자도 이와 같이 작업을 하기 바란다.

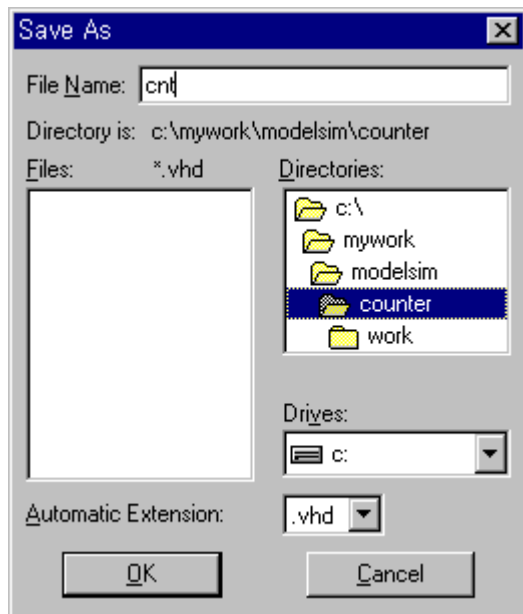


Figure 4 Save Window

OK button 을 클릭하면 그림 5 와 같이 될 것이다.

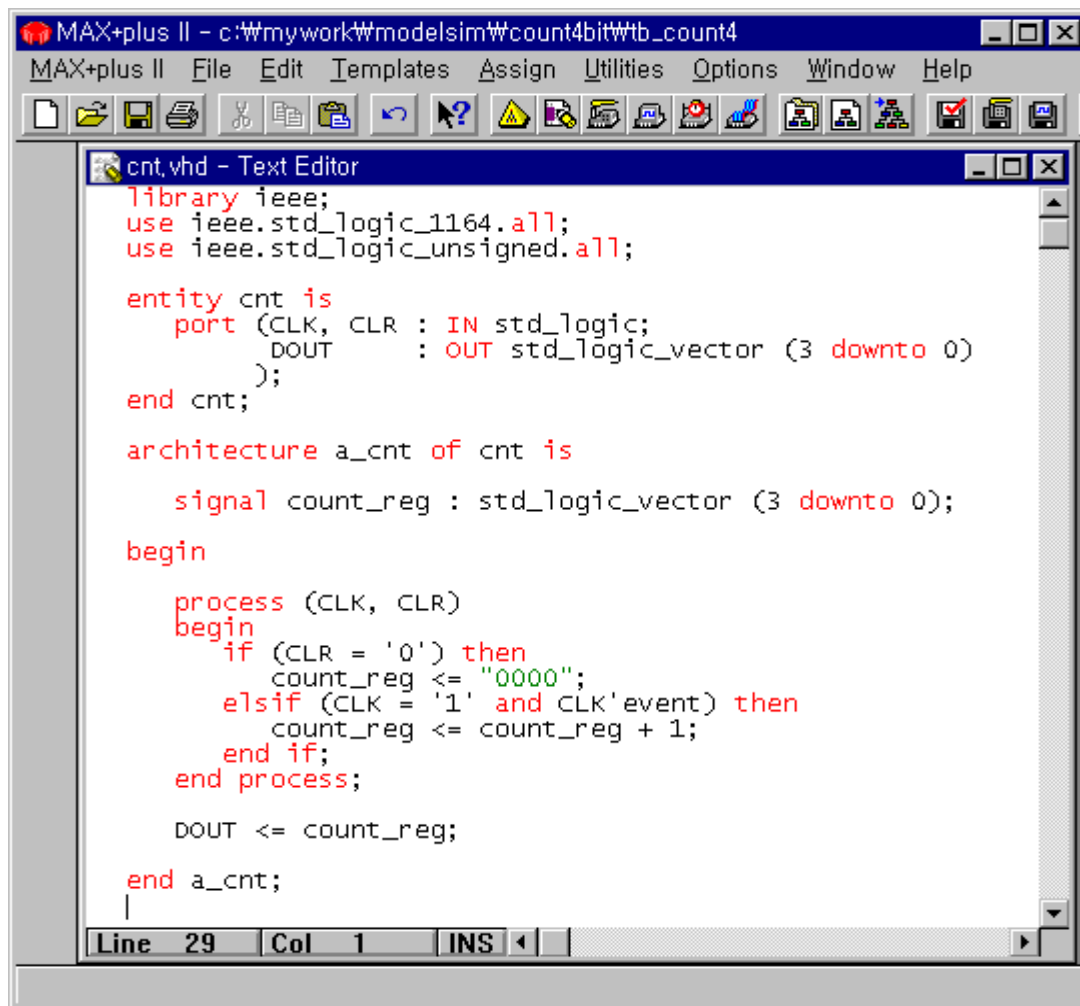


Figure 5 Source code 를 CNT.VHD 로 저장한 Window

상기와 같은 방법으로 Simulation 용 Test Bench File 을 만든다.

그림 6 과 같을 것이다.

```

MAX+plus II - c:\Wmywork\Wmodelsim\Wcount4bit\Wtb_count4
MAX+plus II File Edit Templates Assign Utilities Options Window Help
cnttb.vhd - Text Editor
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity cnttb is
end cnttb;

architecture a_cnttb of cnttb is

component cnt
  port (CLK, CLR : in std_logic;
        DOUT   : out std_logic_vector(3 downto 0)
       );
end component;

signal clr_s : std_logic := '0';
signal clk_s : std_logic := '0';
signal dout_s : std_logic_vector(3 downto 0);

begin

  clk_s <= not clk_s after 20 ns;
  clr_s <= '1' after 30 ns ;
  I1 : cnt port map (CLK =>clk_s, CLR => clr_s, DOUT => dout_s);

end a_cnttb;

configuration cnttb_c of cnttb is
  for a_cnttb
    for I1 : cnt use entity work.cnt (a_cnt);
    end for;
  end for;
end cnttb_c;
Line 1 Col 1 INS

```

Figure 6 Function Simulation 용 Testbench VHDL Source Code Window

위의 그림과 같이 Source Code 를 작성한 후 C:\Wmywork\Wmodelsim\Wcounter 폴더에 CNTTB.VHD file 로 저장한다.

위의 Testbench file 은 function simulation 용이므로 timing simulation 용 Testbench file 로 작성한다.

그림 7 과 같이 작업을 하면 된다.

Configuration 문을 사용하지 않아도 Simulation 을 수행할 수 있지만 여기서는 Configuration 문을 사용하여 작업을 수행하도록 한다. Timing simulation 용 Testbench File 에서 CNT 가 사용할 Architecture name 이 WEPF10K30ETC144-3W로 되어 있는지를 반드시 확인한다. 좀 더 자세한 사항은 Timing Simulation 을 할 때 부가 설명하도록 하겠다.

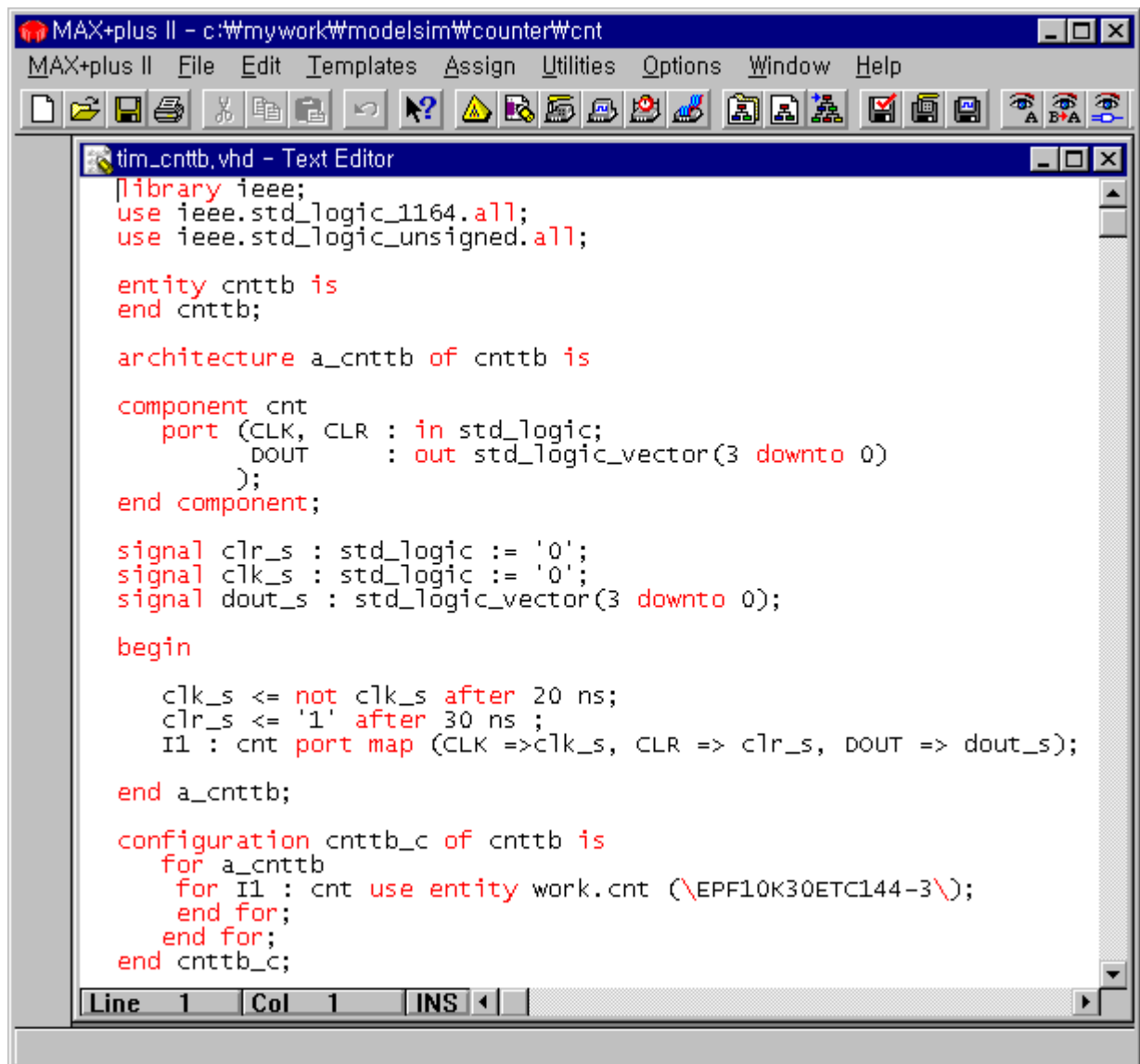


Figure 7 Timing Simulation 용 Testbench VHDL Source Code Window

2. Modelsim 을 이용하여 Function Simulation 수행하기

여기에서는 Modelsim 5.3d Altera Editon 을 사용한다.

Modelsim 에서 새로운 Project 를 만들어서 사용하는 방법도 있지만 여기서는 전형적인 방법으로 설명을 하기로 한다. ModelSim 을 기동한다. 그림 8 과 같은 화면이 나타난다. 초기화면이 아래 그림과 같지 않을 수도 있다. 여기서는 바로 Transcript Window 가 나타나는 것을 가정하고 있다.

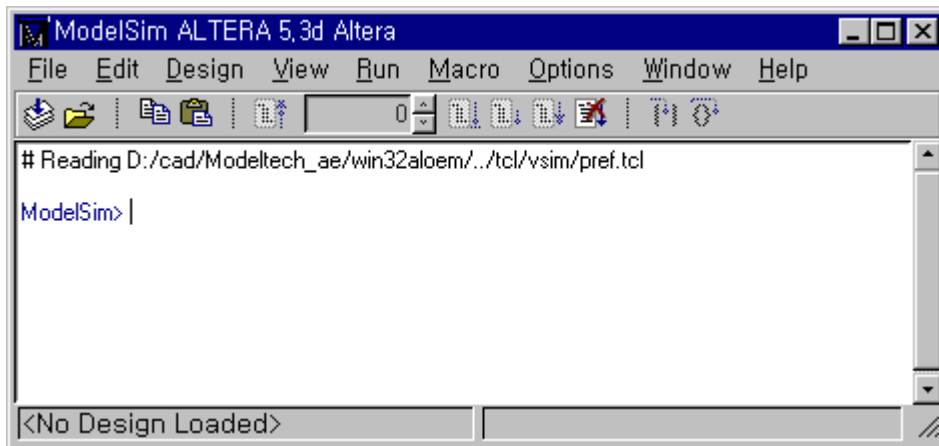


Figure 8 ModelSim 의 초기화면 (Transcript Window)

먼저 Pull-down Menu 에서 File → Change Directory... 을 선택하여 작업된 결과와 Source Code 가 있는 디렉토리를 선택한다.

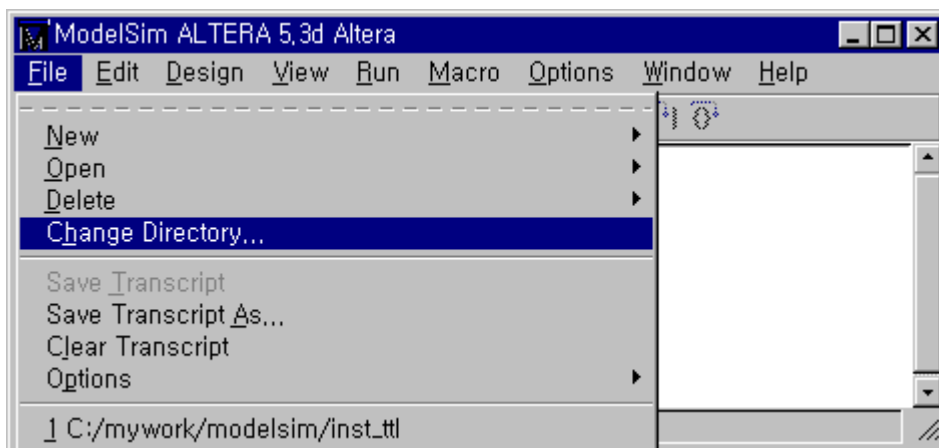


Figure 9 Change Directory Window

아래 그림과 같이 여기서는 C:\mywork\modelsim\counter 폴더를 선택한 후 열기 버튼을 클릭한다.

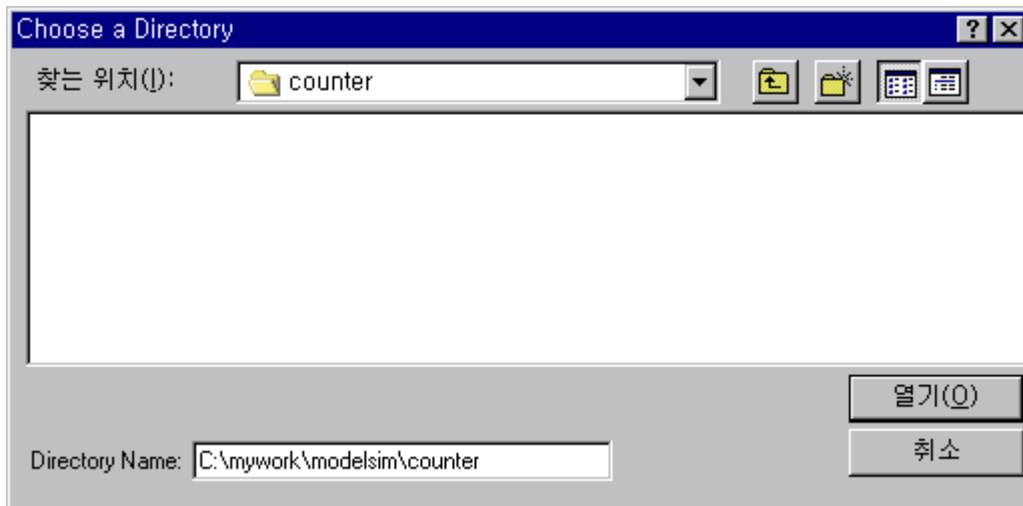


Figure 10 Choose a Directory Window

작업할 디렉토리를 선택한 후 VHDL Source Code 를 Compile 한 결과를 저장할 라이브러리를 만들어야 한다. 기본적으로 현재 선택된 디렉토리에 work 라는 library 를 만든다. Modelsim Top menu 에서 Design → Create a New Library... 메뉴를 선택하면 그림 11 과 같은 대화 창이 나타날 것이다.

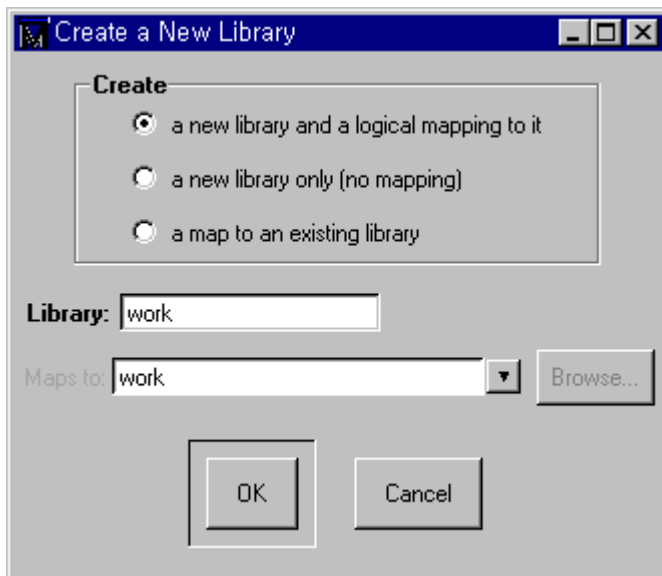


Figure 11 Create a New Library Window

위의 그림처럼 Library 에 work 을 넣고 OK 버튼을 클릭하면 현재 선택된 디렉토리에 work 이라는 서브 디렉토리를 만들고 그 곳에 Modelsim 이 작업하면서 필요한 자료들을 저장한다.

다음 단계는 VHDL Source Code 를 Compile 하여 Functional simulation 에 필요한 자료들을 만드는 과정이다. Modelsim Top menu 에서 Design → Compile... 메뉴를 선택하면 그림 12 와 같은 대화 창이 나타날 것이다.

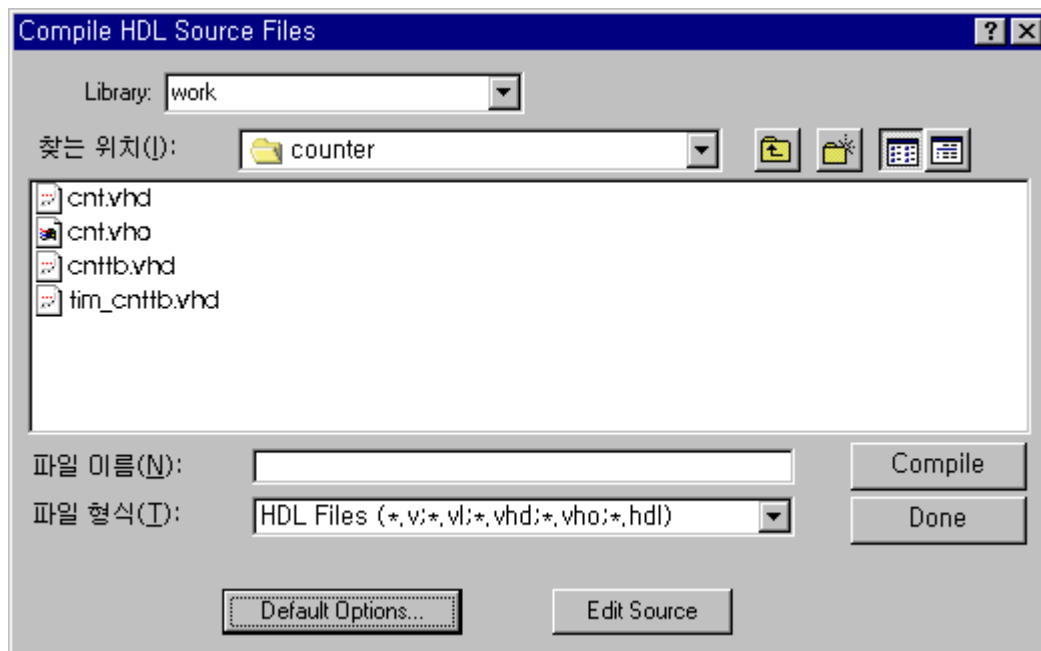


Figure 12 Compile HDL Source Files Window

위의 그림에서 Default Options... 버튼을 클릭하여 그림 13 과 같이 선택 사항들을 맞춘다.

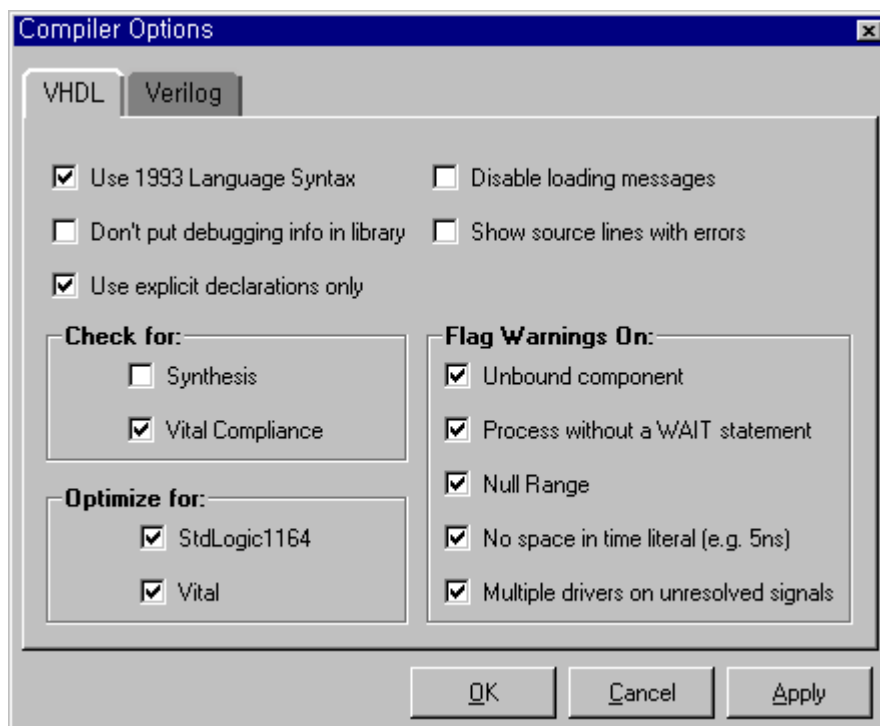


Figure 13 Compiler Options Window

필요한 선택 사항을 표시한 후 OK 버튼을 클릭하여 그림 12 화면으로 되 돌아 간다.

그림 14 와 같이 cnt.vhd 와 cnttb.vhd file 을 선택한다.

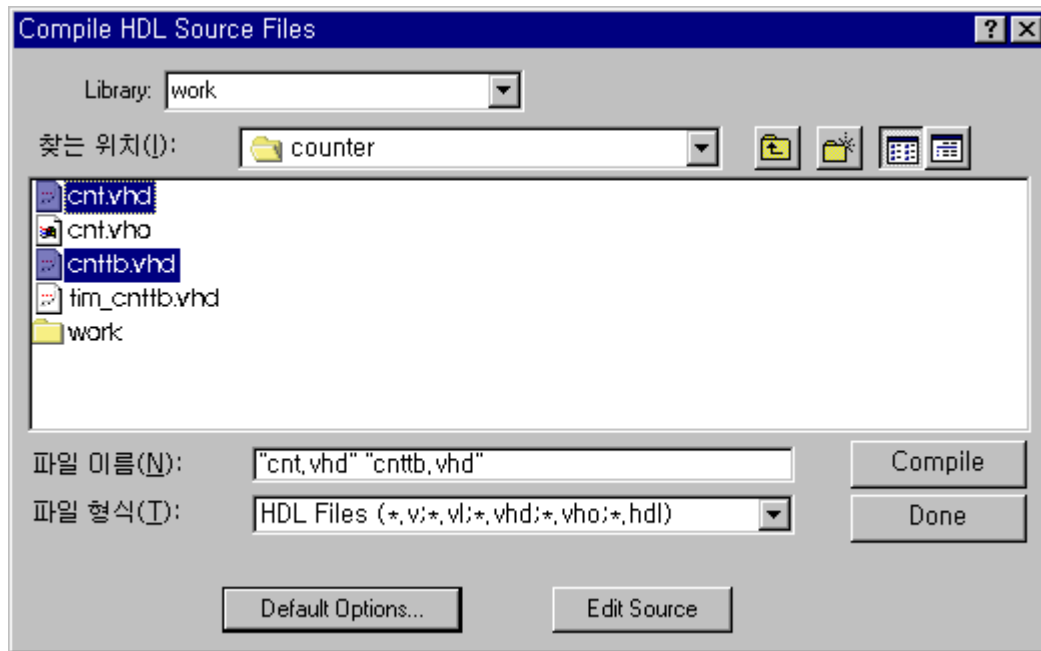


Figure 14 Compile 될 파일들이 선택된 Window

위의 그림에서 Compile 버튼을 눌러 compile 을 실행한다. 그림 15 와 같이 정상적으로 동작되었는지를 확인한다. 만일 Error 가 발생하면 문제점이 있는 부분을 수정하여 다시 Compile 을 실행하여 그림 15 와 같은 결과를 얻을 수 있도록 한다.

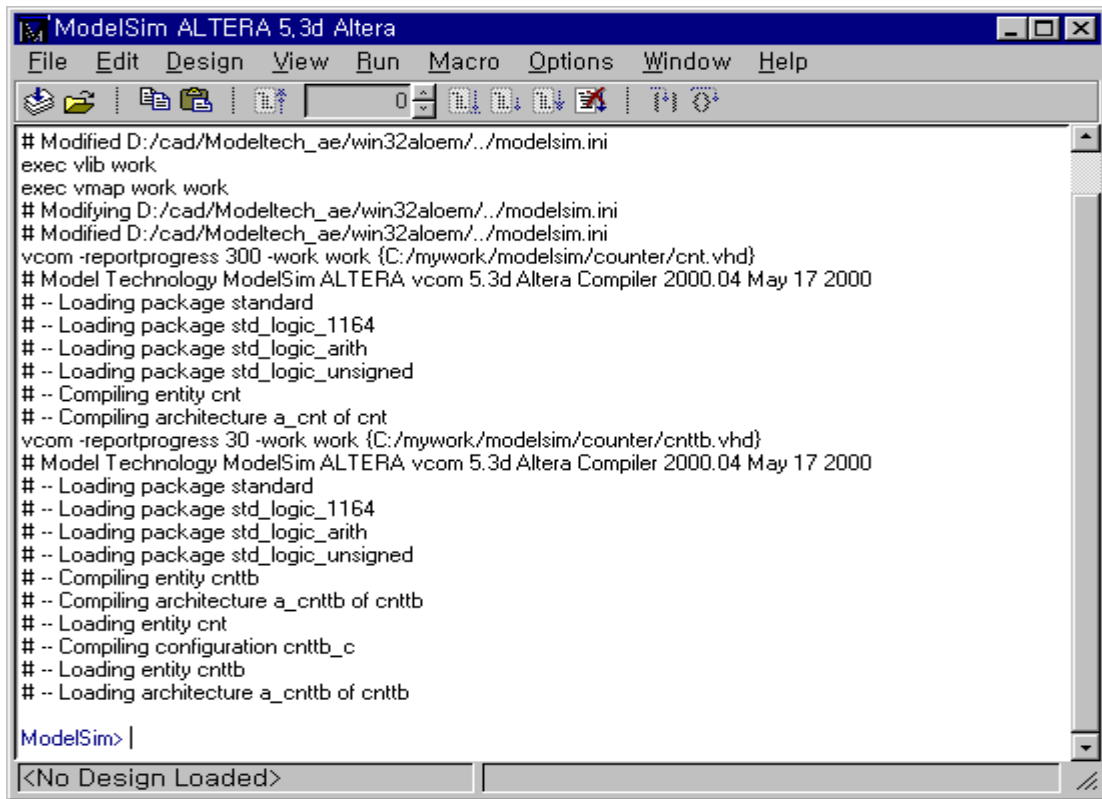


Figure 15 Compile 이 진행된 결과를 보여주는 Windows

지금까지는 Functional Simulation 에 필요한 자료를 만드는 과정이었고, 지금부터 디자인을 ModelSim 에 넣어서 Waveform 을 통해 사용자가 Coding 한 것이 올바르게 동작하는지를 확인하는 과정이다. ModelSim Top menu 에서 Design → Load New Design... 메뉴를 선택하면 그림 16 과 같은 대화 창이 나타날 것이다.

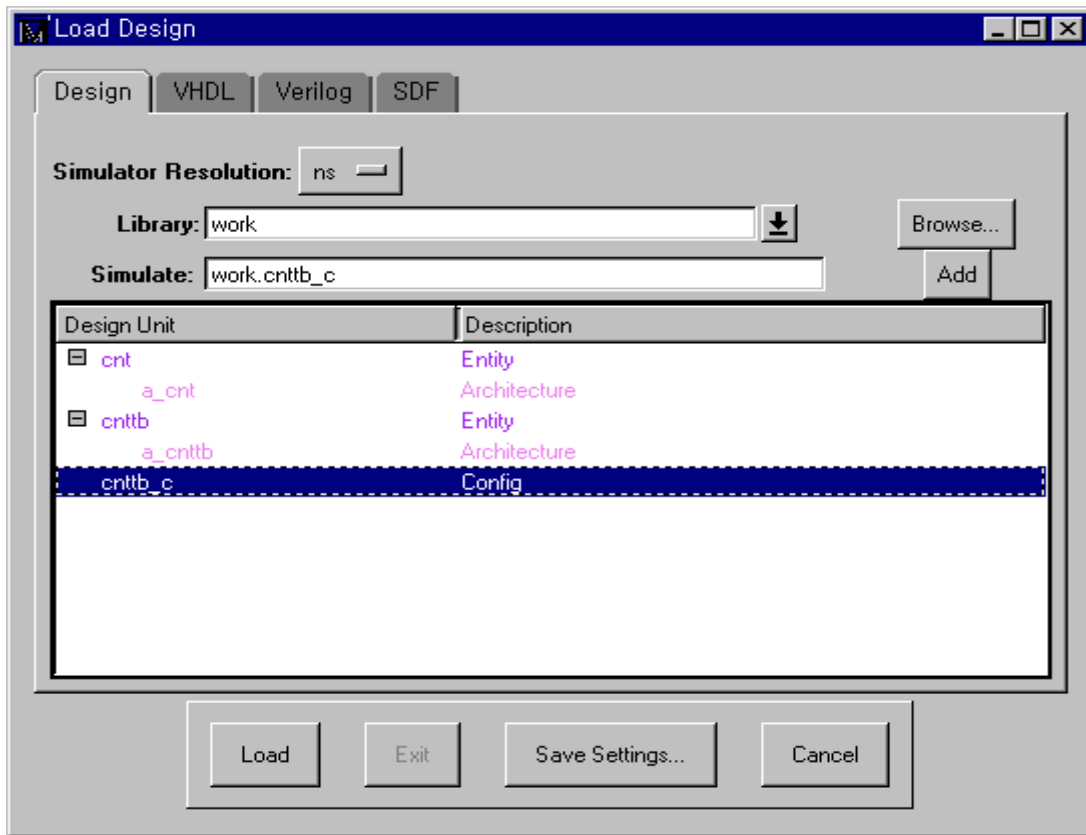


Figure 16 Load Design Window

그림 16 과 같이 선택을 한 후 Load button 을 클릭하여 그림 17 과 같은 결과가 나오는 가를 확인한다.

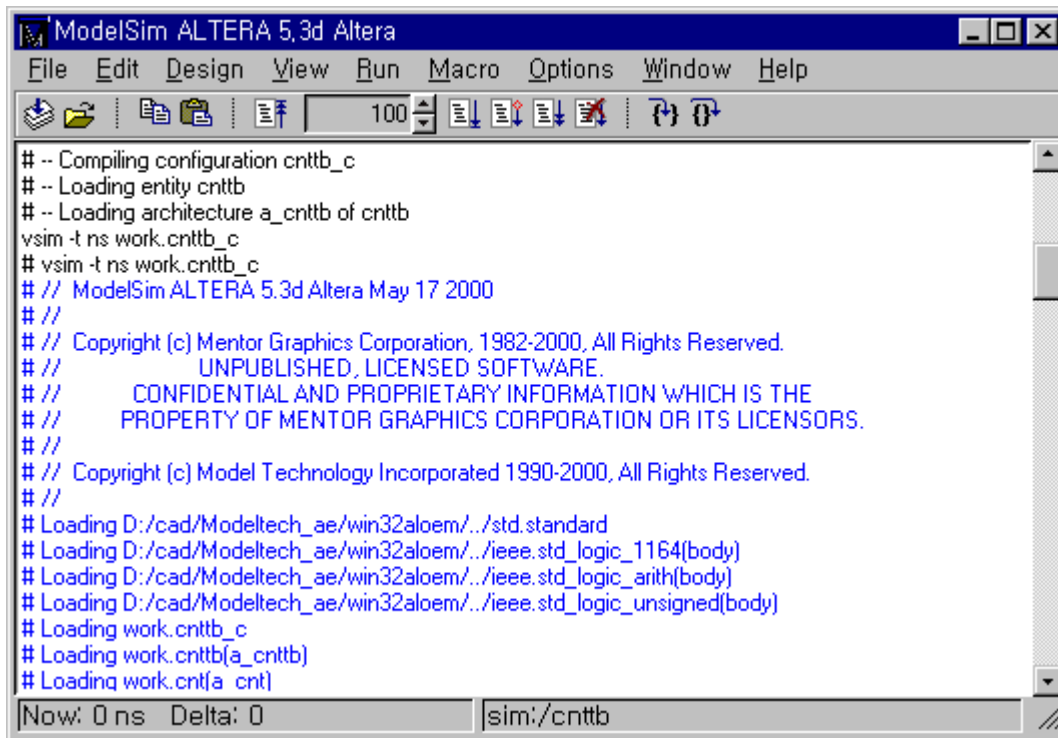


Figure 17 Design 이 Load 된 결과 Window

그림 17의 Message를 살펴보면 cnttb.vhd file의 Configuration문에서 정의한 것을 읽어 들였음을 알 수 있다. Configuration문을 한 번 살펴 보기로 하겠다.

```

configuration cnttb_c of cnttb is
  for a_cnttb
    for I1 : cnt use entity work.cnt (a_cnt);
    end for;
  end for;
end cnttb_c;

```

위의 문장을 해석하면 entity name 이 cnttb 는 cnttb_c 라는 configuration name 을 가지며, a_cnttb 라는 architecture 을 위하여 I1 이라는 instance name 을 가지며, cnt 라는 component 는 working directory 에 있는 entity name 이 cnt 이고 architecture name 이 a_cnt 라는 VHDL code 을 사용하라는 뜻이다.

다음 단계는 Modelsim 에서 사용하는 모든 창들을 열어서 Simulation 을 직접 수행하는 것이다. ModelSim Top menu 에서 View → All 메뉴를 선택하면 그림 18 과 같은 9 개의 대화 창이 나타날 것이다.

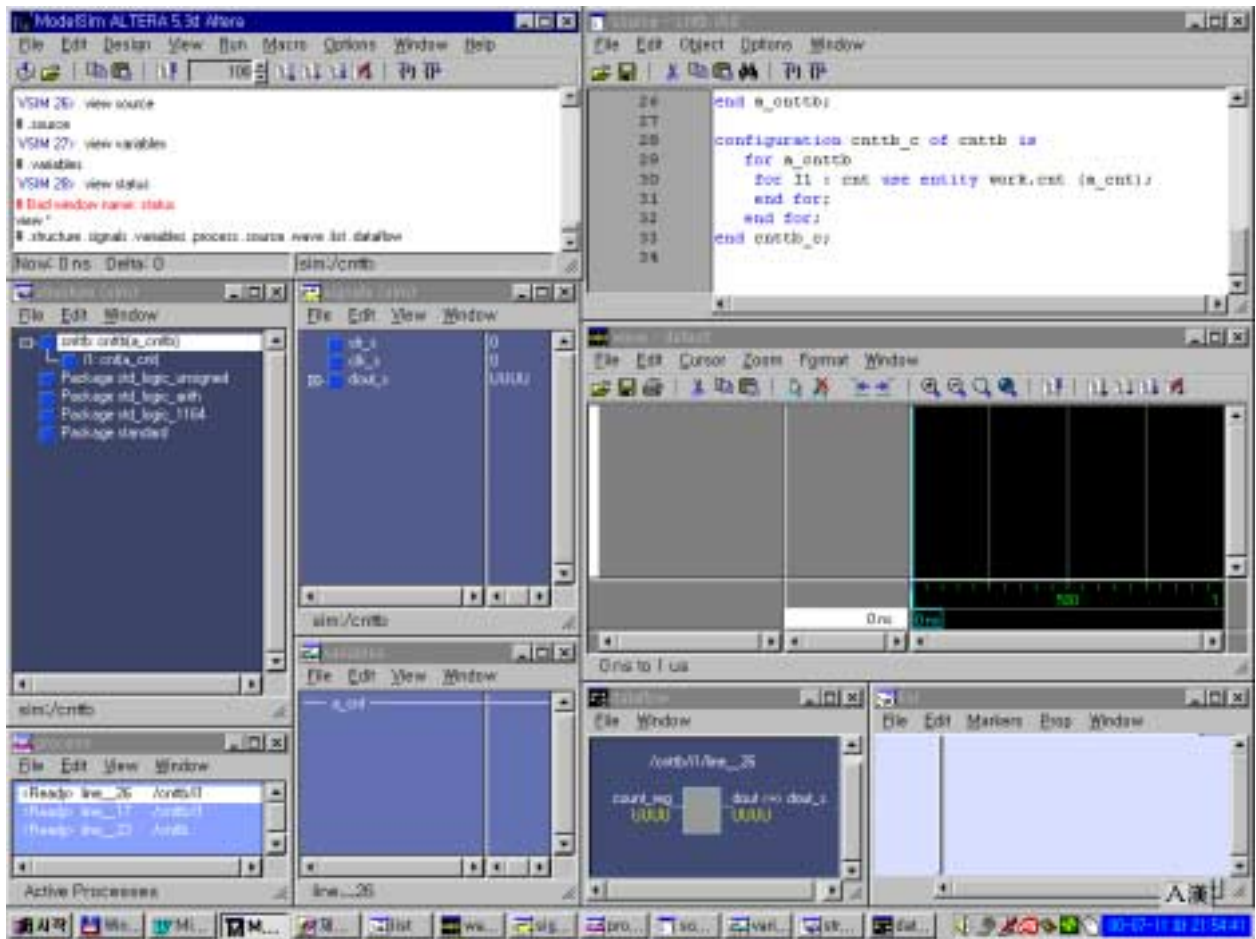


Figure 18 ModelSim 의 9 개 대화 창들

위의 각 창들에 대한 설명은 아래를 참조하길 바란다.

- **Transcript** – Simulation 수행을 제어한다.
- **Source** – VHDL source 를 read only mode 로 표시한다.
- **Structure** – Design 의 Hierachical view 를 표시한다.
- **Variables** – 현재 Process 내부의 constant, variable 및 generic 들을 표시한다.
- **Signals** – Structure window 에서 선택된 영역에 대한 signal 및 signal value 를 표시한다.
- **List** – Simulation 결과를 table 형태로 표시한다.
- **Process** – Design 내부의 Process 들을 표시한다.
- **Wave** – Simulation 수행 결과를 wave 로 표시한다.
- **Dataflow** – HDL item 들의 flow 를 표시한다.

Signals 대화창에 있는 Top menu 의 View → Wave → Selected in Region 을 선택하여 보고자 하는 신호들을 Wave window 에 표시한다.

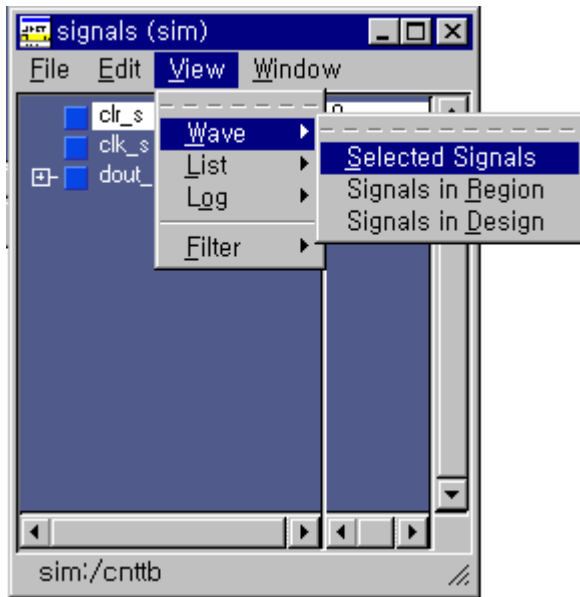


Figure 19 Wave 에 보고자 하는 신호를 선택하는 Window

그림 20 과 같이 Wave window 에 보고자 하는 신호들이 올라 오는지를 확인한다.

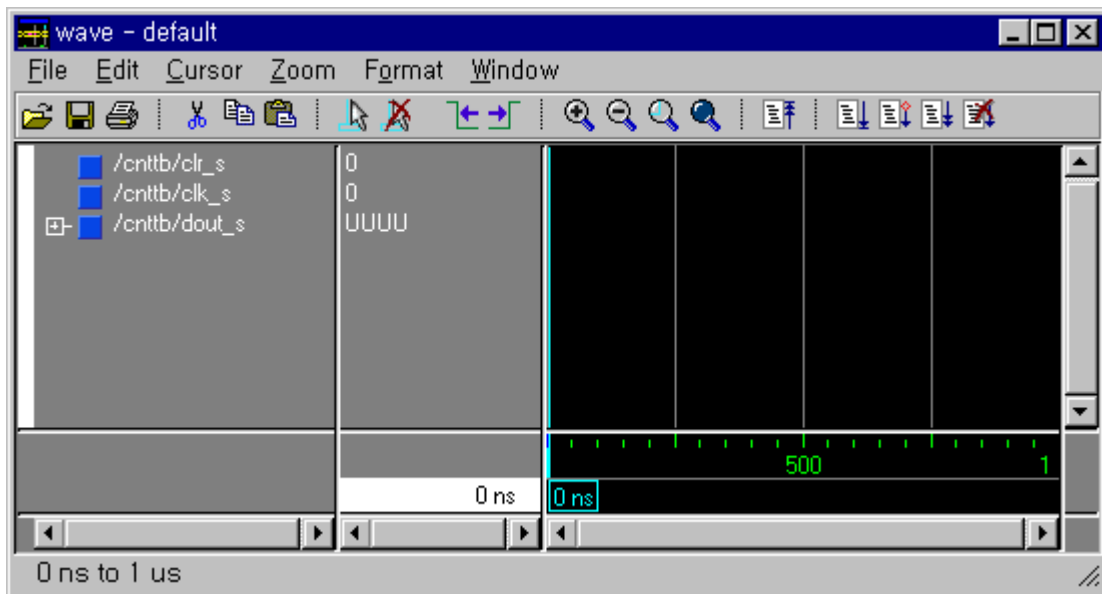


Figure 20 Wave Window 에 cnttb 의 Signal 들이 실린상태

최종적으로 ModelSim 대화창에서 run 1000 명령을 실행시킨 후 wave window 를 살펴보기로 한다. 그림 21 과 같은 모양이 될 것이다.

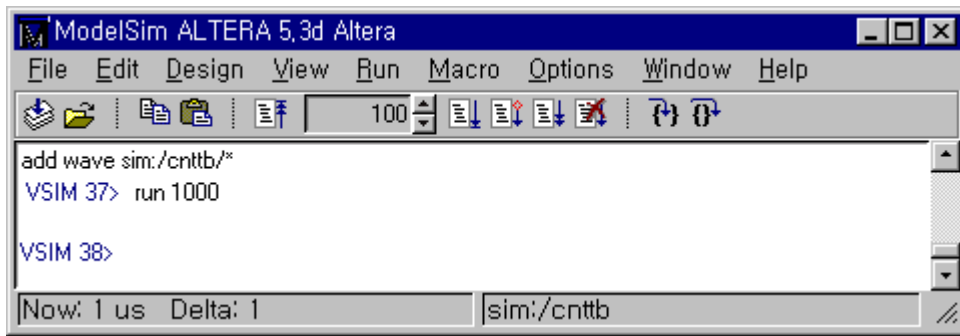


Figure 21 1000ns Simulation 을 수행하는 Transcript Window

wave window 의 "+" 표시 Icon 을 눌러 화면을 확대하여 counter 가 제대로 동작하고 있는 지를 확인한다. 그림 22 와 같을 것이다.

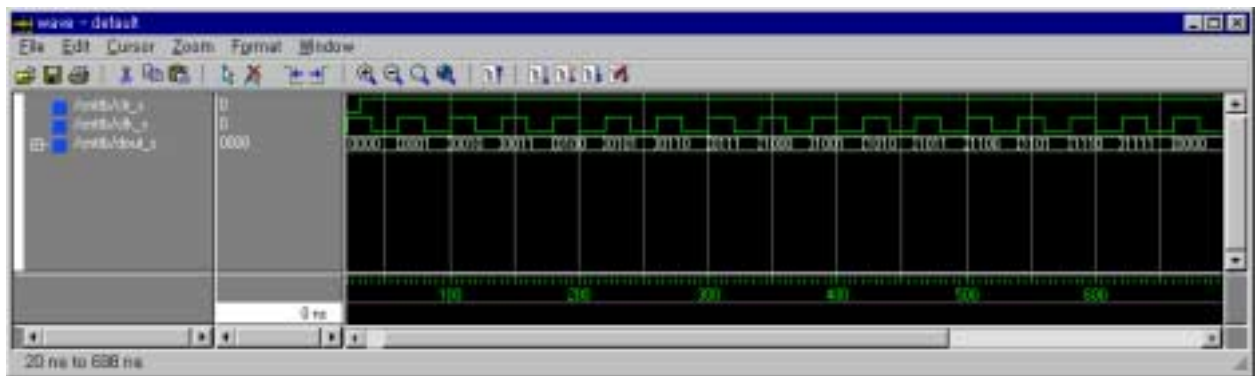


Figure 22 Counter 의 결과가 표시된 Wave Window

지금까지 수행한 것은 net delay 또는 logic delay 가 없는 Function Simulation 을 수행한 것이다. 다음 단계는 Source code 를 Synthesis 를 하여 MAX+plus II 또는 Quartus 에서 사용하는 디바이스 (CPLD, FPGA)에 맞춰서 Place & Route 를 수행하고 Timing Simulation 을 수행할 Data 를 만들어 내는 과정이다.

3. MAX+plus II 에서 Place & Route (Compile) 및 VHDL output, SDF (Standard Delay Format) 만들기.

MAX+plus II 에서 VHDL OUTPUT 정의하기.

MAX+plus II 에서 cnt.vhd file 을 open 한 후 이 파일을 Project 로 정의 한다.

Assign → Device menu 에서 EPF10K30ETC144-3 Device 을 설정한다.



아이콘을 클릭하면 Compile window 가 foreground 형태로 나타난다.

Interface → VHDL Netlist Writer 를 선택한다. 그림 3-1 에서 보는 것과 같이 VHDL Netlist Writer Menu 왼쪽에 check mark 가 보일 것이다. Interface → VHDL Netlist Writer Settings...을 선택한다.

그림 23 이 나타날 것이다. 대화 상자 안의 선택 사항을 그림 24 와 동일하게 한다.

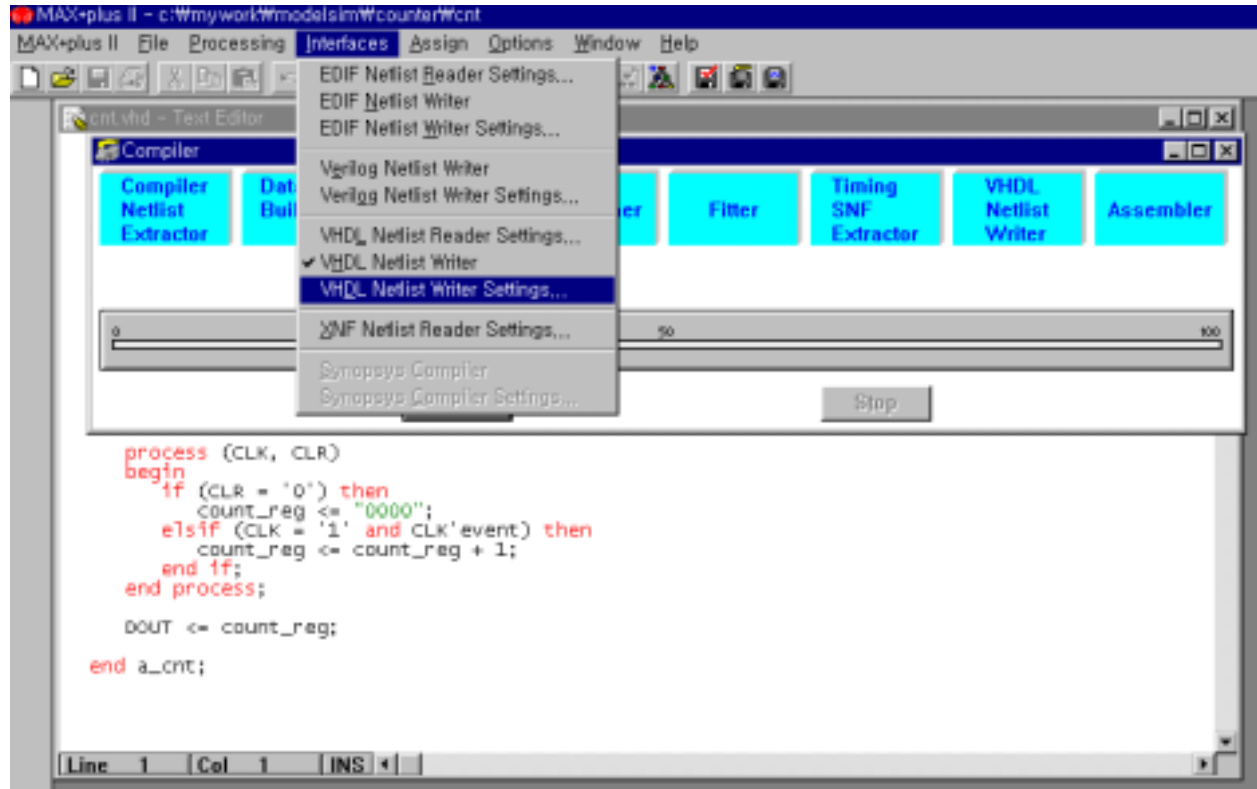


Figure 23 VHDL Netlist Writer Window

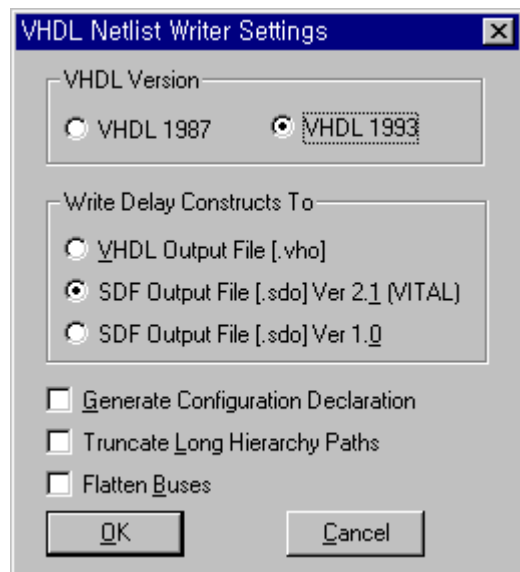


Figure 24 VHDL Option Window

위의 그림에서 보는 것과 같이 VHDL OUTPUT 형식을 VHDL 1993 으로 맞추고 Delay 정보를 VITAL 을 사용하는 SDF Output File [.sdo] Ver 2.1 (VITAL)을 선택한다. OK Button 을 클릭하여 다음 단계로 넘어 간다.

Compile 한 후 VHDL Output File 과 Delay File 확인하기

Compile 이 완료되면 작업 디렉토리에 CNT.VHO 와 CNT.SDO 두개의 file 이 만들어 졌을 것이다. 여기서 CNT.VHO File 은 Altera 의 Physical component 로 표현된 VHDL output file 이며 CNT.SDO 는 Altera 의 Physical component 의 delay 정보를 가지고 있는 Delay file 이다.

4. ModelSim 에서 Timing Simulation 수행하기.

Vital Library 에 대하여...

Altera VHDL simulation model libraries (VITAL)

Modelsim 에서 Timing simulation 을 수행하기 위해서는 우선 Altera 용 VHDL simulation model libraries 가 필요하다. ModelSim 5.3d Altera Edition 은 ModelSim directory 의 altera subdirectory 에 이미 Altera 용 VHDL, Verilog Vital Libraries 가 제공이 되고 있으므로 바로 사용을 하면 된다. 만약 없을 시에는 사용자가 만들어 주어야 한다.

ModelSim 에서 alt_vtl library 확인하기.

ModelSim 의 Top menu 에서 Design → Browse libraries... 을 선택하면 그림 4-1 창이 나온다.

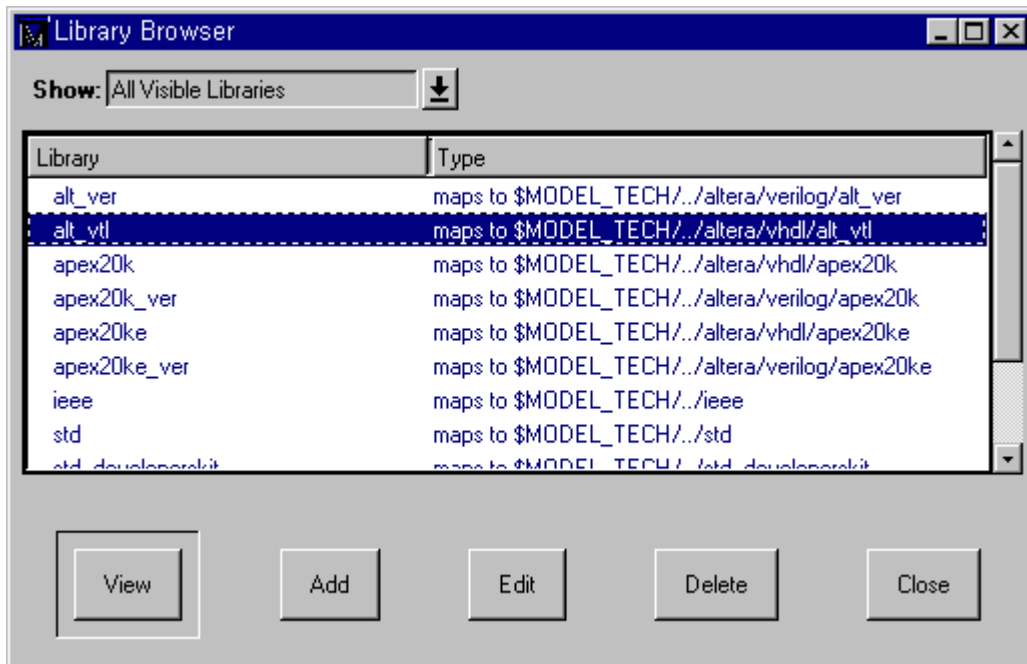


Figure 25 Library Browser Window

위의 그림에서 보면 ALTERA 용 VITAL library 가 이미 만들어져 mapping 되어 있음을 알 수 있다. Close button 을 클릭한다.

CNT.VHO 와 TIM_CNTTB.VHD File Compile 하기.

ModelSim Top menu 에서 Design → Compile...을 선택하여 MAX+plus II 에서 생성된 cnt.vho file 을 compile 한다. 그림 26 에 있는 것과 동일하게 작업을 한다.

이상 없이 수행이 되면 그림 27 과 같이 Transcript window 의 결과가 같은지 확인한다. 만일 Error 가 있으면 그림 27 의 내용과 같이 될 때까지 문제점을 찾아 수정하여 재 compile 한다.

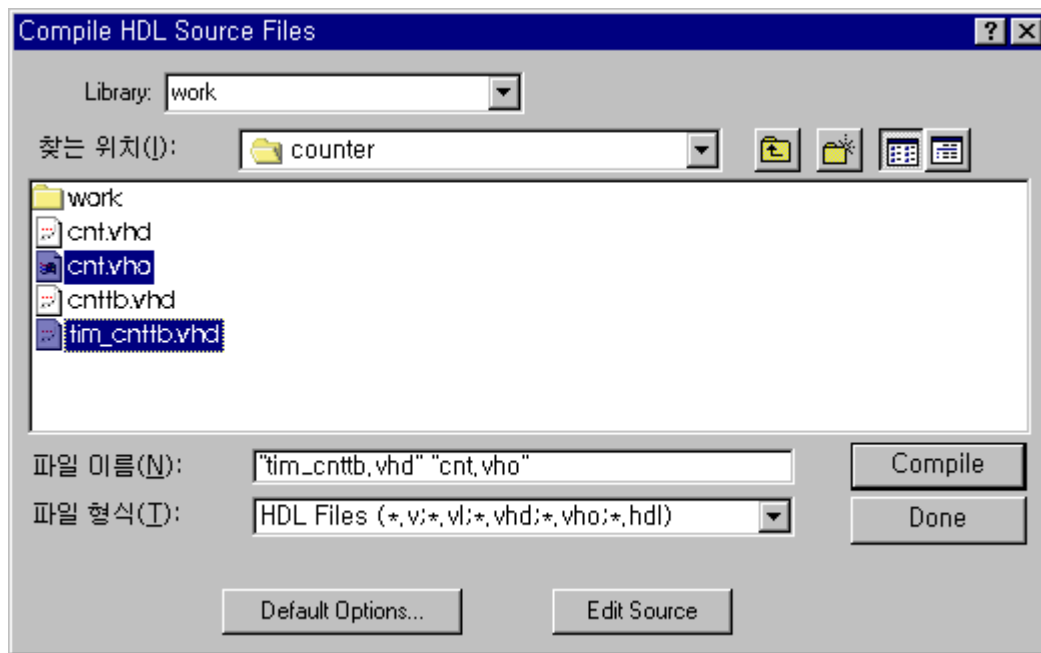


Figure 26 Timing Simulation 용 VHDL File Compile Window

```

ModelSim ALTERA 5.3d Altera
File Edit Design View Run Macro Options Window Help
# Reading D:/cad/Modeltech_ae/win32aloem/./tcl/vsim/pref.tcl
cd C:/mywork/modelsim/counter
# Modified D:/cad/Modeltech_ae/win32aloem/./modelsim.ini
vcom -reportprogress 300 -work work {C:/mywork/modelsim/counter/tim_cnttb.vhd}
# Model Technology ModelSim ALTERA vcom 5.3d Altera Compiler 2000.04 May 17 2000
# -- Loading package standard
# -- Loading package std_logic_1164
# -- Loading package std_logic_arith
# -- Loading package std_logic_unsigned
# -- Compiling entity cnttb
# -- Compiling architecture a_cnttb of cnttb
# -- Loading package vital_timing
# -- Loading package vcomponents
# -- Loading entity cnt
# -- Compiling configuration cnttb_c
# -- Loading entity cnttb
# -- Loading architecture a_cnttb of cnttb
vcom -reportprogress 30 -work work {C:/mywork/modelsim/counter/cnt.vho}
# Model Technology ModelSim ALTERA vcom 5.3d Altera Compiler 2000.04 May 17 2000
# -- Loading package standard
# -- Loading package std_logic_1164
# -- Loading package vital_timing
# -- Loading package vcomponents
# -- Compiling entity cnt
# -- Compiling architecture \EPF10K30ETC144-3\ of cnt
# -- Loading package vital_primitives
# -- Loading package support
# -- Loading entity tribuf
# -- Loading entity delay
# -- Loading entity xor2
# -- Loading entity or1
# -- Loading entity and1
# -- Loading entity dff
# -- Loading entity inv
# -- Loading entity filter
# -- Loading entity or2
# -- Loading entity and2
# -- Loading entity or3
# -- Loading entity and3
# -- Loading entity or4
# -- Loading entity and4

ModelSim> |
<No Design Loaded>

```

Figure 27 Compile 된 결과 화면

Done button 을 클릭하여 다음 단계로 진행한다.

Load a New Design.

Standard Delay Format 으로 만들어진 CNT.SDO 파일을 TIM_CNTTB.VHD 에 적용하여 실제로 얼마의 physical delay 를 가지는지 알 수가 있다.

Design → Load a New Design 을 선택한다.

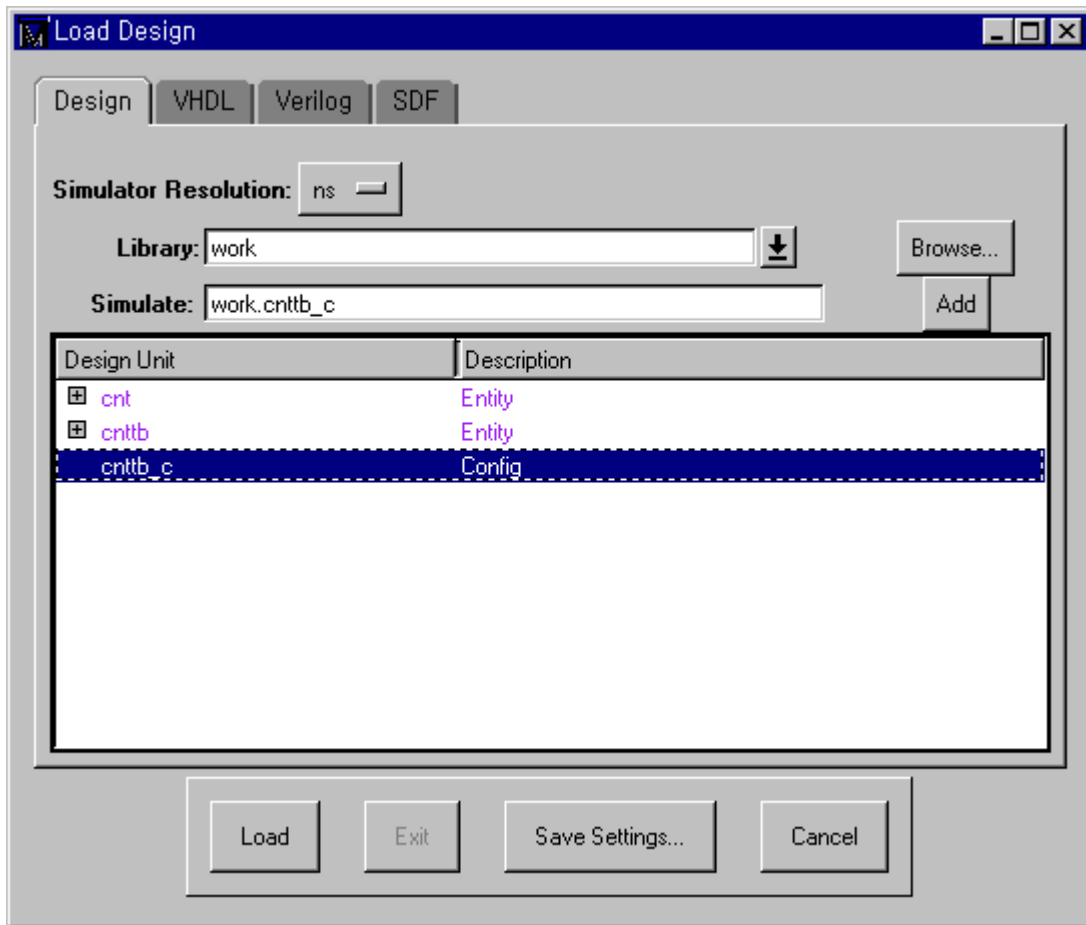


Figure 28 Load a New Design Window

위의 그림에서 SDF Tab 을 눌러서 MAX+plus II 에서 만든 CNT.SDO File 을 정의한다. ADD Button 을 눌러 SDO file 이 있는 경로를 찾아 cnt.sdo 를 정의하고 tim_cnttb.vhd 파일에서 정의한 component instance name 인 I1 을 Apply to Region 에 넣어 준다.

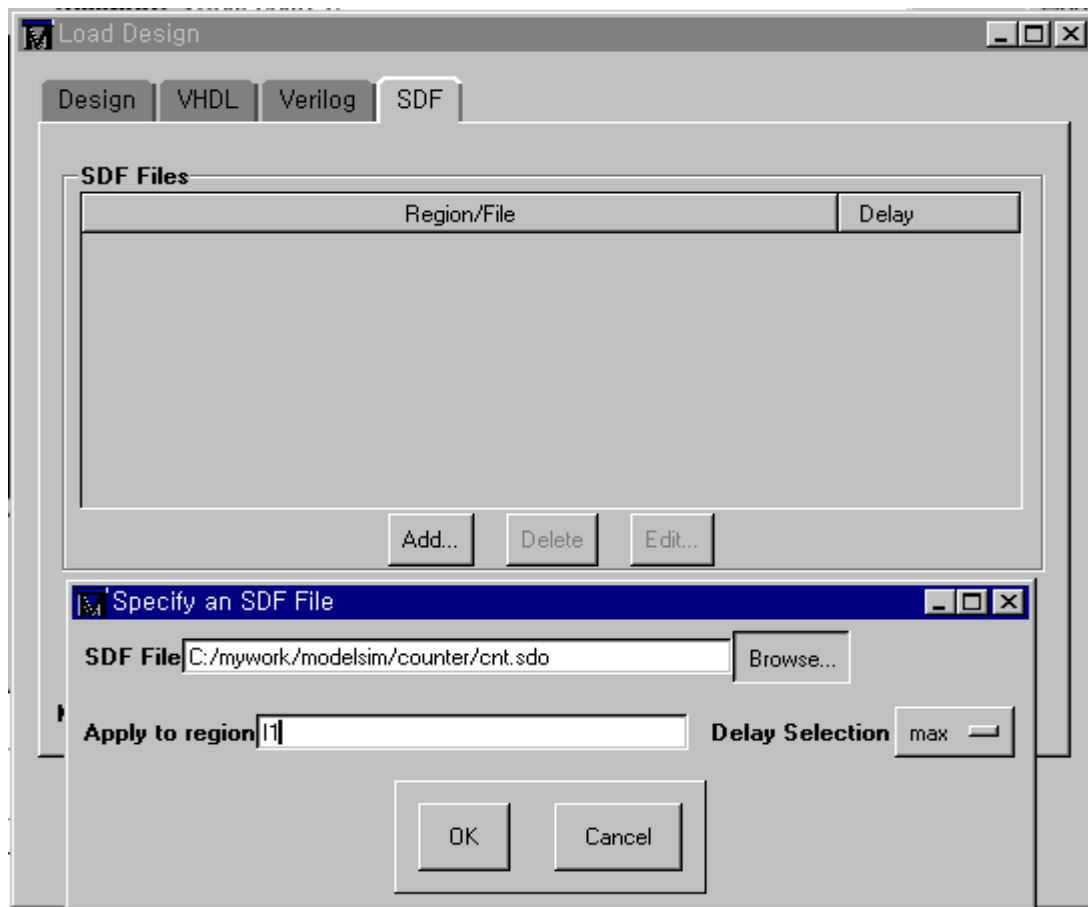


Figure 29 SDO File 정의하기

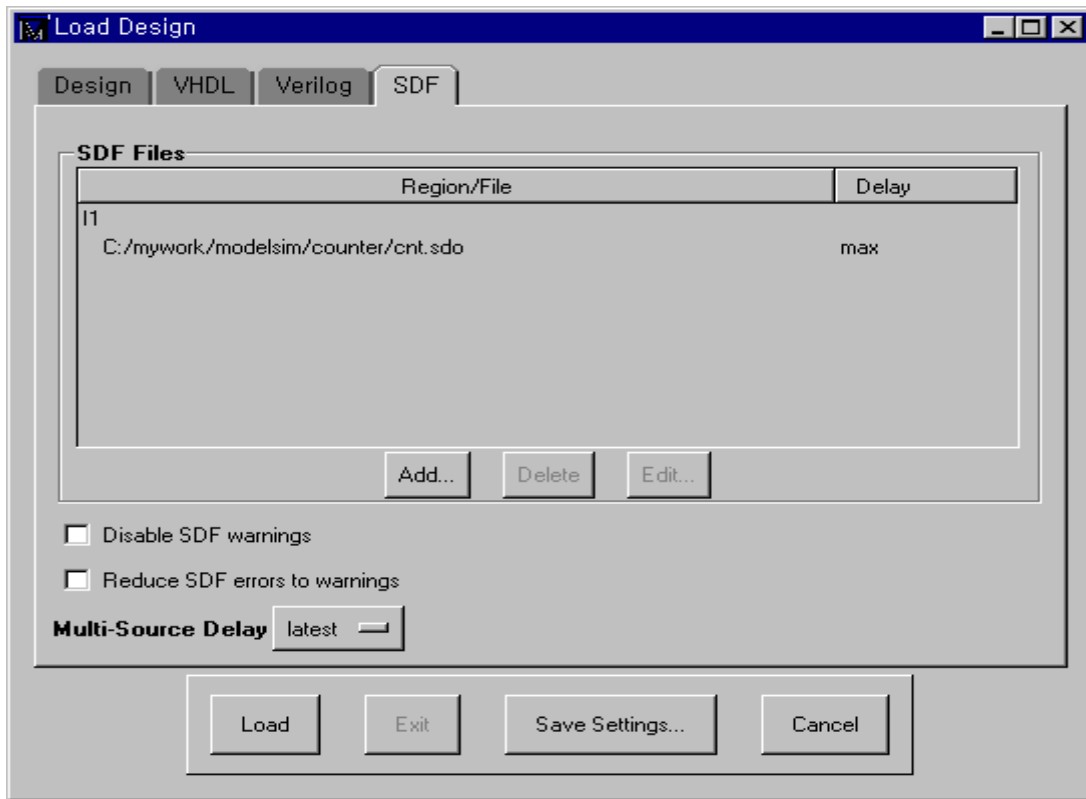


Figure 30 SDO File 을 정의한 SDF Window

Load button 을 눌러 디자인을 불러들인다.


```

ModelSim ALTERA 5.3d Altera
File Edit Design View Run Macro Options Window Help
100
vsim -sdfmax I1=C:/mywork/modelsim/counter/cnt.sdo -multisource_delay latest -t ns work.cnttb_c
# vsim -t ns -sdfmax I1=C:/mywork/modelsim/counter/cnt.sdo -multisource_delay latest work.cnttb_c
# Loading C:/mywork/modelsim/counter/cnt.sdo
# // ModelSim ALTERA 5.3d Altera May 17 2000
# //
# // Copyright (c) Mentor Graphics Corporation. 1982-2000. All Rights Reserved.
# // UNPUBLISHED, LICENSED SOFTWARE.
# // CONFIDENTIAL AND PROPRIETARY INFORMATION WHICH IS THE
# // PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS.
# //
# // Copyright (c) Model Technology Incorporated 1990-2000. All Rights Reserved.
# //
# Loading D:/cad/Modeltech_ae/win32aloem/./std.standard
# Loading D:/cad/Modeltech_ae/win32aloem/./ieee.std_logic_1164(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./ieee.std_logic_arith(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./ieee.std_logic_unsigned(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./std.textio(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./ieee.vital_timing(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.vcomponents
# Loading work.cnttb_c
# Loading work.cnttb(a_cnttb)
# Loading D:/cad/Modeltech_ae/win32aloem/./ieee.vital_primitives(body)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.support
# Loading work.cnt(\EPF10K30ETC144-3\)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.tribuff(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.delay(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.xor2(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.or1(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.and1(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.dff(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.inv(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.filter(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.or2(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.and2(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.or3(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.and3(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.or4(altvital)
# Loading D:/cad/Modeltech_ae/win32aloem/./altera/vhdl/alt_vtl.and4(altvital)
VSIM 5> |
Now: 0 ns Delta: 0 | sim:/cnttb

```

Figure 31 Design 을 Load 한 결과 Windows

상기 그림과 같은 결과가 나오는지 확인한다.

Timing Simualtion 결과 확인하기.

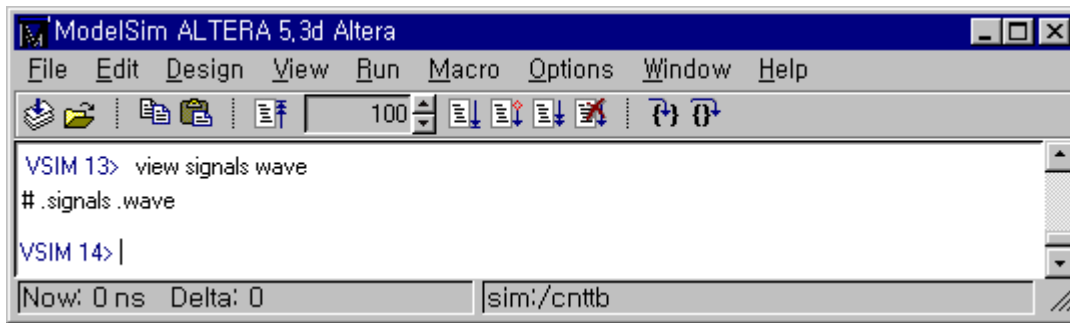


Figure 32 Signals, Wave Window 부르기

위의 그림처럼 Transcript Window 에서 View signals wave 라는 명령을 이용하여 Signals 와 Wave Window 만 화면에 나타나게 할 수 있다.

Signal Window 의 Menu 에서 View → Wave → Signals in Region 을 선택하여 현재 Signal Window 에 있는 신호들을 Wave Window 에 표시 시킨다.

Transcript Window 에서 run 1000 ns 명령을 주어 1000 ns 까지 Timing Simulation 을 수행한다.

Wave Window 에서 CLK to Out Delay (Tcko)을 확인해 본다.

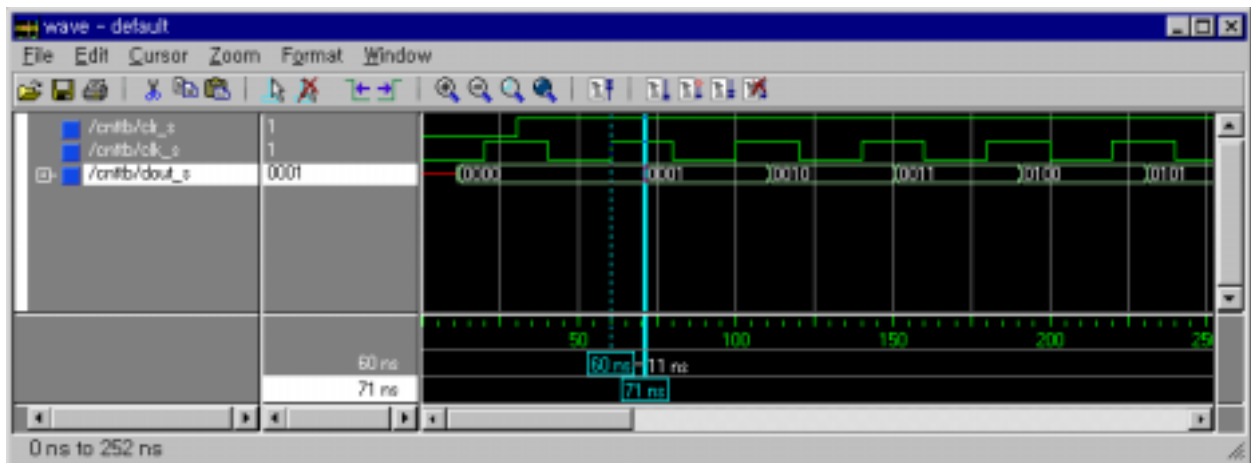


Figure 33 Timing simulation Result Window

Revision History

- 2000-7-10 - Ver 1.0: Initialize Release...
- 2000-8-6 – Ver 1.1 Text Modified by C.W.Yang