



**BASIC TRAINING**

**F1.5i Day 1 Labs**



**Hyun Myung Electronics Co., Ltd.**  
Tel 02-3141-0147 / Fax 02-3141-0149  
<http://www.hmelec.co.kr>

***X<sub>S</sub> A<sub>T</sub> C<sub>E</sub> T<sub>P</sub>***

# Contents

소개 (INTRODUCTION).....	3
프로젝트 만들기 (CREATING A NEW PROJECT).....	3
SCHEMATIC 시작하기 (STARTING SCHEMATIC EDITOR).....	4
심볼 배치하기 (PLACING SYMBOLS).....	5
선 연결하기 (DRAWING WIRES).....	7
네트 이름 주기 (NAMING WIRES).....	8
핀 위치 지정하기 (ADDING PIN LOCATIONS).....	9
SCHEMATIC의 변경 (SCHEMATIC CHANGES).....	10
기존 PROJECT의 LIBRARY를 불러쓰기.....	10
VIEWLOGIC 데이터 받아들이기 (IMPORTING A VIEWLOGIC MACRO SCHEMATIC).....	11
네트이름 반복하기 (REPEATING NET NAMES).....	13
새 심볼 만들기 (CREATING A NEW SYMBOL).....	14
VHDL MACRO 편집하기 (EDITING A VHDL MACRO).....	16
VHDL 오류 찾기 (FINDING VHDL ERRORS).....	19
회로합성 하기 (RUNNING SYNTHESIS).....	20
SCHEMATIC MACRO 만들기 (CREATING A SCHEMATIC MACRO).....	21
MACRO SCHEMATIC 저장하기 (SAVING THE MACRO SCHEMATIC).....	25
상위 SCHEMATIC 끝내기 (FINISHING THE TOP LEVEL SCHEMATIC).....	26
기능적 시뮬레이션 (FUNCTIONAL SIMULATION).....	27
시뮬레이션 데이터 만들기 (CREATING TEST VECTORS).....	28
시뮬레이션 실행하기 (RUNNING SIMULATION).....	30
SCHEMATIC에서 결과 보기 (VIEWING RESULTS ON THE SCHEMATIC).....	31
TIMESPEC SYMBOL 첨가하기 (ADDING TIMESPEC SYMBOL).....	31
XACT 실행하기 (RUNNING XACT).....	33
TIMING SIMULATION 하기 (TIMING SIMULATION).....	36
ROM FILE 만들기 (CREATING A MCS DATE FOR PROM).....	38
후기.....	41

## 소개 (Introduction)

본 실습교재에는 Foundation Series 1.5를 사용하여 XACTstep M1을 지원하는 프로젝트를 만드는 방법이 설명됩니다. 설계 단계별로 중요한 부분들의 효율적인 설명을 위해서, 여기서는 실제로 하나의 프로젝트를 직접 만들어 갈 것입니다.

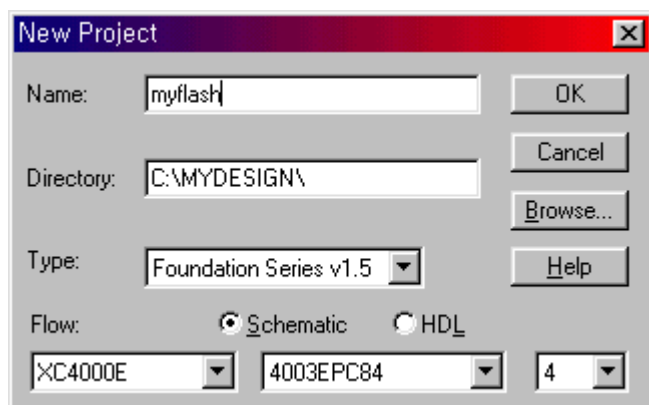
여기에서 주어지는 각 단계별 명령들을 잘 따라 한다면, 예제로 주어지는 FLASH design 이라고 정의한 프로젝트를 만들 수 있습니다.

예제 프로젝트에 주어지는 Chip의 Pin locations을 적용하시면 XILINX 4000 DEMO board에서 직접 설계된 회로를 검증하실 수 있습니다.

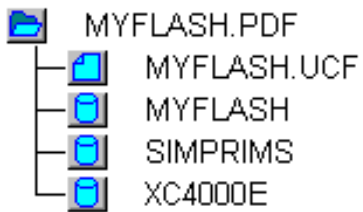
다음 장부터는 편의상 존칭은 생략합니다.

## 프로젝트 만들기 (Creating a New Project)

1. 바탕화면에 위치한 **Foundation Series** 아이콘을 더블 클릭하여 **Project Manager**를 시작한다.
2. **Project Manager**의 **File** 메뉴 중 **New Project.....**를 선택한다.
3. **New Project....** 창에서 **Name**항목에는 MYFLASH라고 기입하고, **Directory**는 사용자가 원하는 경로를 선택하면 된다. 여기서는 mydesign directory를 선택한다. **Type**항목은 **Foundation Series v1.5**을, **Family**항목은 **XC4000**, **Part**항목은 **4003EPC84**, 그리고 **Speed**항목은 **4**를 선택한다. **Design Flow**는 **Schematic**을 선택한다. 설정을 마친 후 **OK** 버튼을 클릭하면 새로운 프로젝트가 생성된다.



4. 새로운 프로젝트가 만들어 졌으면 아래그림에 나오는 것처럼 **Hierarchy Browser**에 라이브러리들이 생성이 될 것이다.



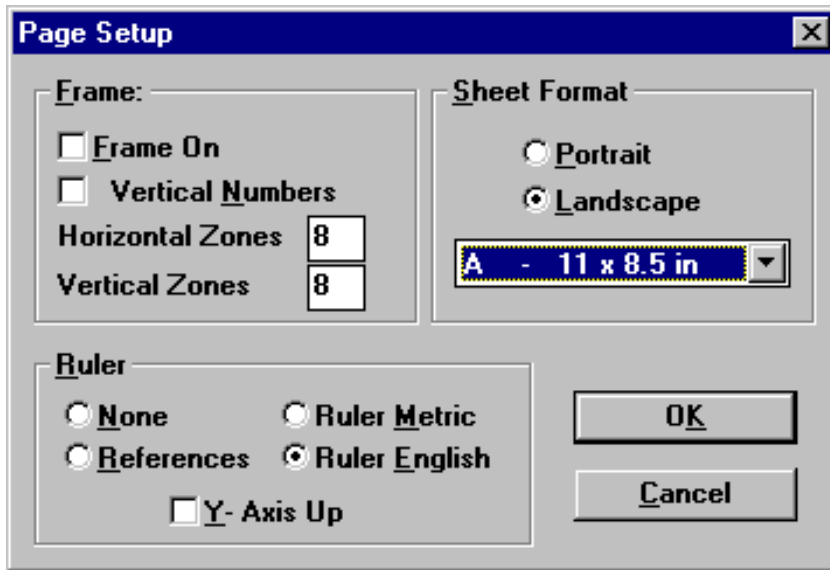
- **MYFLASH**는 프로젝트 라이브러리이다. 이 라이브러리는 현재 프로젝트에서 User가 작성하는 모든 Macro와 심볼들의 정보를 저장하는데 사용한다.
- **X4000E**는 XC4000E devices의 사용 시 필요한 시스템 라이브러리이다.
- **SIMPRIMS**는 Post - layout Netlists의 Simulation(Timing Simulation)에 필요한 시스템 라이브러리이다.

## Schematic 시작하기 (Starting Schematic Editor)


1. **Project Manager**상의 Flow diagram에 있는 Design Entry안의 and symbol icon 버튼을 클릭한다.

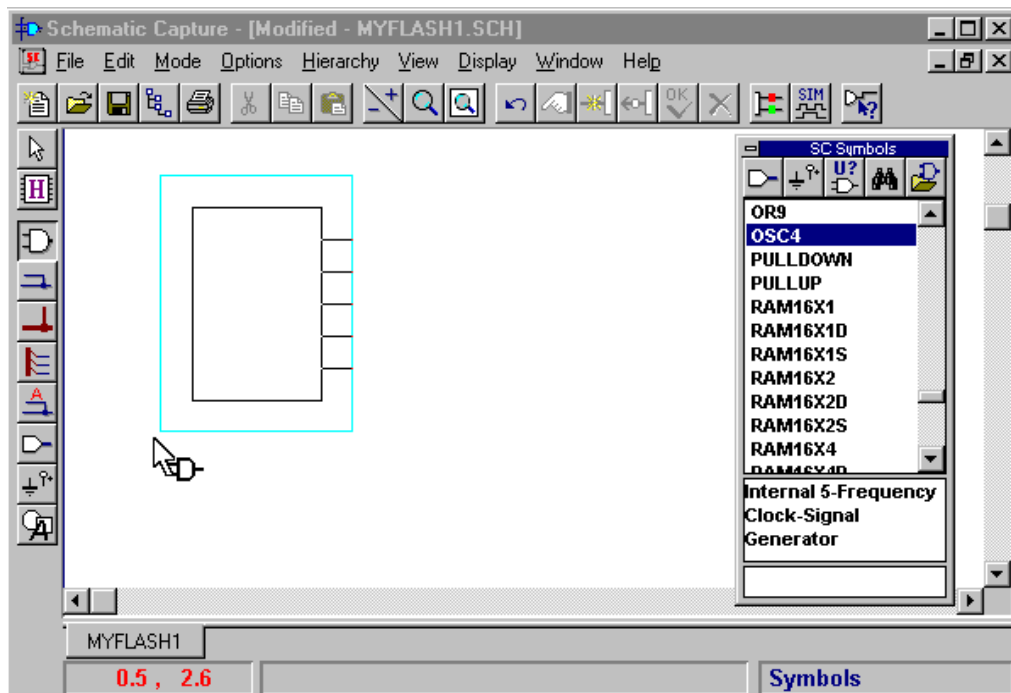


2. Schematic Editor에 MYFLASH1.SCH라는 이름으로 새로운 Schematic sheet가 열린다.
3. **File** 메뉴에 있는 **Page Setup**명령을 선택한다.
4. **Page Setup** 창이 화면에 나타나면, 그 중 **Sheet Format**부분을 Landscape, size A 11 x 8.5 in으로 선택한다.

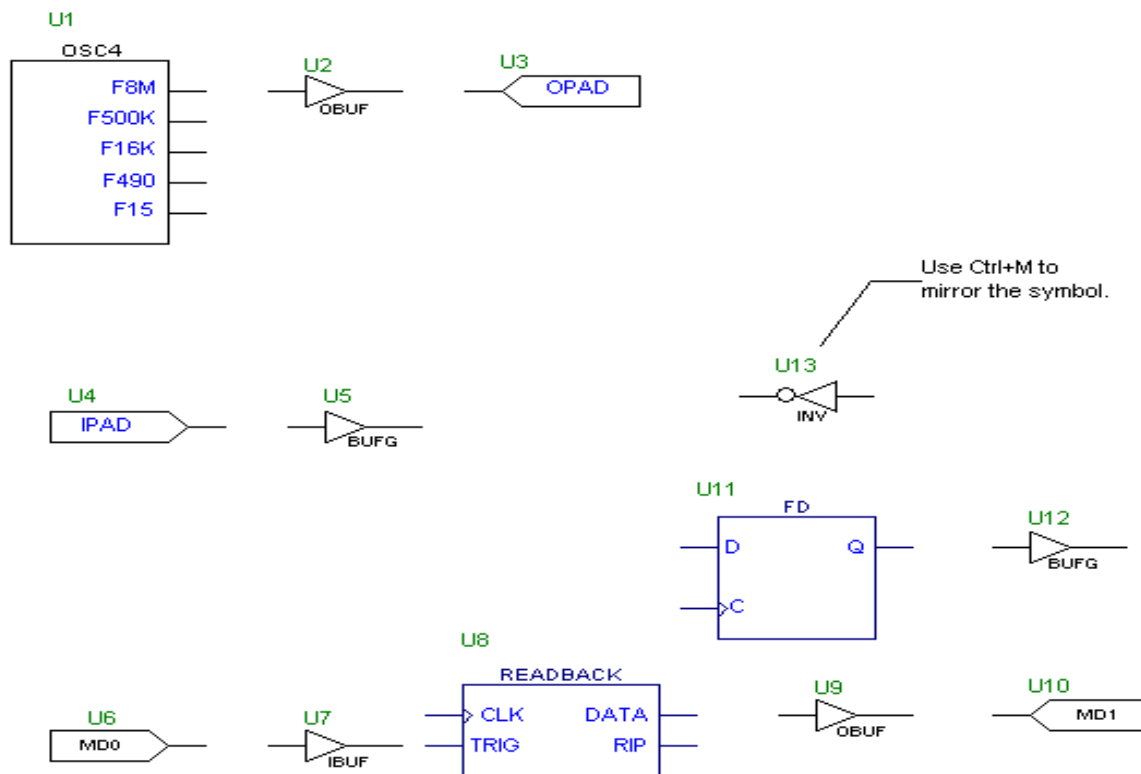


## 심볼 배치하기 (Placing Symbols)



1. 왼쪽에 위치한 Schematic 툴 바 중  버튼을 클릭한다.
2. 화면에 나타난 심볼 toolbox의 아래 입력란에 OSC4라고 입력한다.
3. 왼쪽으로 Cursor를 이동하면 Cursor를 따라서 선택한 심볼의 외형이 나타난다.




4. **Mouse**의 왼쪽버튼을 클릭하면 현 위치에 심볼이 배치된다.
5. 위의 순서를 반복하여 아래 그림처럼 각 심볼들을 배치한다:




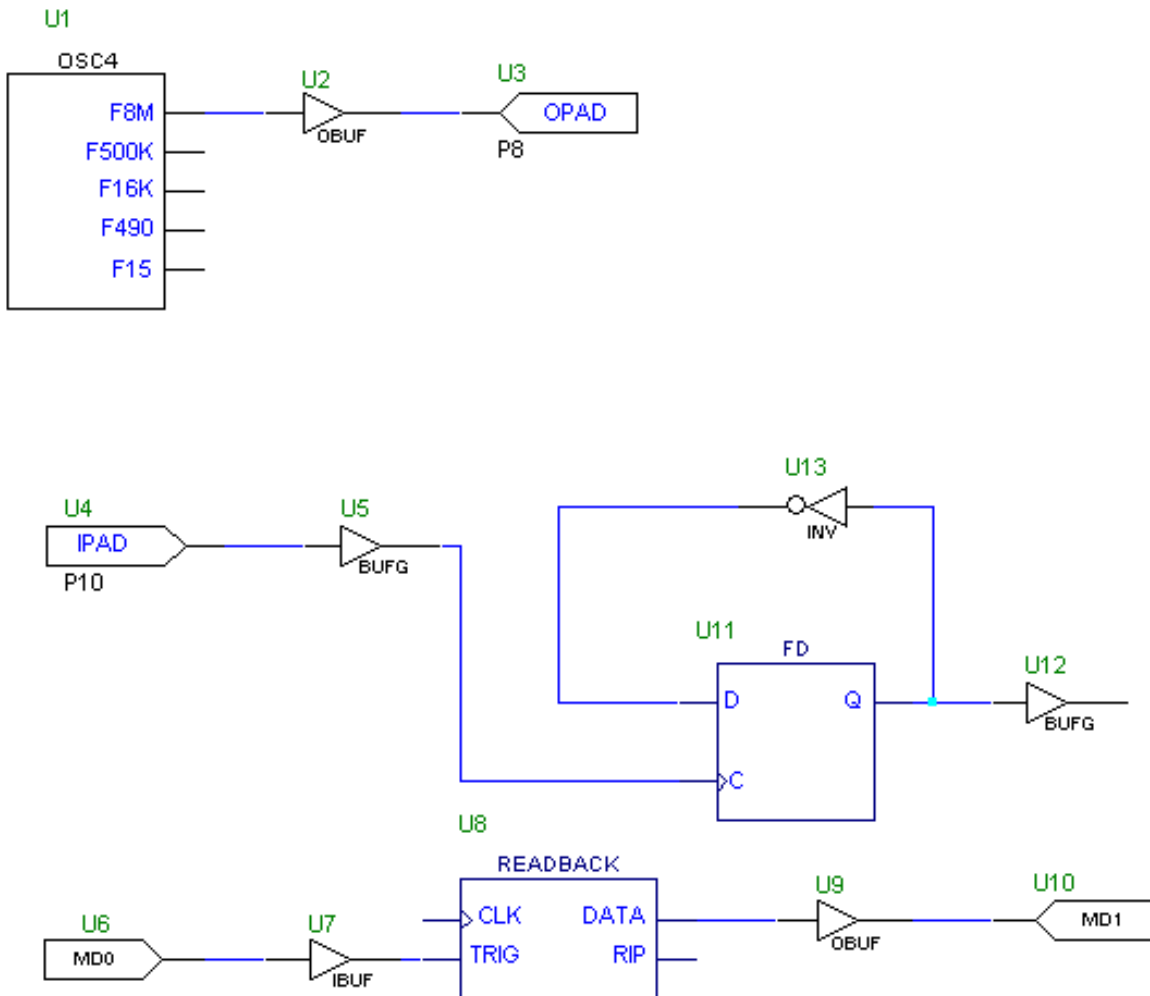
**NOTES:**

- 심볼을 배치하기 전에 **Ctrl+M** 키를 누르면 심볼의 좌우를 바꿀 수 있는데, 여기에서는 **INV** 심볼에다가 적용을 해보자. (**Ctrl+L** 키는 심볼의 회전에 사용한다.)
- 만약 심볼을 잘못 선택했다면, **Esc** 키를 눌러서 마우스 포인터에 붙어있는 잘못된 심볼을 없애버릴 수 있다.
- 임의의 심볼이 선택된 상태에서 포인터가 화면상의 경계에 가까이 다가가면 **schematic page**는 자동으로 스크롤이 된다.
- 전체 페이지를 보려면 **PgDn** 키를 누르고, 바로 전 상태로 돌아가려면 **PgUp** 키를 누른다. 이 키들은 키보드의 숫자키 부분에 있다.
- 잘못하여 틀린 심볼을 배치했다면, **Del** 키를 눌러서 삭제할 수 있다.
- 심볼 **toolbox**를 닫으려면, 왼쪽 편 툴 바에 있는  버튼을 누른다.
- **Schematic**의 특정부분을 확대하려면, 상단 툴 바에 있는  버튼을 누른다. 그런 다음 포인터로 확대할 영역을 사각형모양으로 드래그한다. 선택이 완료되면 눌러진 마우스 버튼에서 손을 떼다.

- 전체화면으로 돌아가려면, 상단 툴 바에 있는  버튼을 누른다.  
 현화면의 **Refresh**는, **F10** 키를 누른다.


## 선 연결하기 (Drawing Wires)

1. 선을 그리려면, 왼쪽 툴 바에 있는  버튼을 누른다.
2. 다음에는, 선이 시작될 핀 위에서 클릭을 하고 선이 종료될 다른 핀 위에서 클릭을 한다.
3. 기존에 그려진 선의 끝을 클릭하거나 다른 선을 시작하면, 자동적으로 junction(접합점)이 생성된다.
4. 위에서 설명한 순서대로 아래에 보이는 그림처럼 선들을 그린다.




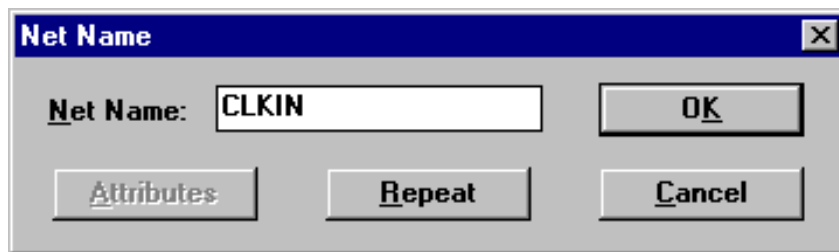
**NOTES:**

- 선 그리기 모드에서 빠져나가려면 **Esc** 키를 누른다.
- 선을 지우려면 **Esc** 키를 누르고, 삭제할 선을 클릭하고 **Del** 키를 누른다.

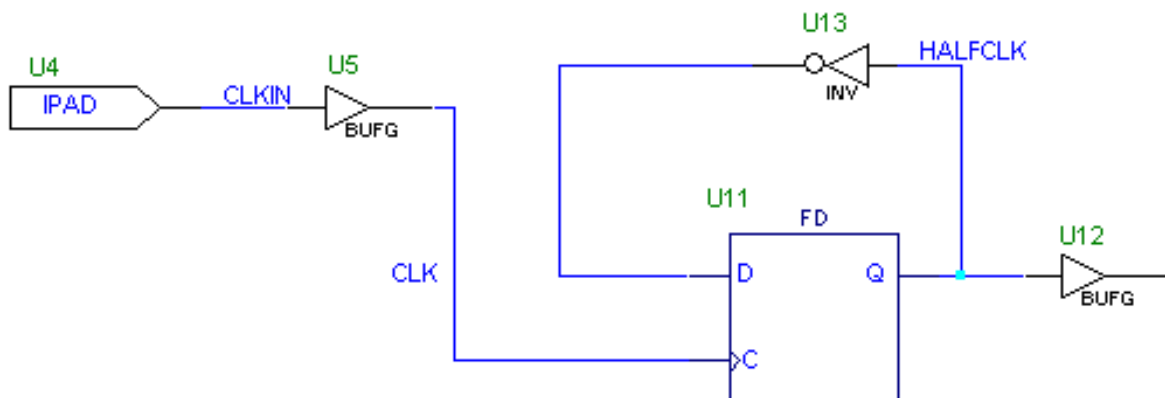
- Schematic을 저장하려면 상단 툴 바에 있는  버튼을 누른다.

**네트 이름 주기 (Naming Wires)**

1. 네트에 이름을 부여하려면, 왼쪽 툴 바에 있는  버튼을 누른다.



2. **Net Name** 창에 네트이름을 입력하고 **OK** 버튼을 누른다.
3. 다음으로, 네트이름이 위치할 네트로 커서를 이동하고 클릭한다.
4. 녹색으로 네트이름이 표시된다면 이름이 선에 대응되지 않은 것이다.
5. 다음에 나오는 선들에 네트이름을 부여한다:
  - CLKIN
  - CLK
  - HALFCLK



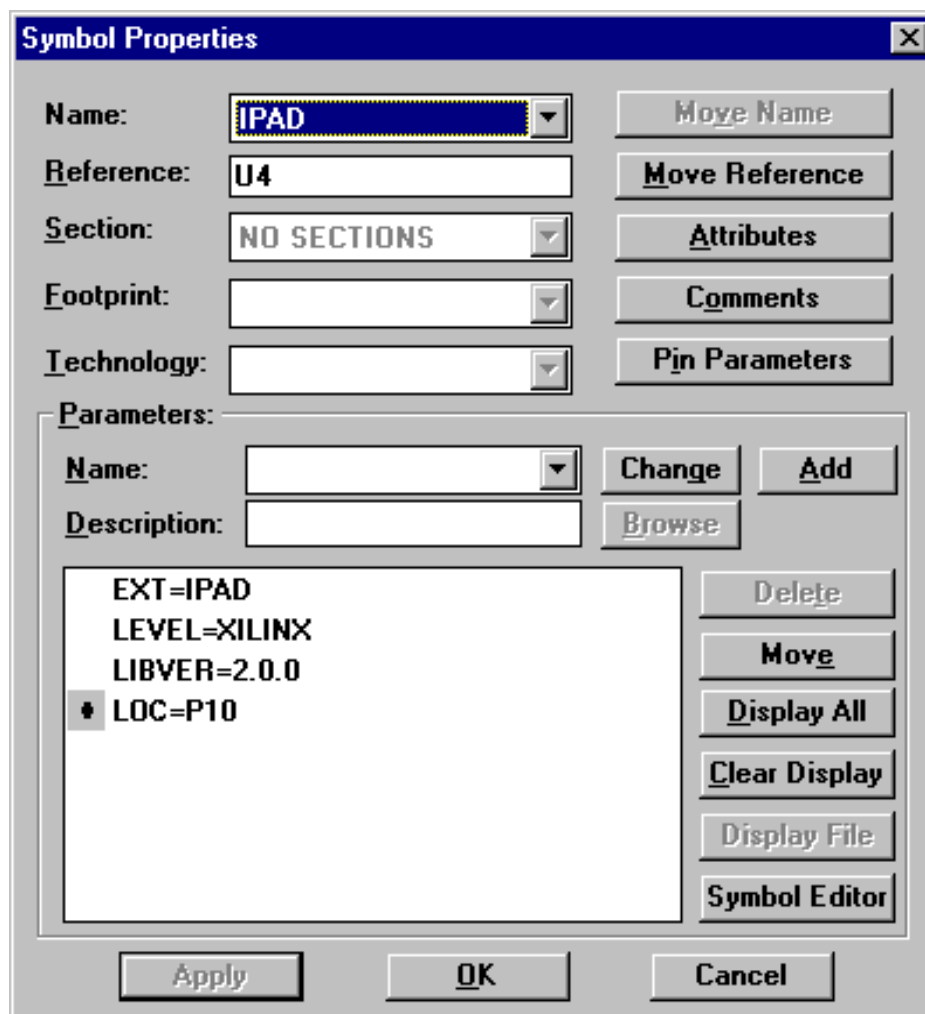


## 핀 위치 지정하기 (Adding Pin Locations)

Xilinx Demonstration Board에 이 회로를 적용하고 싶다면, Schematic 상에 정의할 수 있는 핀 번호 지정이 필요하다. 핀 번호를 정의하려면 다음의 순서를 따른다:


1. 선택모드로 이동하기 위해서 **Esc** 키를 누른다.
2. U4로 이름이 붙여진 IPAD 심볼을 더블 클릭한다.
3. Name:란에 LOC를 Description:란에 P10 이라고 입력한다. (Parameters section).
4. 아래쪽 목록에 추가 시키기 위해 **Add** 버튼을 누른다.

**NOTE:** Your reference designators may be different on your screen or may be invisible. Please modify the appropriate component.





5. **OK** 버튼을 누른다.
6. IPAD 심볼 아래에 P10 parameter가 보일 것이다.
7. 같은 순서로 OPAD 심볼 U3에 LOC=P8 parameter를 추가한다.

## Schematic의 변경 (Schematic Changes)


모든 회로의 변경은 **select mode**에서 실행된다. 이 mode는 왼쪽 툴 바에 있는  버튼이 눌러져 있을 때 활성화된다. 이 **select mode**로 빨리 되돌아 가는 방법은 **Esc** 키를 누르면 된다.

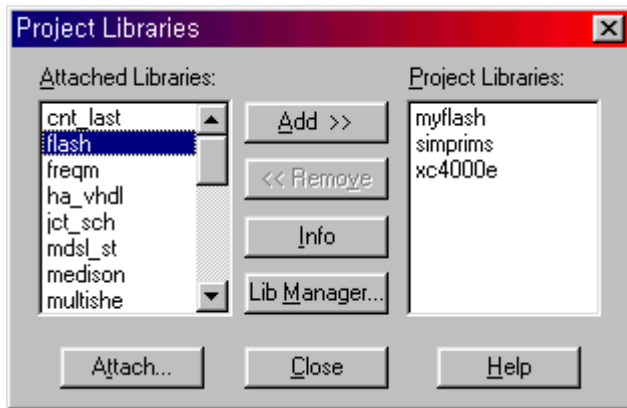
다음의 항목들은 편집 작업의 수행 시 공통적으로 쓰이는 방법들이다:

- 심볼을 이동하려면, 해당 심볼을 클릭하고 원하는 위치까지 드래그하면 된다.
- 회로의 일 부분을 이동할 때는 원하는 부분의 주위를 **box**로 감싸고, 선택된 심볼 중에 하나를 클릭하여 원하는 위치로 드래그하면 선택된 모든 심볼들이 모두 이동한다.
- 선택된 그룹에서 임의의 개체를 추가 삭제하려면 **Shift** 키를 누르고 그 개체를 클릭한다.
- 심볼에 연결된 선들을 그대로 유지한 채 심볼만을 지우려면, 지울 심볼을 선택하고 상단 툴 바에 있는  버튼을 누른다. 그리고 나서 **Del** 키를 누른다. 심볼이 없어진 선은 **Hanging wire**로 된다.
- **Hanging wire**에 심볼의 핀을 연결하려면, **schematic** 툴바에 있는  버튼을 누르거나, 핀을 선의 끝으로 드래그한 후 마우스로 버튼을 다시 눌러주면 된다.

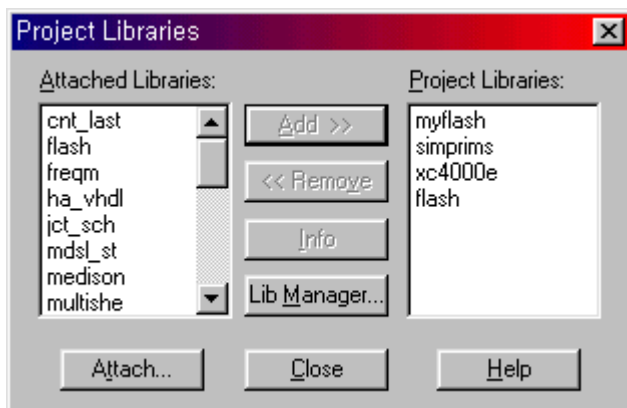
## 기존 Project의 Library를 불러쓰기

이번 장에서는 기존에 만들어 둔, 또는 작업이 끝난 **project**을 이용하여 현재 **project**에서 필요한 **component**을 선택하여 사용하는 방법을 알아 보기로 한다. 크게 두 가지 방법이 있으며 여기서는 기존의 **project**을 **library**화 하여 사용하는 **attaching the project**를 설명한다. 다른 한 가지는 **library manager**를 사용하여 필요한 **component**를 현재의 **project**에 **copy**하여 사용하는 방법이다..

1. **Project Manager** 의 상위 툴 바 버튼에서 **Project libraries icon button**  을 누른다.
2. 아래 대화상자가 나타나면 대화 상자의 내용을 아래와 같이 설정한다.



3. Attached Libraries는 기존에 작업이 된 project를 의미하며 Project Libraries는 현재 project가 참조하고 있는 library를 뜻하고 있다. flash를 선택한 후 Add >> button을 누른다.
4. 아래 그림과 같이 오른쪽에 flash가 첨가되면 성공적으로 flash project가 attach 된 것이다.



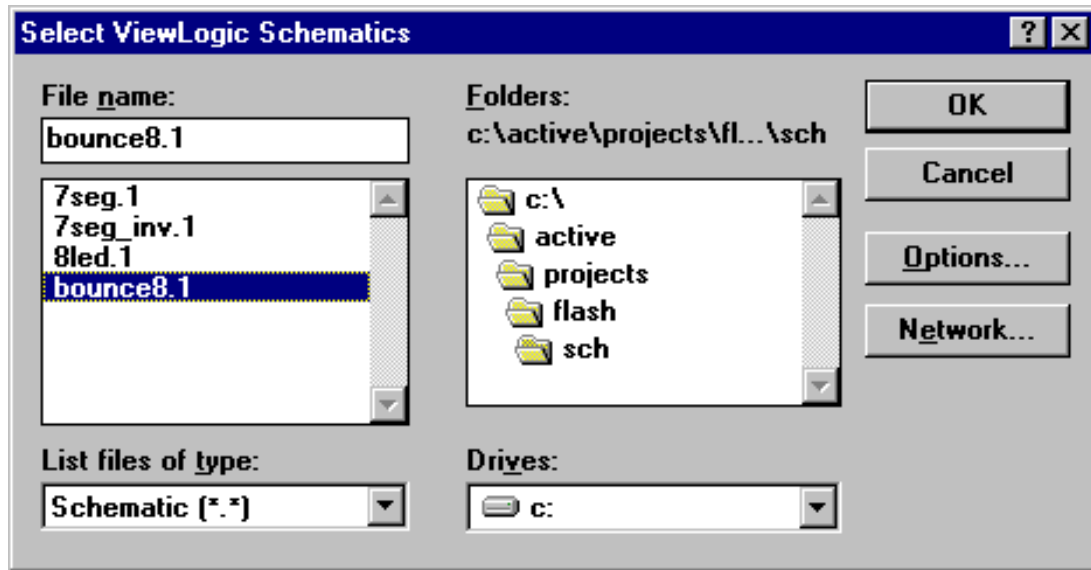
5. Close button을 눌러 Project Libraries 대화 창을 빠져 나온다.

## Viewlogic 데이터 받아들이기 (Importing a Viewlogic Macro Schematic)

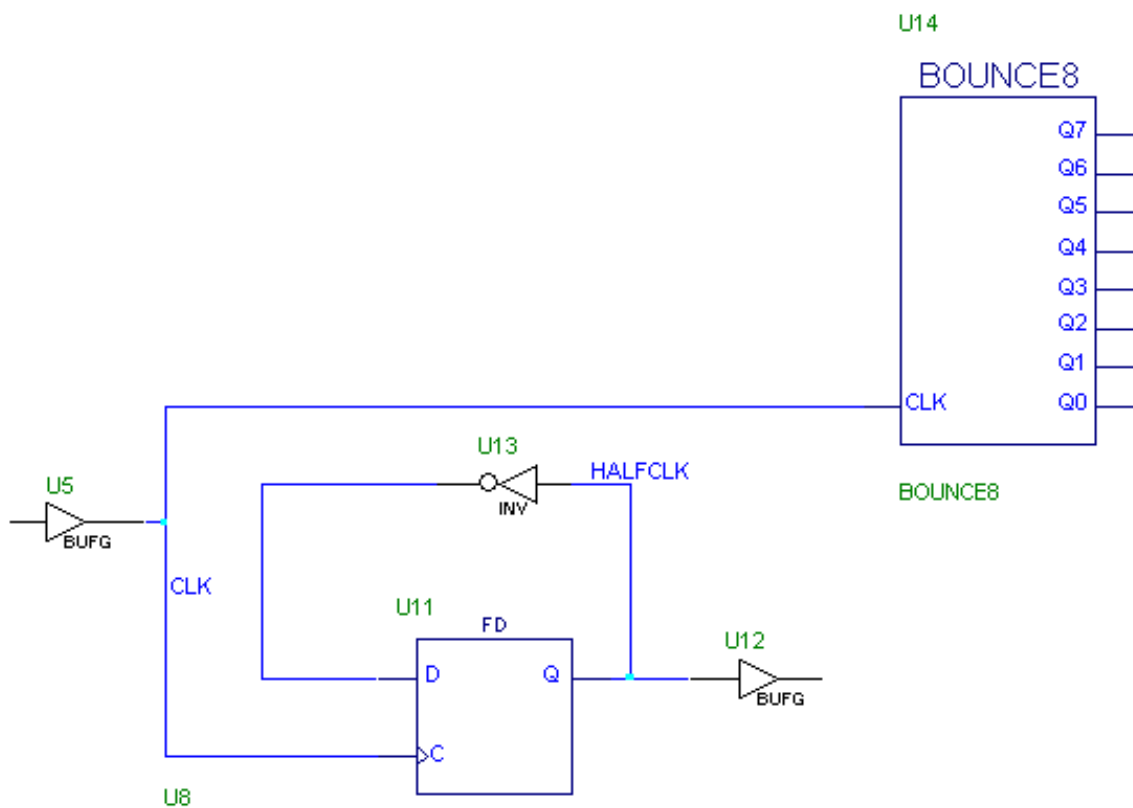
*이 예제는 선택 사항이므로 Foundation Series 1.5로 실습을 할 때는 생략하기로 하며, 필요한 component를 c:\fndtn\active\projects\flash에 있는 것을 copy하여 사용한다.*

Viewlogic schematic을 Foundation Schematic Capture로 Import하는 예제로, 현 회로에 Viewlogic schematic macro중 하나를 받아들여 본다.

1. **File** 메뉴에 있는 **Import Viewlogic Schematic** 명령을 선택한다.
2. **List Files of Type** 필드에서 Schematic (\*.\*)을 선택하고 (**Select Viewlogic Schematics** 창 내에서), **File Name** 필드에서 C:\ACTIVE\PROJECTS\FLASH\SCH\BOUNCE8.1 schematic을 선택한다.



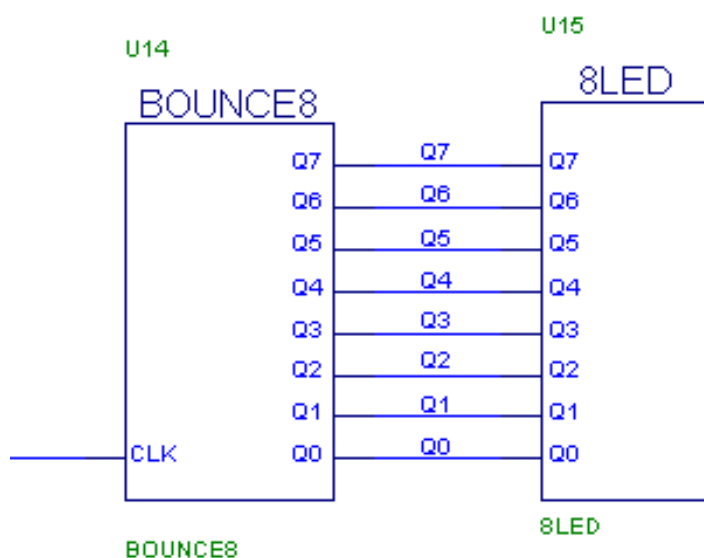
3. **OK** 버튼을 누르면 Viewlogic schematic을 받아들이기 시작한다.
4. BOUNCE8 schematic macro가 MYFLASH 프로젝트 라이브러리에 추가가 될 것이다.
5. 결과리포트를 보길 원하냐는 내용의 대화상자가 나타나면, **NO** 버튼을 선택한다.
6. **File** 메뉴의 **Update Libraries**를 실행하면 받아들여진 심볼이 라이브러리에 보태어진 것을 확인할 수 있다.
7. 받아들인 macro를 다음에 보이는 회로처럼 배치한다.




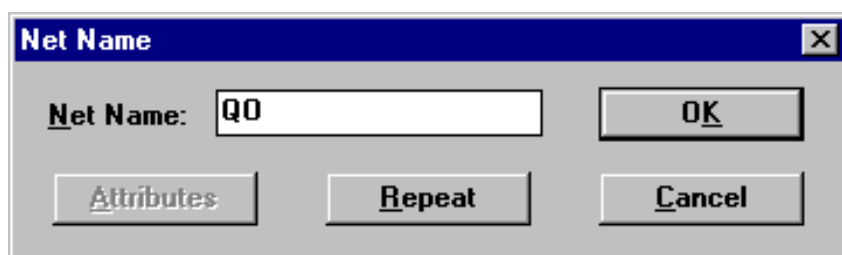
8. 동일한 방법으로 같은 디렉토리에 있는 7SEG.1와 8LED.1을 import한다. **Ctrl** 키를 사용하면 한번에 여러 개의 schematic을 선택할 수 있다.

## 네트이름 반복하기 (Repeating Net Names)

1. 아래의 그림처럼 8LED 심볼을 배치하고 선을 연결한다.



2. 네트이름을 추가하려면, 왼쪽 툴 바의  버튼을 선택하고 Q0 를 입력한다.

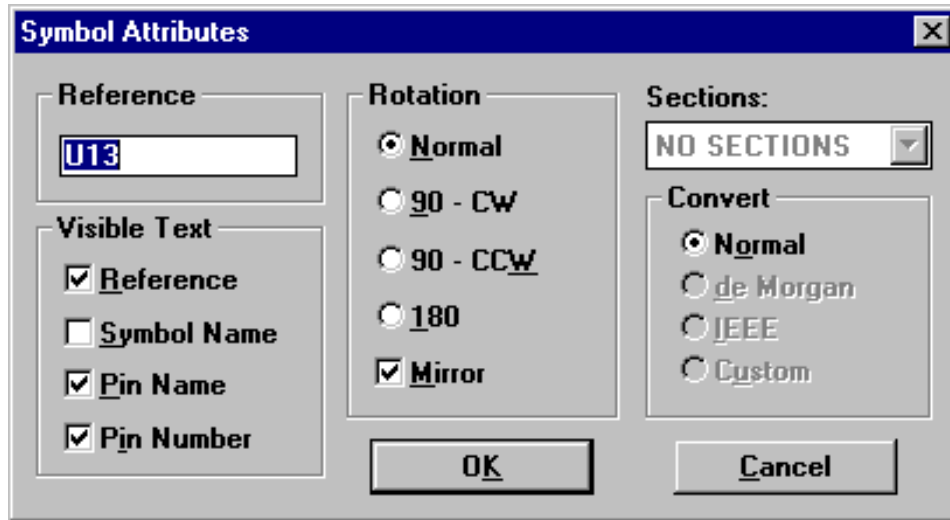


3. 동일한 성격의 선 이름을 빠르게 주기위해 **Repeat** 버튼을 누른다.
4. 프로그램은 반복모드로 설정이 된다. 매번 선을 클릭할 때마다 새로운 이름이 하나씩 증가하면서 배치가 된다. 이런 방법으로 8개의 선 이름을 추가한다. (이때 주의할 사항으로서 선 위를 정확히 클릭하여 네트이름과 네트의 색이 동일하도록 해주어야 한다. 만약 네트와 네트이름의 색이 다를 경우 네트이름은 라벨로 처리가 되었음을 의미한다.)

**NOTE:** 심볼명을 나타내기 위해 **Esc**를 누르고, 심볼을 더블 클릭한다. 그리고 **Symbol Properties**

창의 **Attributes** 버튼을 선택한다. **Symbol Attributes** 창 내 **Visible Text** 부분의 **Symbol Name** 을 선택하고 **OK** 를 누른다.

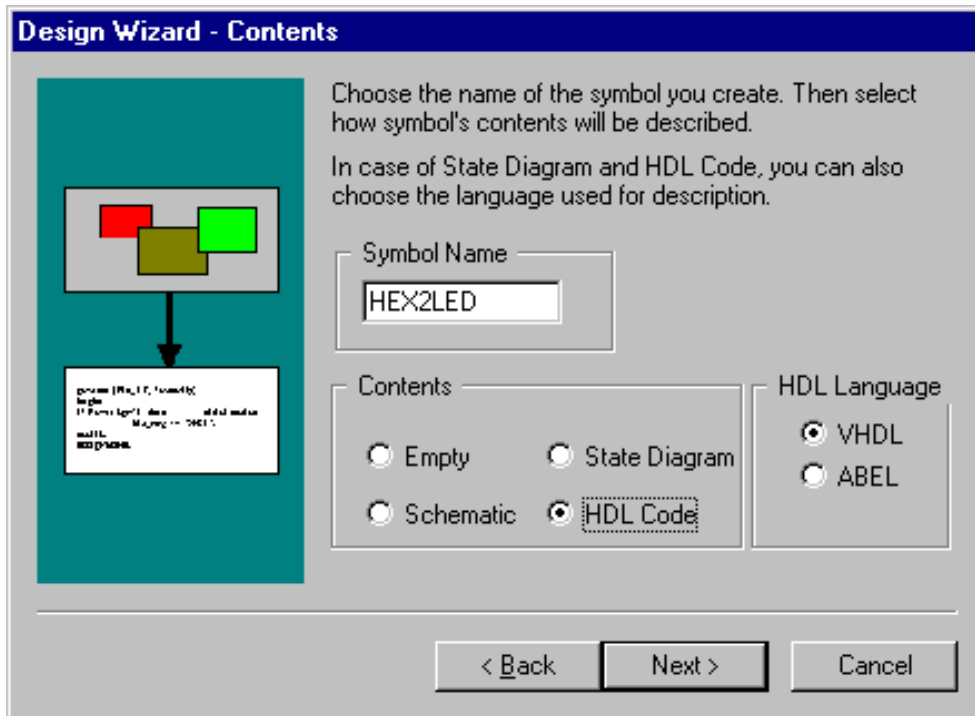
다음으로 **Symbol Properties** 창에 있는 **OK** 버튼을 누르면 심볼명이 표시된다.



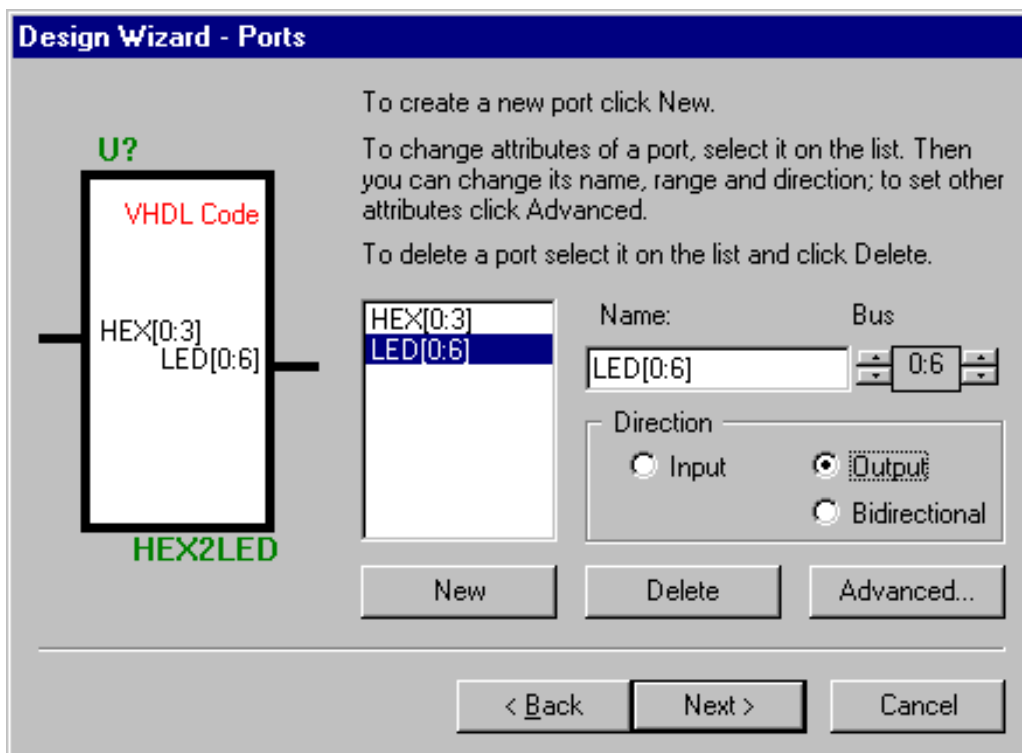
## 새 심볼 만들기 (Creating a New Symbol)

VHDL을 사용하여 만들 예제로서 HEX Data 값을 7Segment LED 출력 값으로 변환하는 Decoder를 만들 것이다.

1. **File** 메뉴의 **New Sheet** 명령을 선택하여 새로운 schematic을 생성한다.
2. **Hierarchy** 메뉴의 **New Symbol Wizard** 을 선택한다.
3. **Design Wizard** 창의 환영 메시지가 나타나면 **Next>** 를 클릭한다. 그러면 **Design Wizard - Contents** 창이 나타날 것이다.

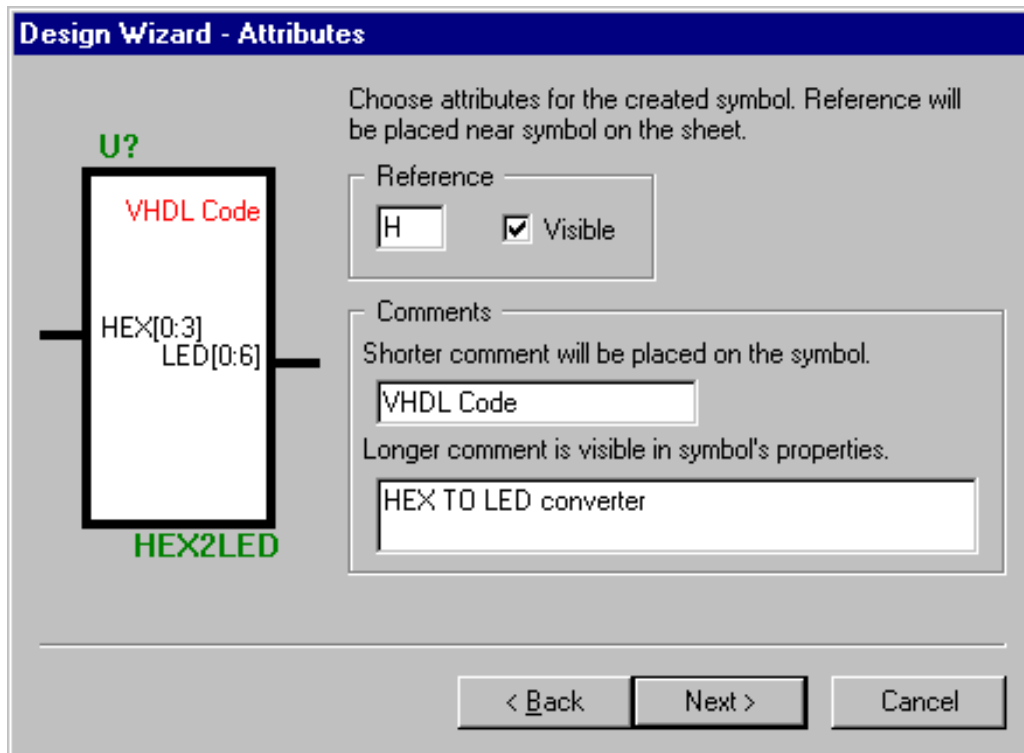


4. **Symbol Name** box에 HEX2LED라고 입력한다.
5. **Contents** 필드에서 다음의 내용을 선택한다:
  - Contents: HDL Code
  - HDL Language: VHDL.
6. **Next>** 버튼을 누르면, **Design Wizard - Ports** 창이 나타날 것이다.




7. **Design Wizard - Ports** 창에서

- **New** 버튼을 누른다.
- **Name:** 필드에 HEX[0:3]을 입력한다.
- **Direction** 필드에서 Input을 선택한다.
- 다시 **New** 버튼을 누른다.
- **Name:** 필드에 LED[0:6]을 입력한다.
- **Direction** 필드에서 Output을 선택한다.
- **Next>** 버튼을 누르면, **Design Wizard - Attributes** 창이 나타난다.



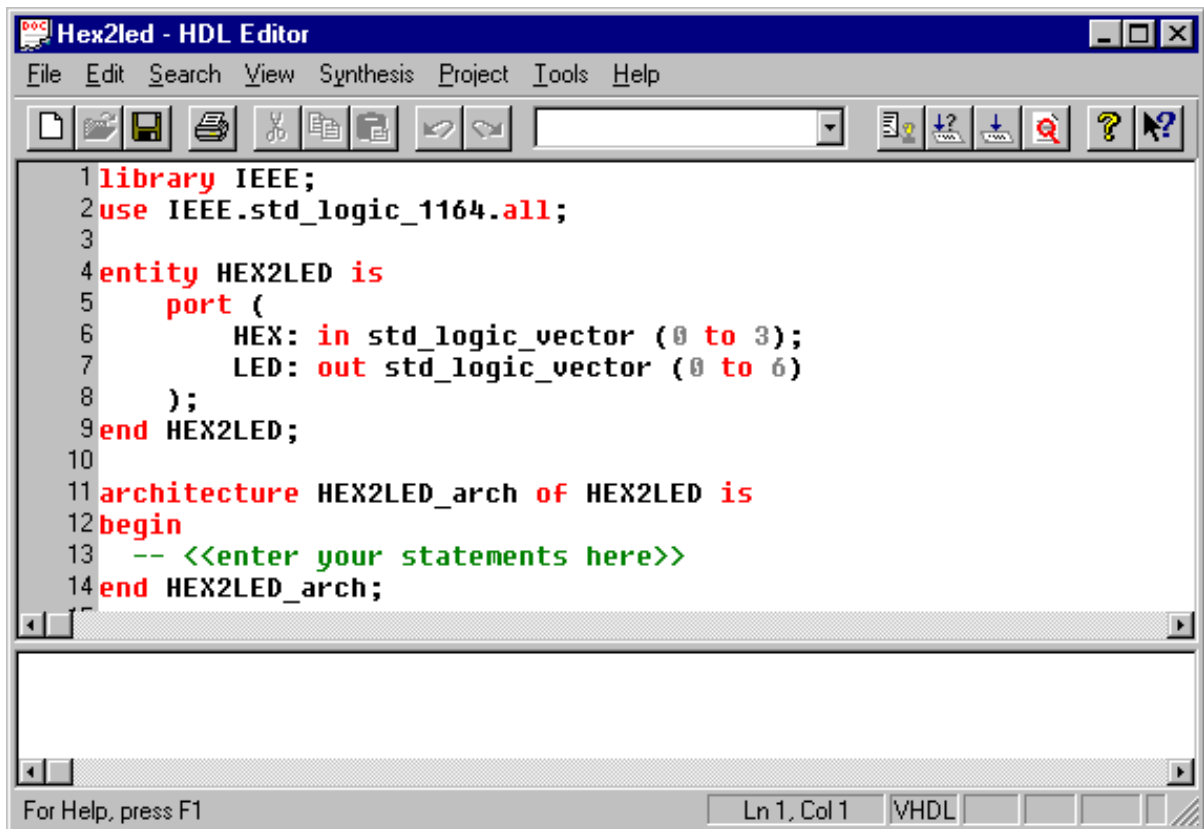
8. **Design Wizard - Attributes** 창에서 Longer comment~ 란에 "HEX to LED converter"라고 입력한다.
9. 계속하기 위해 **Next**를 누르면 심볼이 생성된다. 이 새 심볼은 프로젝트 라이브러리에 배치된다.
10. **Finish**를 누르면 심볼이 생성이 되는데, 만들어진 **Macro**는 프로젝트 라이브러리에 추가된다.
11. HEX2LED 심볼을 새 도면에 배치한다 (MYFLASH2.SCH).

## VHDL MACRO 편집하기 (Editing a VHDL Macro)

1. 왼쪽 툴 바에서  버튼을 선택한다.
2. Schematic 상에서 HEX2LED 심볼을 두 번 클릭 한다.



3. HDL Editor가 HEX2LED 매크로의 port 정의를 보여 주는 아래 그림과 같은 윈도우를 열어 준다.

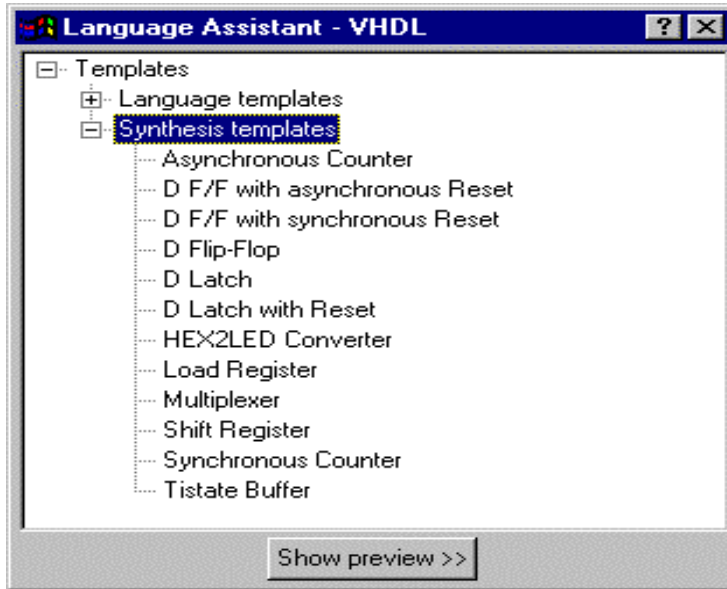


```

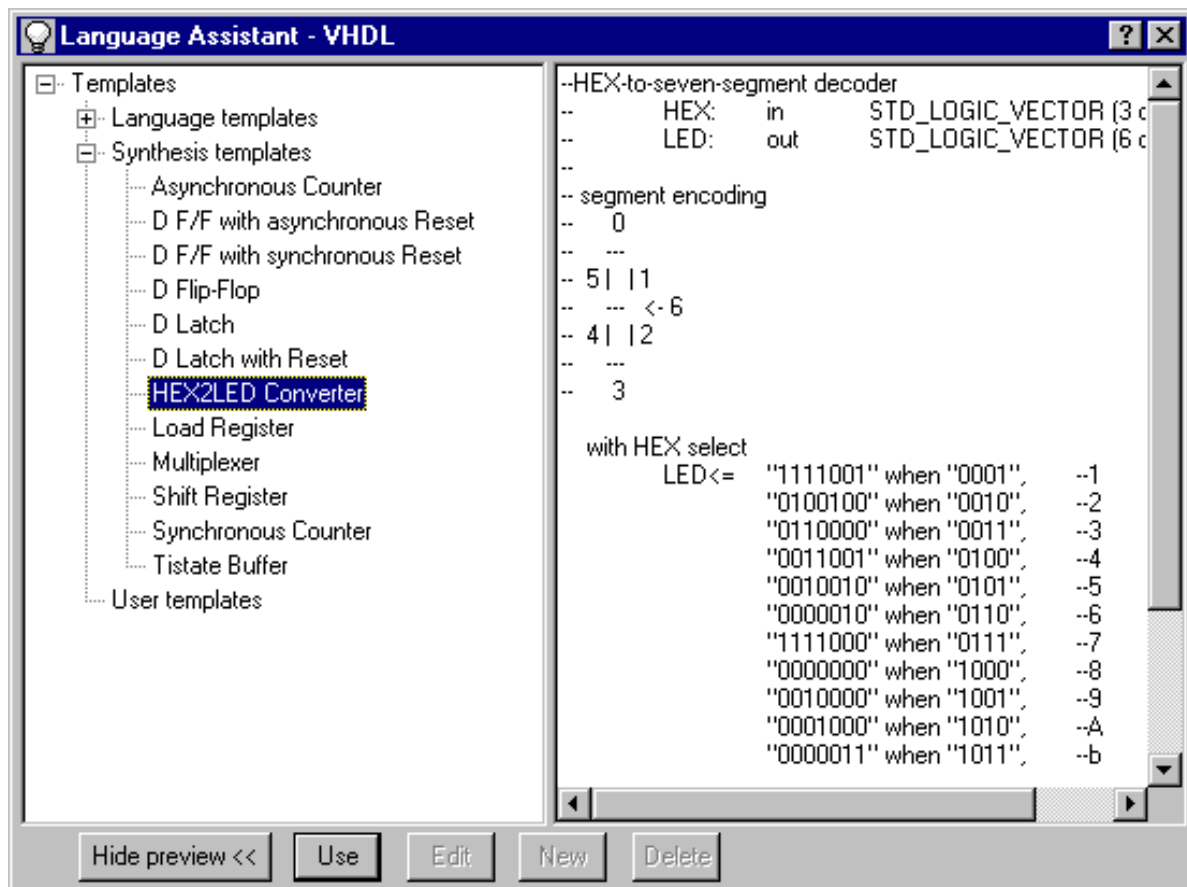
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity HEX2LED is
5     port (
6         HEX: in std_logic_vector (0 to 3);
7         LED: out std_logic_vector (0 to 6)
8     );
9 end HEX2LED;
10
11 architecture HEX2LED_arch of HEX2LED is
12 begin
13     -- <<enter your statements here>>
14 end HEX2LED_arch;

```

4. **entity** section위에 있는 모든 line들을 지운다.
5. **port** section 안에 있는 선언들을 아래와 같이 변경한다.  
 HEX: in BIT\_VECTOR (3 downto 0);  
 LED: out BIT\_VECTOR (6 downto 0)  
 bit 0 는 LSB(Least Significant Bit)를 의미한다.
6. <<enter your statements here>> line을 지운다.
7. Architecture section의 begin 바로 아래 라인에 커서를 위치한다. 그리고 나서 **Tools** 메뉴에서 **Language Assistant**를 선택한다.



8. **Synthesis Templates**를 선택하고 **Select Template** 목록에서 HEX2LED Converter를 찾는다.
9. **Show preview>>** 버튼을 누른다. template의 문장들이 **Language Assistant** 창의 오른쪽 패널에 보여질 것이다.



10. Template을 복사하기 위해 **Use** 버튼을 누른다.

11. 아래 보여진 code를 만들기 위해 필요한 변경 작업들을 수행하라.

### Final Version of HEX 2LED Macro Code:

```

entity HEX2LED is
  port (
    HEX: in BIT_VECTOR (3 downto 0);
    LED: out BIT_VECTOR (6 downto 0)
  );
begin
end HEX2LED ;

architecture HEX2LED_arch of HEX2LED is
begin
  --HEX-to-seven-segment decoder
  -- HEX:   in   STD_LOGIC_VECTOR (3 downto 0);
  -- LED:   out  STD_LOGIC_VECTOR (6 downto 0);
  --
  -- segment encoding
  --   0
  --   ---
  -- 5 |   | 1
  --   ---   <- 6
  -- 4 |   | 2
  --   ---
  --   3

  with HEX select
    LED<= "1111001" when "0001", --1
          "0100100" when "0010", --2
          "0110000" when "0011", --3
          "0011001" when "0100", --4
          "0010010" when "0101", --5
          "0000010" when "0110", --6
          "1111000" when "0111", --7
          "0000000" when "1000", --8
          "0010000" when "1001", --9
          "0001000" when "1010", --A
          "0000011" when "1011", --b
          "1000110" when "1100", --C
          "0100001" when "1101", --d
          "0000110" when "1110", --E
          "0001110" when "1111", --F
          "1000000" when others; --0

end HEX2LED_arch;

```

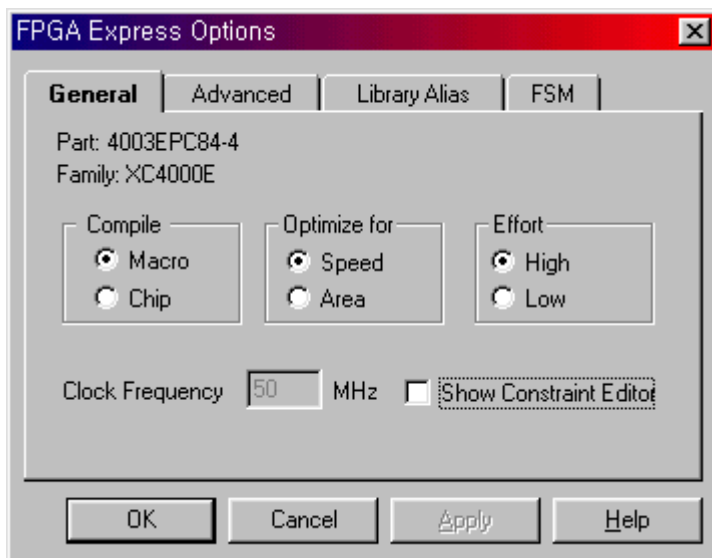
## VHDL 오류 찾기 (Finding VHDL Errors)

VHDL 오류 찾기 기능을 보여주기 위해 하나의 오류를 만들어 본다.

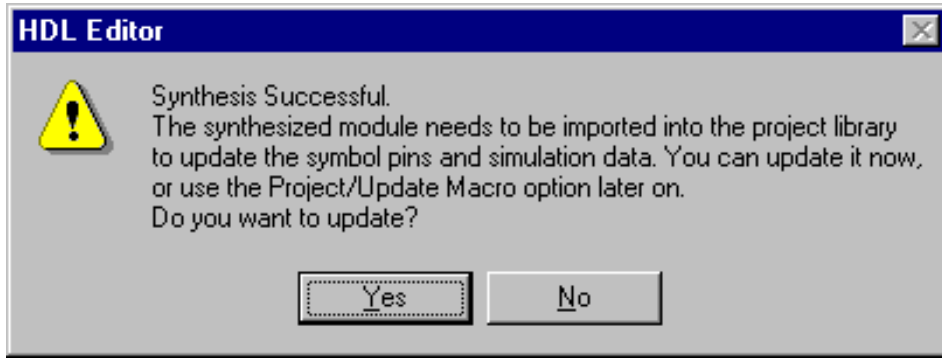
1. port declaration section 안에 있는 ‘;’ 문자를 지운다.
2. **Synthesis** 메뉴에서 **Check Syntax** 명령을 선택한다.
3. **HDL Editor** 창 아래 안에 있는 오류 메시지를 볼 수 있을 것이다. **Source code**의 왼쪽 편에 붉은 화살표가 오류가 어디서 발생했는지 알 수 있는 라인을 지시한다.
4. 여러 개의 오류를 가지고 있다면 아마도 **VHDL code**을 편집하는 동안 실수를 했을 것이다. 앞 페이지의 **VHDL Source code**를 살펴 보고 이상 없이 수정한다.

## 회로합성 하기 (Running Synthesis)

1. **Synthesis** 메뉴에서 **Options**를 선택한다.



2. **Effort Level Low** option을 선택한다.
3. **OK** 버튼을 한번 눌러 준다.
4. **Synthesis** 메뉴에서 **Synthesize**를 선택한다. 이것은 Synopsys 사에서 제공하는 FPGA Express Synthesis compiler를 기동시키고 선택된 **VHDL code**로부터 **XNF netlist**를 생성한다.
5. Synthesis가 다 완료되면, HDL Editor는 매크로를 update 할 것인지를 물어 본다.

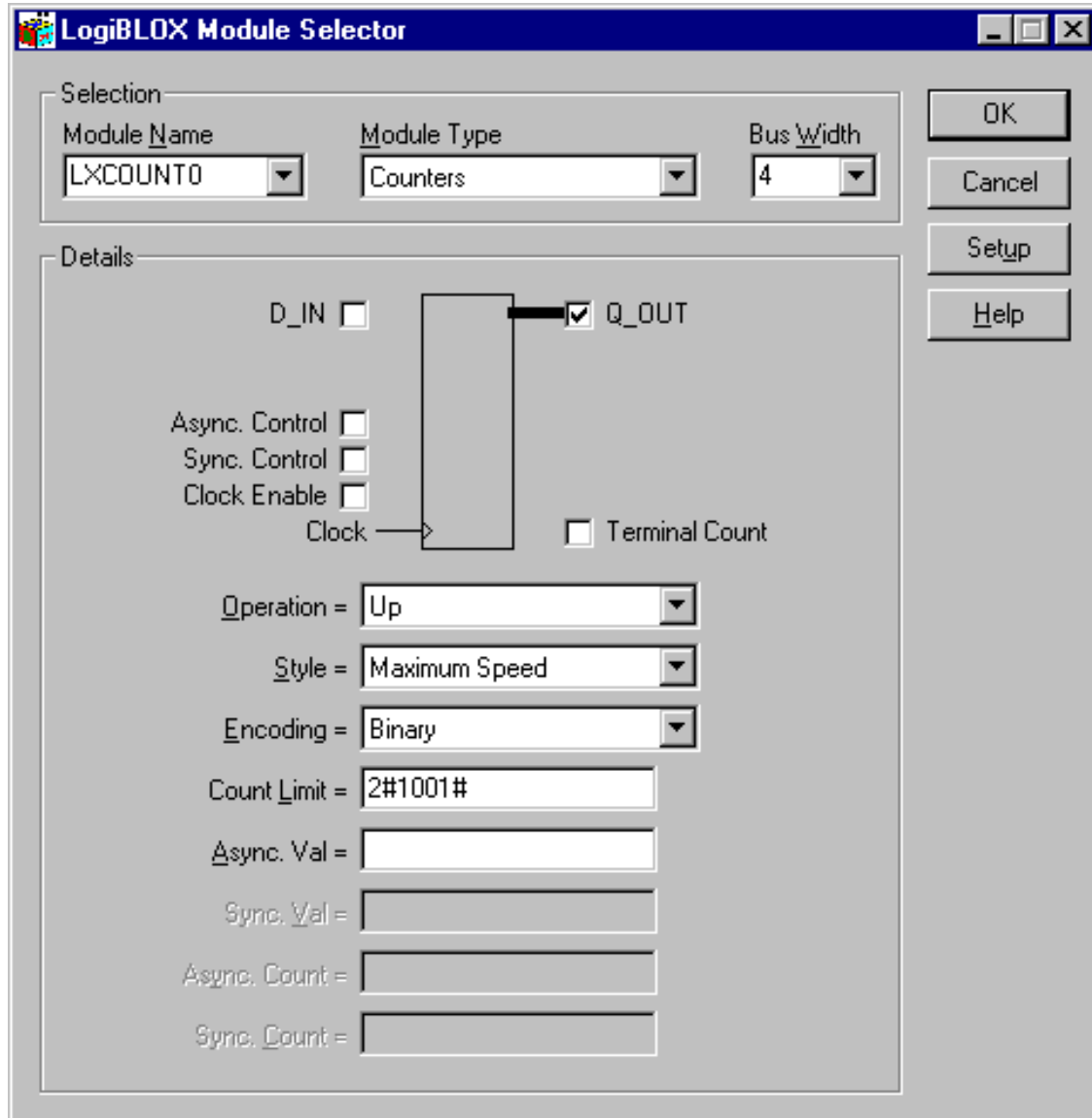


6. **Yes**를 누른다. 이것은 프로젝트 working 라이브러리 안의 매크로 netlist와 심볼을 update 시킬 것이다.
7. HDL Editor가 그 심볼이 성공적으로 Update 되었다는 메시지를 보여주면 **O.K** 버튼을 눌러 준다.
8. HDL Editor를 종료한다.

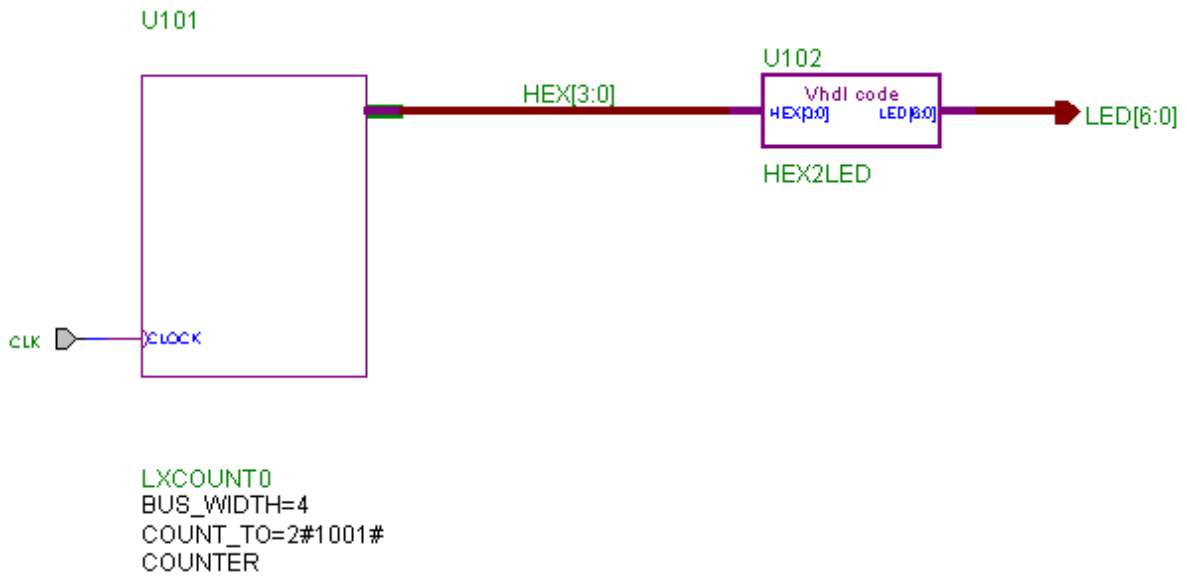
## Schematic Macro 만들기 (Creating a Schematic Macro)

HEX2LED macro을 Update한 후에, 카운터의 나머지 부분도 완성이 될 것이다.

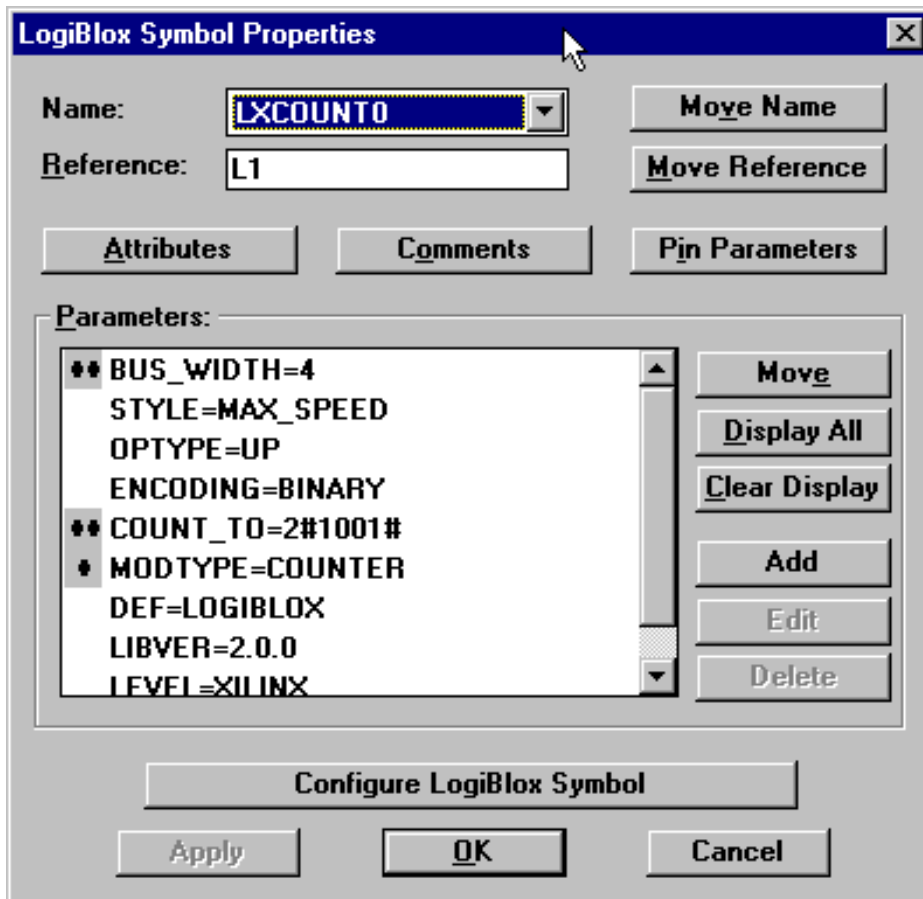
1. **Options** 메뉴에서 **LogiBLOX...**를 선택한다. 이것은 **LogiBLOX Module Selector** dialog를 보여 주는 LBGUI program를 호출 한다.



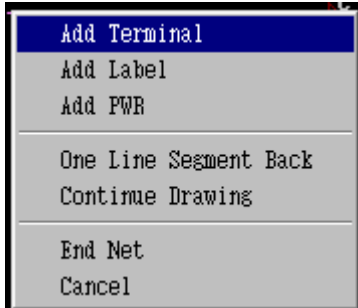
2. Select Counter from the **Module Type** box에서 Counter를 선택 후 **Bus Width** box에 4를 넣는다.
3. **Module Name** box에 LXCOUNT0 라고 넣는다.
4. **Details** group 안에서 **D\_IN** 과 **Clock Enable** boxes를 선택하지 않는다. **Operation** 안에 Up을 선택, **Encoding** 안에는 Binary, 그리고 **Count Limit** field 에는 2#1001#를 넣는다.
5. 프로젝트 라이브러리 안에 LXCOUNT0 logiblox 심볼을 넣기 위하여 **OK**를 누른다.
6. 성공적으로 심볼이 만들어 진 것을 보고하는 메시지를 확인한다.
7. Schematic상에 LXCOUNT0 component을 위치 시킨다.



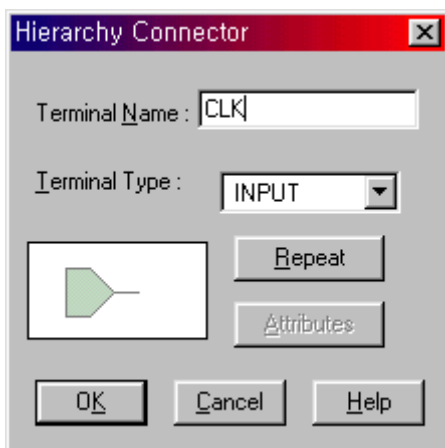
8. **LogiBlox Symbol Properties** 대화 상자를 열기 위해 LXCOUNT0 logiblox 심볼을 두 번 연속 누른다. Configure LogiBlox Symbol 버튼은 대화 상자 안에서 logiblox symbol parameters를 직접 변경할 수 있도록 하기 위해 제공되고 있음을 유의한다. parameter 변경은 component의 이름을 변경하지 않는 한 schematic component 와 라이브러리 module 둘 다 영향을 미친다. 아무 것도 변경하지 않고 대화 상자를 닫는다.




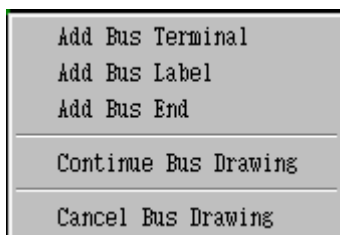
9. 왼쪽편의 매크로의 Clock 핀으로부터 wire 그리기를 시작한다.
10. Wire 그리기를 끝내기 위해 오른쪽 마우스 버튼을 누른다.



11. 위의 대화 상자에서 Add Terminal을 선택한다.
12. 아래 그림에서 보는 것과 같이 CLK 이름을 넣는다.

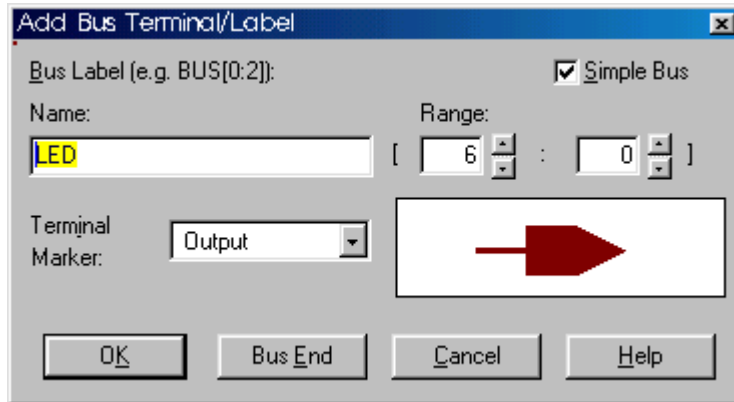


13. 왼쪽 툴 바상에서  버튼을 사용하여, HEX2LED macro의 LED[6:0] 핀으로부터 버스를 오른쪽으로 그린다.
14. 버스 그리기를 멈추기 위해 마우스 오른쪽 버튼을 누른다.




15. 아래 그림과 같이 Bus I/O terminal name을 LED[6:0]로 넣는다.





16. LXCOUNT0의 출력 핀으로부터 버스를 HEX2LED macro의 HEX[3:0] 입력으로 버스를 그린다.

17.  버튼을 누른다.

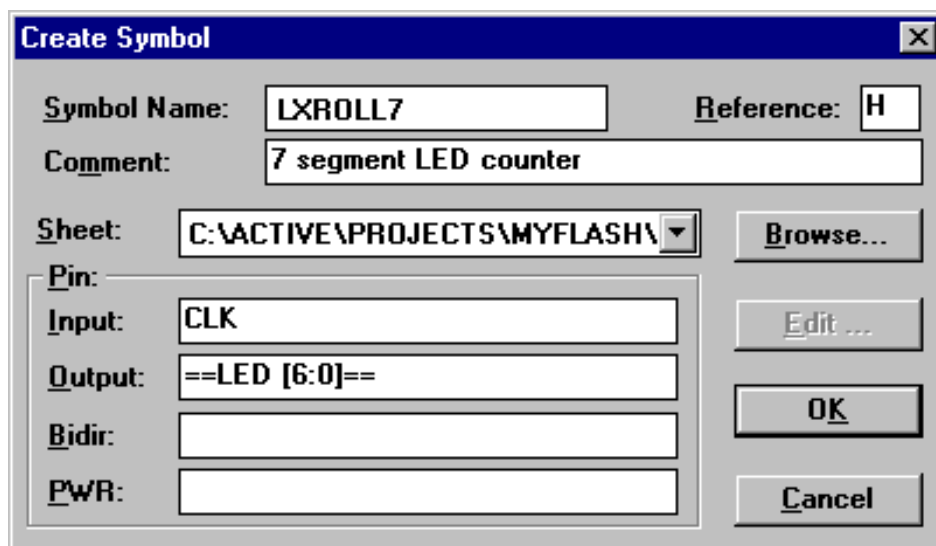
18. Bus name 을 HEX[3:0] 라 넣는다.

## Macro schematic 저장하기 (Saving the Macro Schematic)

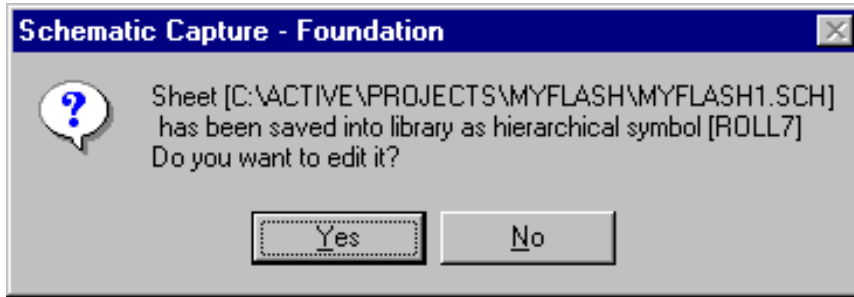
15. Macro schematic이 다 완성되고 나서 **Hierarchy** 메뉴에서 **Create Macro Symbol From Current Sheet** 명령을 선택한다.

16. **Create Macro** 대화창에서 Symbol Name 필드에 LXROLL7 라 이름한다.

17. **Comment** 필드에 "7 segment LED counter"라 주석을 달아 준다.



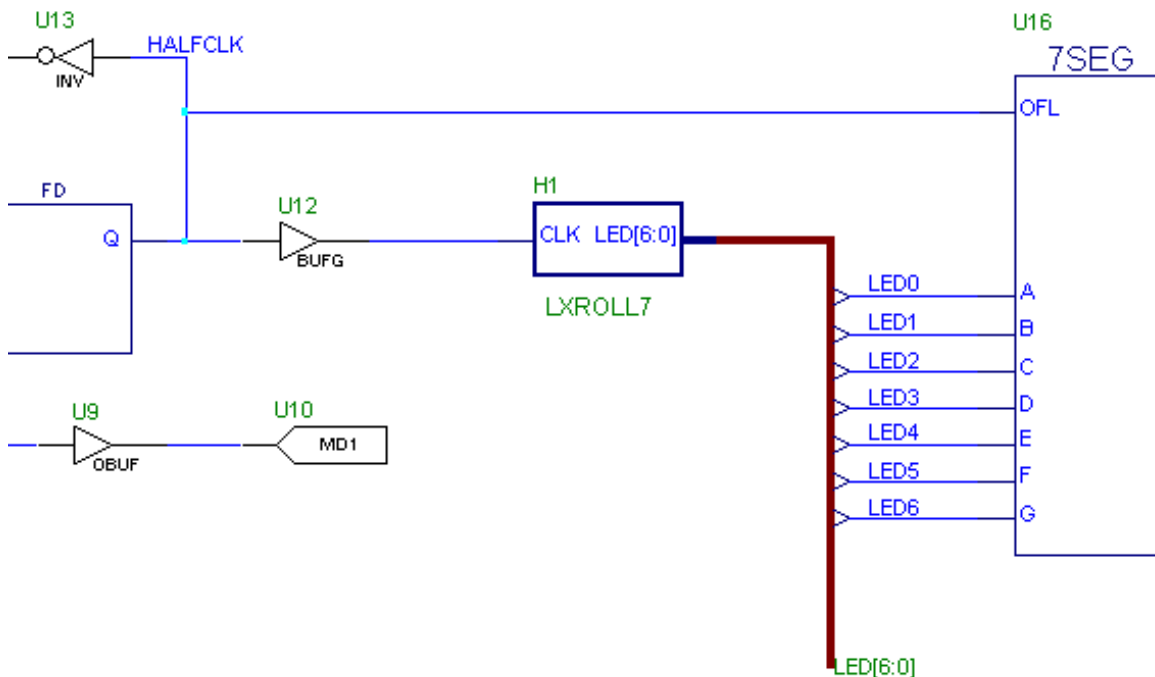
18. Macro을 만들기 위해 **OK** 버튼을 누른다.




4. Macro가 성공적으로 만들어졌다는 정보 창이 보여질 것이다. 이 정보 창이 깜빡일 때 만일 이 매크로를 좀 더 편집하고자 하면 **Yes**를 누르고 **No**를 선택하면 macro schematic이 닫힐 것이다.

## 상위 schematic 끝내기 (Finishing the Top Level Schematic)

1. Top level schematic로 되 돌아 가서 LXROLL7 심볼을 위치시킨다. 이 심볼은 프로젝트 라이브러리에 macro가 저장되는 동안에 소프트웨어에 의해 자동으로 만들어 졌다.
2. 7SEG macro를 위치시키고 아래 그림과 같이 버스 와 Wire를 연결한다.




3.  버튼을 누른다.
4. Bus name (LED[6:0])을 클릭한다.
5. Status line 이 배치를 위해 선택되어진 처음 버스 탭 LED6 을 보여 줄 것이다.




6. Counter의 G 핀을 누른다. 버스 탭이 버스에 그려지고 LED6 로 이름이 붙여질 것이다.
7. 다음 핀들을 F, E, D, C, B, A 순서로 차례차례 누른다.
8. 버스 탭 모드를 빠져 나가기 위해 **Esc** 키를 두 번 누른다.

---


**NOTES:**

- Macro schematic 안으로 들어가기 위해,  버튼을 선택하고 원하는 macro 심볼을 두 번 누른다.
  - 상위 레벨로 되돌아 가기 위해, 하위 레벨 schematic의 background를 두 번 누른다.
- 

## 기능적 시뮬레이션 (Functional Simulation)

1. Functional simulation을 수행하기 위해, schematic 툴 바에서  버튼을 누른다. 디자인의 현재 version이 simulator안으로 들어 올 것이다.
2. Simulation 용 test points을 선택하기 위해, logic simulator 안에 있는  버튼을 누름으로써 schematic 창으로 되돌아 간다. 그런 다음 Schematic Editor 안에 있는  버튼을 누름으로 **Test Points** mode로 들어 간다.
3. Schematic Editor 상에서 다음 네트들을 누른다.  
 CLKIN  
 HALFCLK  
 LED[6:0]  
 Q0, Q1...Q7

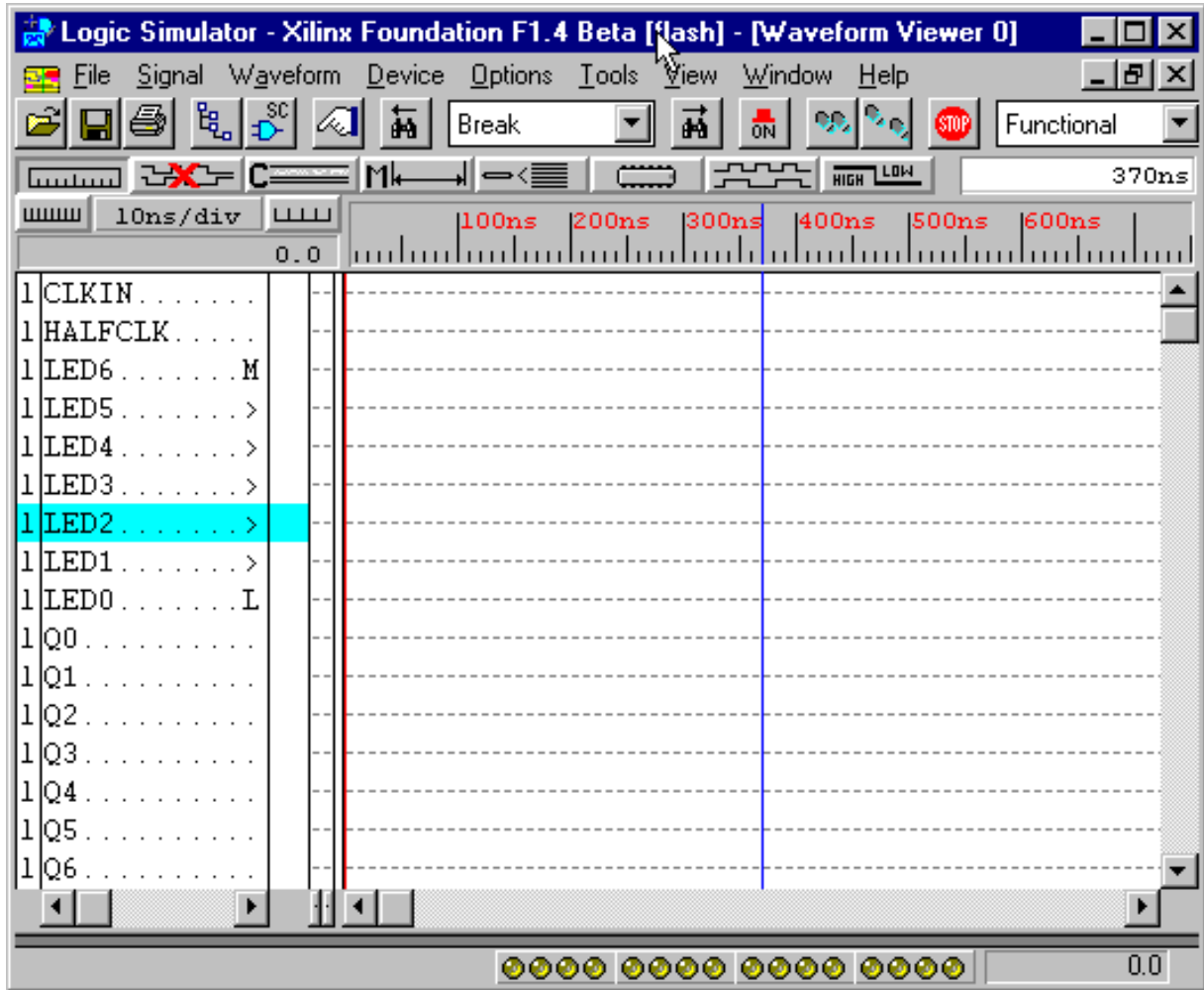
---

**NOTE: Waveform Viewer** 창은 버스를 날개로 또는 묶음 형태 중 하나로 보여 줄 수 있다. Display mode를 바꾸기 위해  버튼을 누른다.

4. Simulator로 되돌아가서 다음 아래에 보이는 것과 같은지 확인 한다.

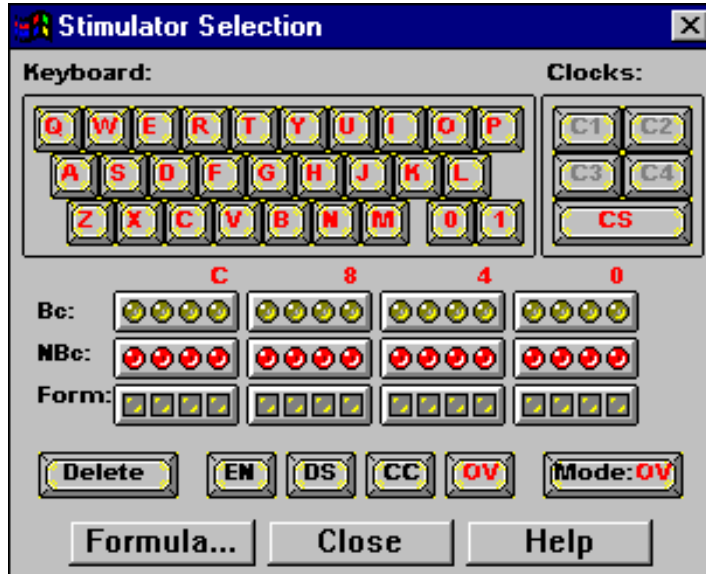
---

**NOTE:** Text Points mode을 벗어 나려면 **Esc** 을 누른다.

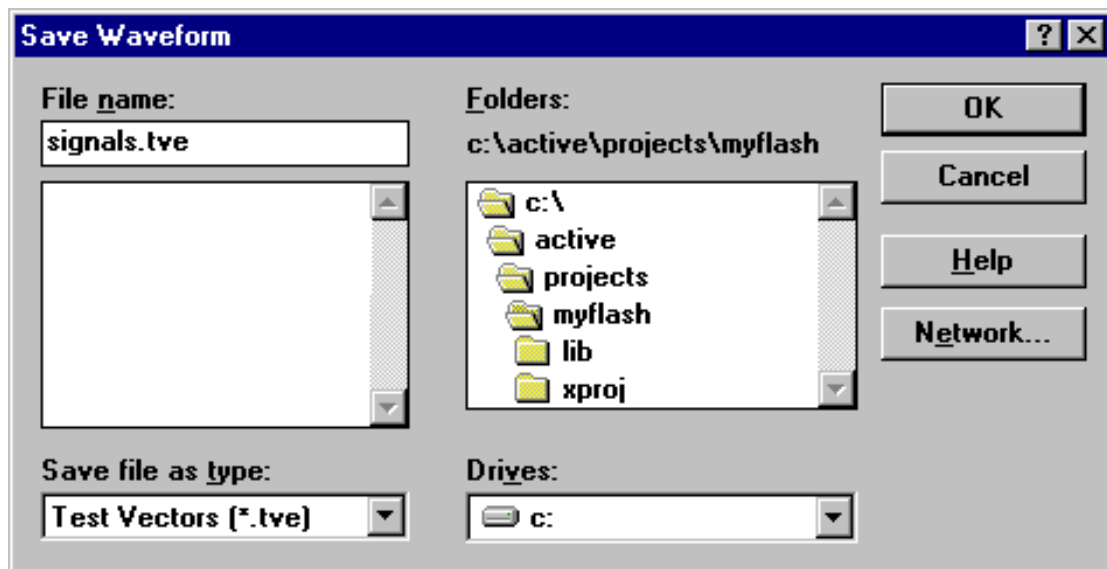


## 시뮬레이션 데이터 만들기 (Creating Test Vectors)

1. Clock Test Vector를 정의하기 위해, **Waveform Viewer**에서 CLKIN signal을 누르면 CLKIN이 highlight 된다.
2. **Signal** 메뉴에서 **Add Stimulators** 을 선택한다.



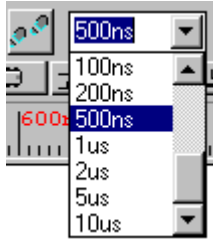
3. Yellow and red *light emitting diode* (LED) icons 은 이진 클럭 발생기의 그림적인 표현들이다. 오른쪽에서 처음 노란 아이콘은 10ns 주기를 가지는 B0 클럭이고, 다음 아이콘은 20ns 주기를 가지는 아이콘이 된다. (Default 설정값을 가지고 있다고 가정하였을 때)
4. 오른쪽에서 두 번째 노란 아이콘을 누르면 **Waveform Viewer** 안에 CLKIN signal에 B1 clock이 보여질 것이다..
5. 선택된 신호를 저장하기 위해서, **File** 메뉴에서 **Save Waveform** 명령을 사용한다.




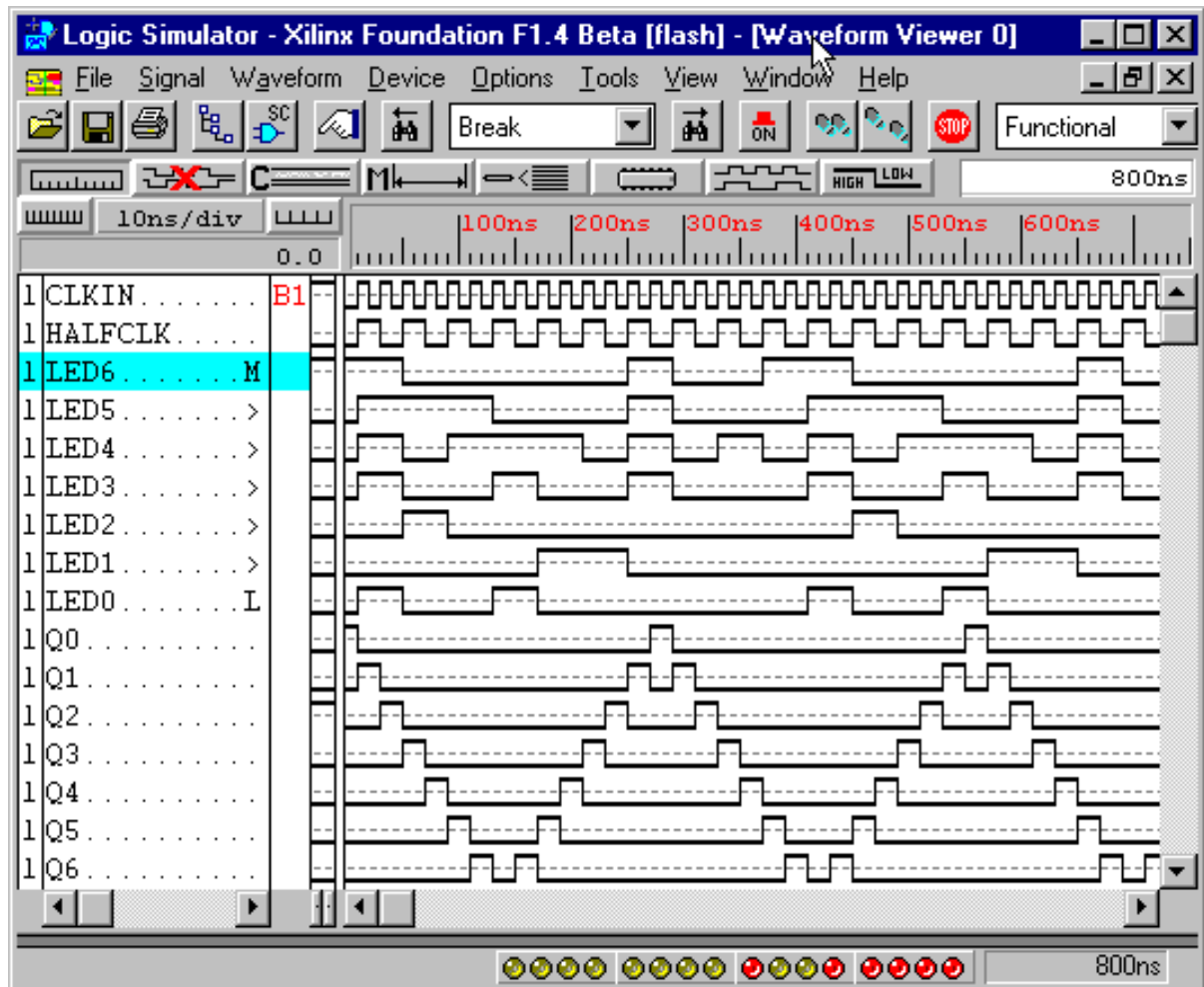
6. 이름을 "signals"이라 치고 **OK** 버튼을 누른다. 이것은 SIGNALS.TVE 파일을 저장한다.

## 시뮬레이션 실행하기 (Running Simulation)

1. Logic Simulator 에서 아래 그림과 같이 simulation period을 정의한 후 발자국 버튼을 누른다.



2. 아래 그림과 같이 waveform이 발생되어 질것이다.
3.  버튼을 여러 번 누른다.



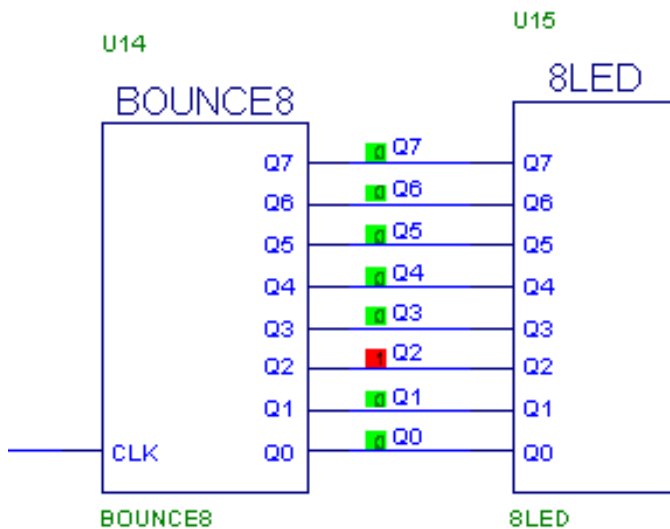
4. **File** 메뉴의 **Save Wafeform** 명령어를 이용하여, 파일명을 FUNCTNL로 하여 simulation 결과를 저장한다. 이 파일은 place and route 후에 나오는 timing simulation 결과와 비교하는

데 사용 되어 질 수 있다.

- 위의 그림 상의 menu는 software의 version에 따라 다를 수 있다.

## Schematic에서 결과 보기 (Viewing Results on the Schematic)

- Schematic 창으로 전환한다.
- 선택된 신호들이 색깔 점검단자로 표시되는 것을 주목하라. 로직의 상태에 따라 다음과 같이 색깔이 대응 된다.
  - Green = "0" logic 0
  - Red = "1" logic 1
  - Blue = "X" (Unknown)
  - Yellow = High Impedance/Not driven
- schematic에서부터 simulation을 계속 진행하려면 **SC Probes** toolbox 안에 있는 **Step** 버튼을 누른다.
- Q0...Q7 signal lines 상에서 red icon이 위 아래로 이동하는 것이 확인하라.

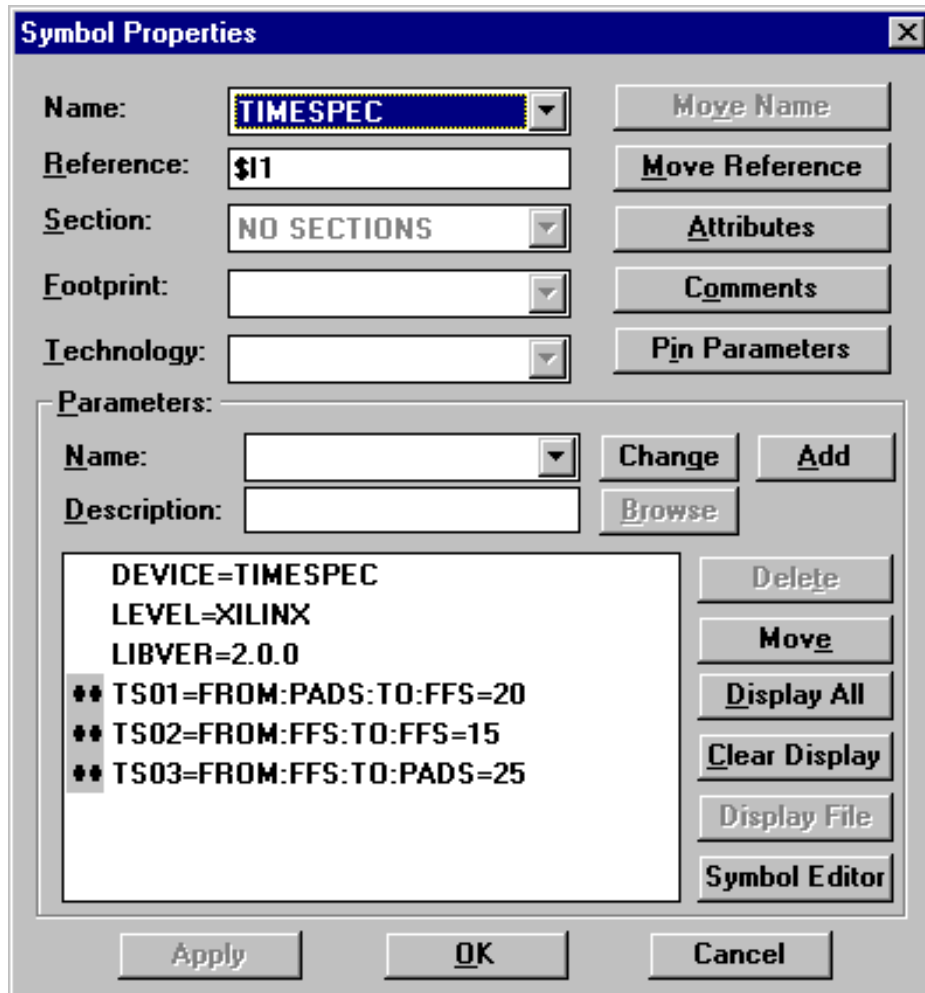


## TIMESPEC Symbol 첨가하기 (Adding TIMESPEC Symbol)

- 라이브러리에서 TIMESPEC 심볼을 선택하여 그 것을 schematic상에 위치 시킨다.
- Esc** 키를 누르고 TIMESPEC 심볼을 두 번 누른다. **Symbol Properties** 대화상자가 나타날 것이다.
- New parameters 첨가하기:
  - Name filed에 TS01을 넣는다.

- description field에 FROM:PADS:TO:FFS=20 이라 넣는다.
  - **Add**를 누른다.
- 다음 파라미터를 추가하기 위해 같은 과정을 반복하라.:
- TS02=FROM:FFS:TO:FFS=15
  - TS03=FROM:FFS:TO:PADS=25
4. name과 description mark(diamonds)가 보이게 하기 위하여 추가된 파라미터들을 누른다.
  5. **OK** 버튼을 누른다.

**NOTE:** The diamonds are responsible for displaying the parameters on the schematic



6. 보이는 심볼이 TIMESPEC이 추가된 결과이다.



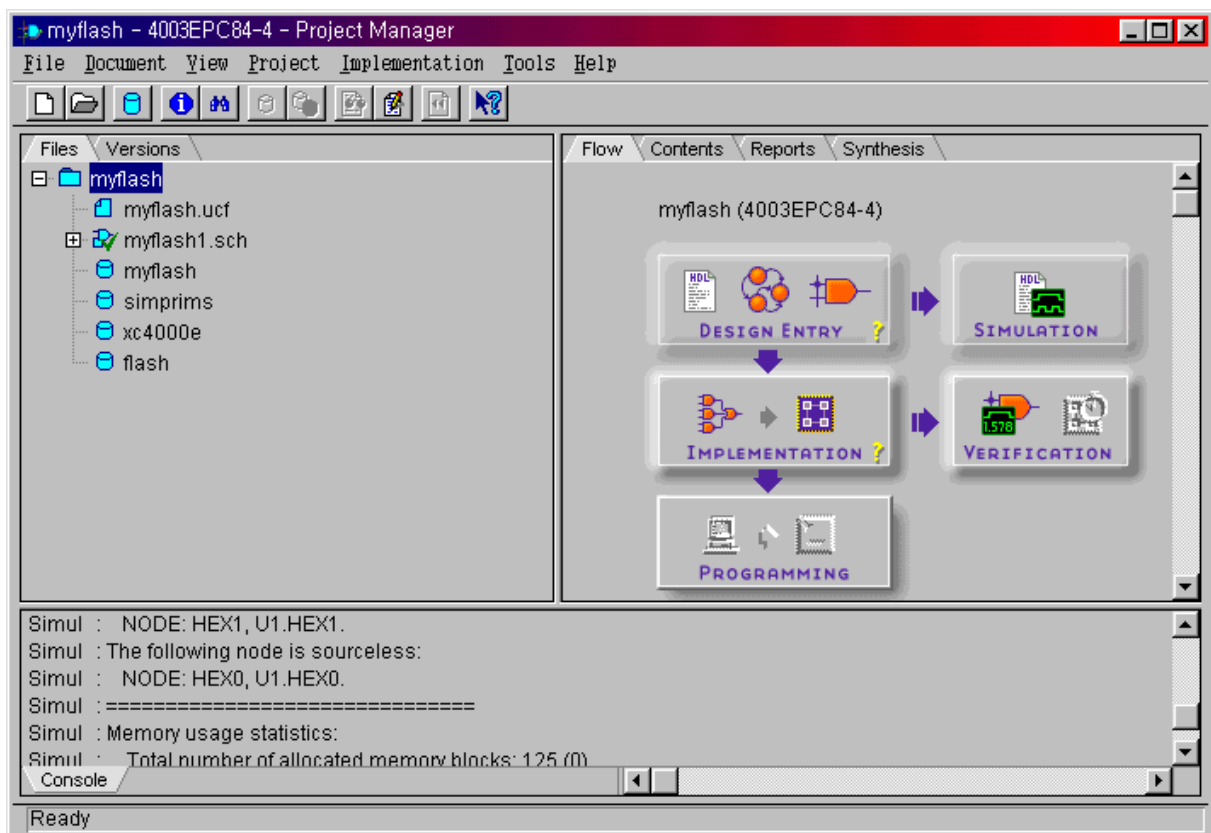
TIMESPEC
TS01=FROM:PADS:TO:FFS=20
TS02=FROM:FFS:TO:FFS=15
TS03=FROM:FFS:TO:PADS=25

7. Schematic의 변경을 저장하기 위해 **Save** 버튼을 누른다.

## XACT 실행하기 (Running XACT)

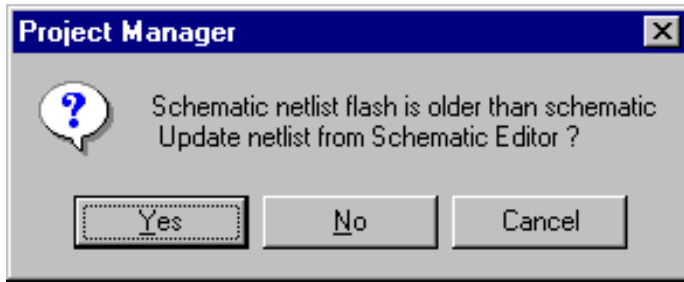
XACT는 Xilinx Automatic CAE Tools의 첫글자를 따서 만들어진 글자로서, 지금까지 작업한 회로를 가지고 실제 Xilinx Device에 들어갈 Data로 변환하는 Place & Route를 수행하는 프로그램을 일컫기도 한다.

1. 먼저 기존에 열어 놓은 **Schematic** 과 **Simulator** 창을 닫는다. (프로그램을 끝낸다는 의미로서, 굳이 실행을 끝마치지 않아도 XACT를 수행하는데 별이상은 없으나, XACT수행시 좀더 많은 메모리를 확보하기 위해서 이런방법이 권장된다.)
2. **Project Manager** 창에서 **Implementation** 버튼을 누른다.

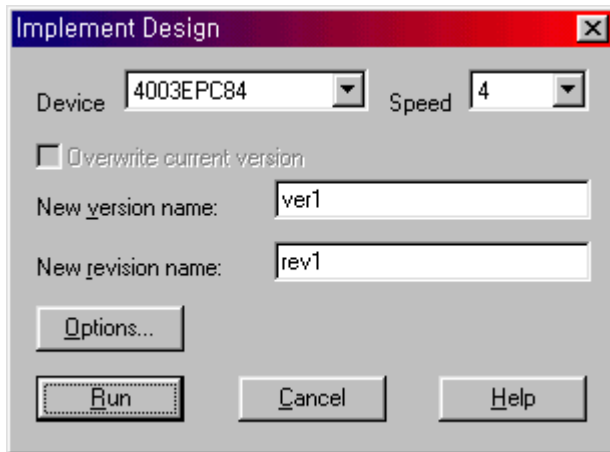


3. Project Manager가 schematic netlist가 up-to-date가 되지 않았다는 정보를 줄 것이다. Netlist를

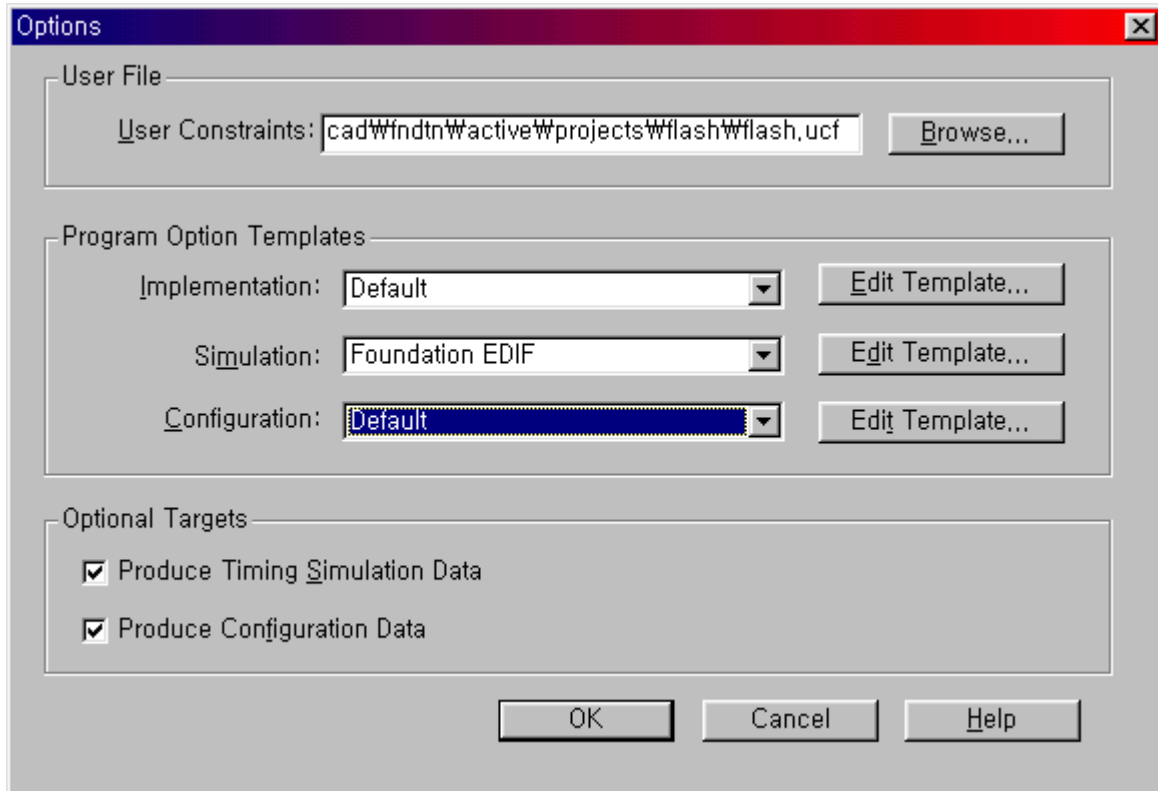
Update하기 위해 **Yes**를 누른다.



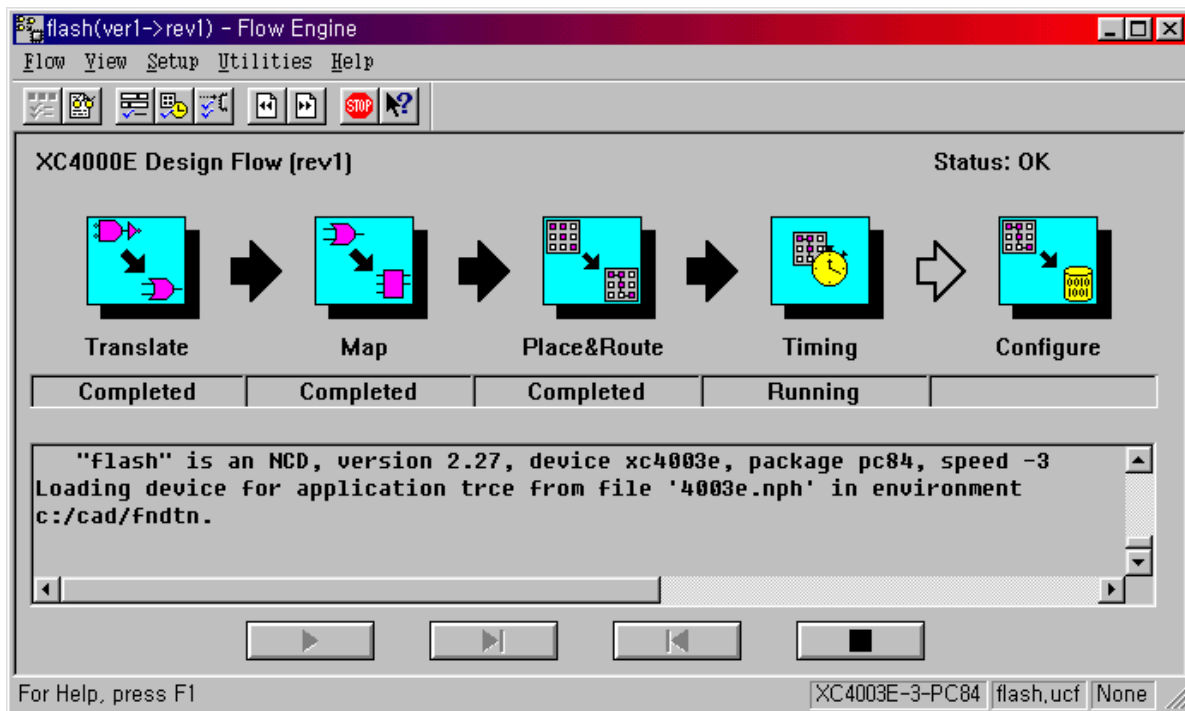
4. 아래 그림과 같은 대화 상자가 나타나면 사용자가 편의에 맞게 Version 및 Revision name을 정의한다.



5. **Options** 버튼을 누른다. 아래의 대화 상자에서 **Produce Timing Simulation Data** check box을 선택하고, **OK**을 누른다.



6. Place and Route 과정을 시작하기 위해 Implement Design 대화 상자에서 **Run** 버튼을 누른다.

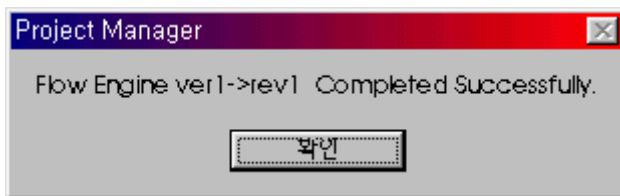


7. PAR(Place and Route)과정이 다 완료되면, Design Manager가 메시지 box를 보여 줄 것이다. Design Manager를 닫기 위해 **OK**를 누른다. 각 단계에 대한 간략한 설명은 아래와 같다.

- TRANSLATE : XNF, EDIF Format의 Netlist 파일을 읽어 들여 AND, OR gate, Decoder,

RAM등과 같은 Logic Element로 표현되는 Logic Design인 NGD(Native Generic Design)파일로 변환하는 과정이다. 이 NGD파일은 Xilinx Internal Database file format이다.

- **MAP** : 디자인된 Logic Element들을 CLB(Configurable Logic Block)나 IOB(Input Output Block)같은 Physical Element들로 할당하는 과정이다.
- **PLACE & ROUTE** : Place는 CLB, IOB등 Logic Block들을 Xilinx Device의 특정 위치에 할당하는 과정이고, Route는 Xilinx Device에 위치한 Logic Block들을 Interconnection Element들로 서로 연결하는 과정이다.
- **TIMING** : 이 과정은 Place & Route된 Design의 Physical Delay을 산출하여 Timing Simulation 수행 시 필요한 Data을 만들며 개략적인 System Performance Report을 만든다.
- **CONFIGURE** : Physical Implementation를 Binary Stream으로 변환하여 design.bit 파일을 만든다. 여기서 만들어진 bit 파일을 Xilinx Device에 Download할 때 사용한다.



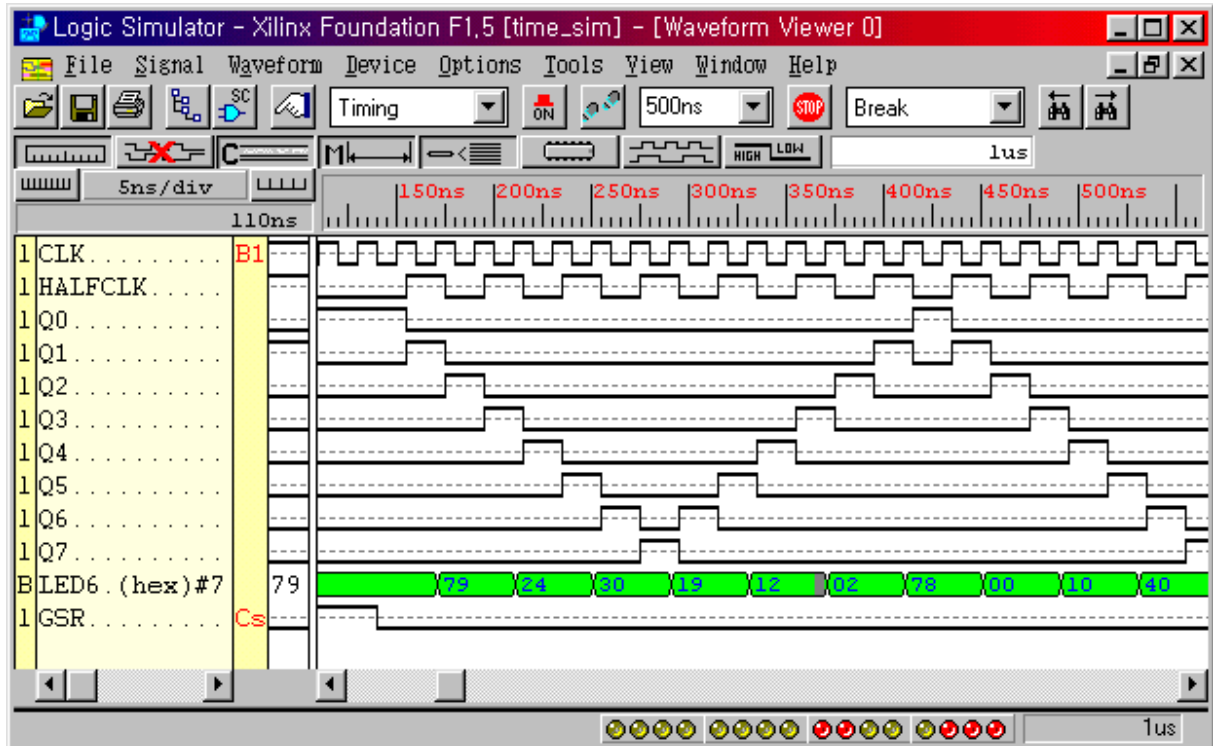
## Timing Simulation 하기 (Timing Simulation)

1. 프로젝트 Flowchart에서 **Verification** 버튼을 누름으로 Timing Simulation을 시작한다.

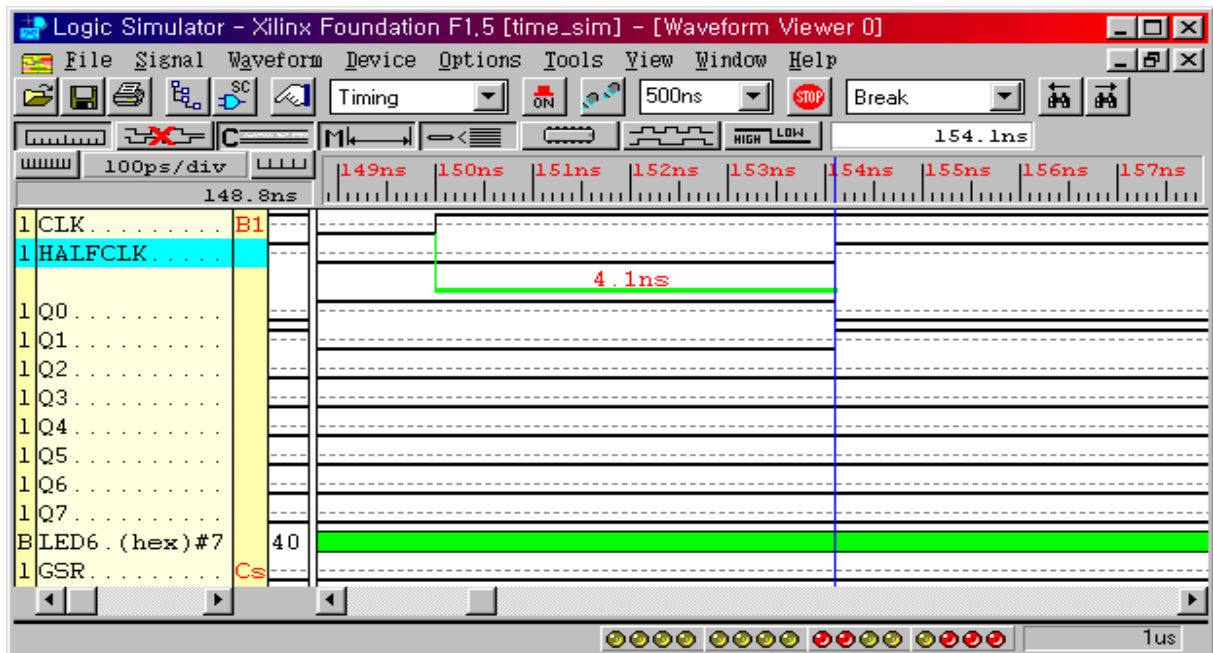


2. Timing simulation Result는 아래 그림과 같을 것이다. 다만 주의해야 할 사항은 FPGA를 디자인 할 때에 Timing Simulation시 Global Reset을 사용하여 Power-On 효과를 내어야 한다는 것이다. 이는 모든 Flip-Flop의 초기값을 "0" 로 만드는 것이다. 여기서는 GSR(Global Set Reset) Signal을 초기에 high로 하여 모든 Flip-Flop을 Clear하고 일정 시간이 경과하면 low로 하여 동작시의 값을 Flip-Flop이 가지도록 한다. (Foundation Series 1.4까지는 위의 작업을 사용자가 일일이 손수해 주었어야 했지만 Foundation Series 1.5에서 부터는 이 작업을

Simulator가 알아서 스스로 해주므로 크게 신경을 쓰지 않아도 된다. 그러므로 사용자는 Timing Simulation시 GSR이란 신호가 Waveform Viewer에 왜 나타나는지만 이해하면 된다.)



3 위의 그림에서 보는 것과 같이 Simulation Mode가 Timing으로 되어 있는 것을 알 수 있을 것이다.



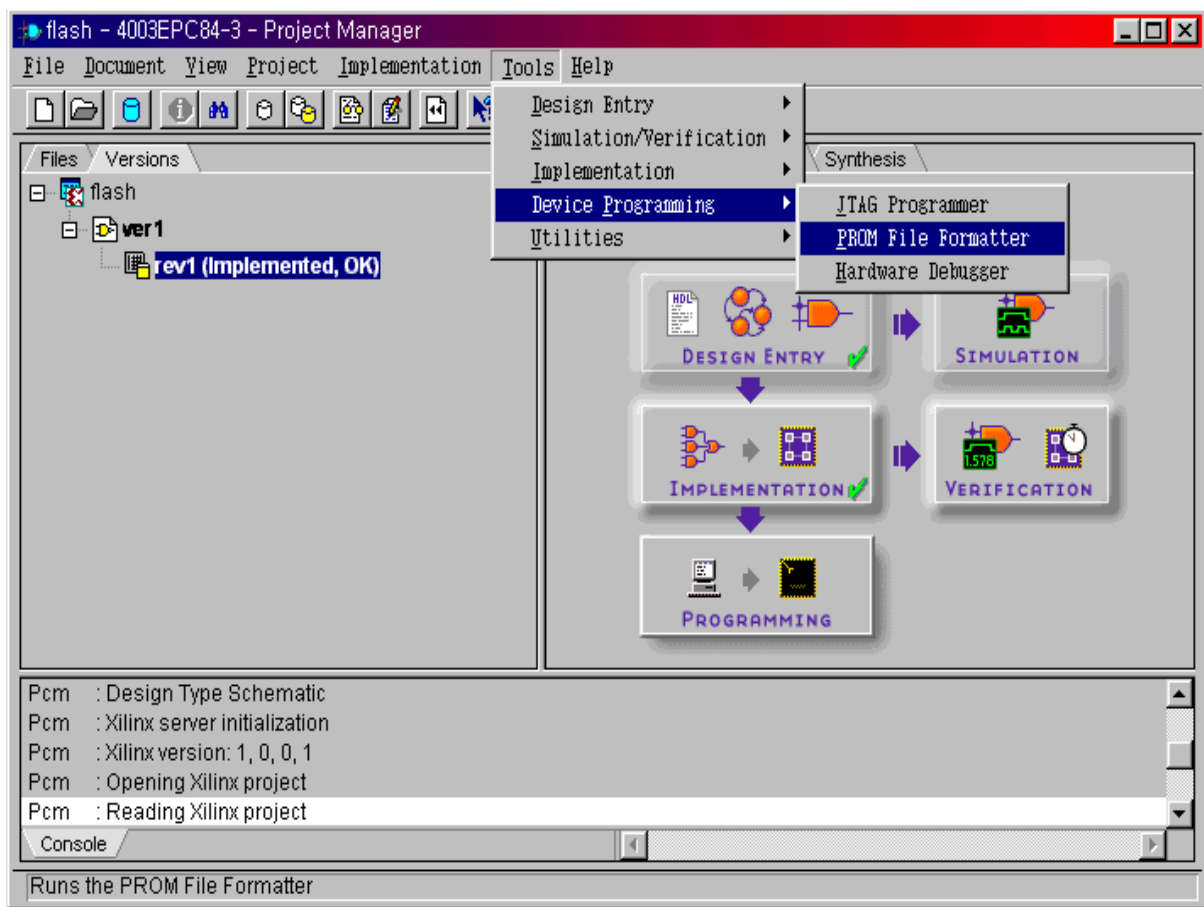
4 위의 그림을 보면 CLK이 분주하여 HALFCLK으로 가는 데 걸리는 실제 delay가 얼마나

생기는 지를 알 수 있을 것이다.

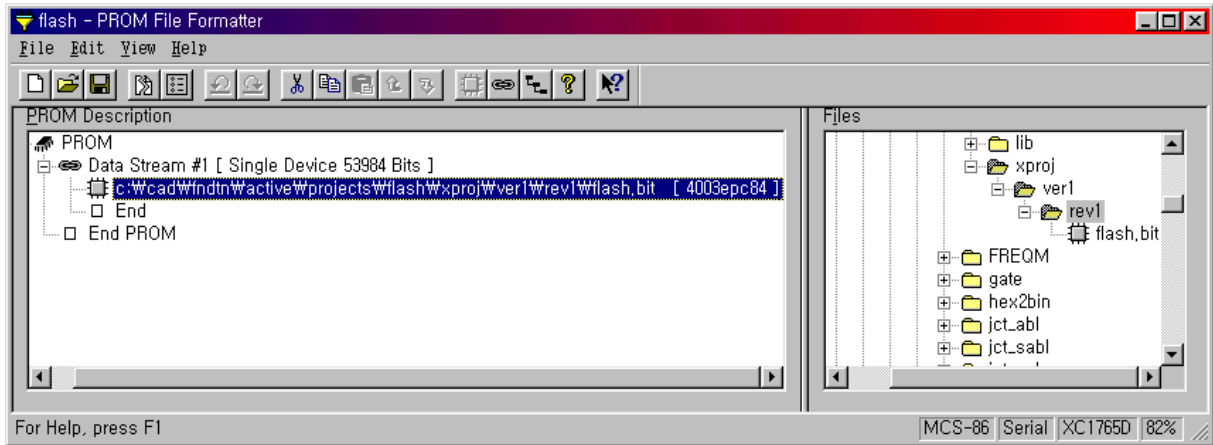
## ROM File 만들기 (Creating a MCS date for PROM)

Xilinx FPGA는 SRAM Based Architecture이기 때문에 별도의 Configuration Data을 저장할 ROM (Read Only Memory)이 필요하며 이를 만족시키기 위해서는 Configuration Data을 ROM file 로 저장하여 범용ROM또는 Xilinx Serial PROM에 저장하여야 한다.

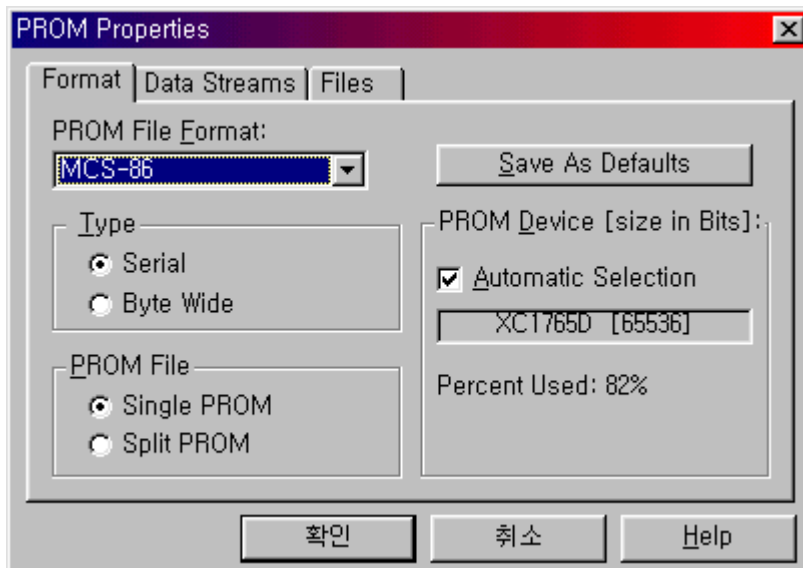
사용자는 ROM에 저장할 Data를 만들어야 한다. 아래 그림에 표현 한 대로 정리하여 만들기 바란다.



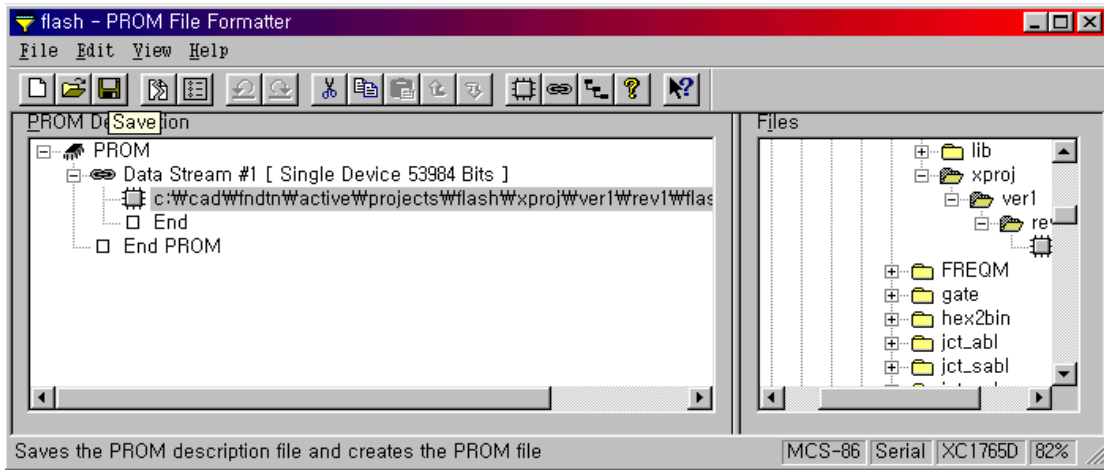
1. 위의 그림에서 보는 바와 같이 Tools -> Device Programming -> PROM File Formatter을 선택한다.
2. 아래 그림처럼 PROM File Formatter라고 하는 별도의 프로그램이 실행되면서 현재 열려있는 Project에서 ROM File을 만들기 위해 선택한 Version의 revision을 자동으로 인식하여 화면 왼쪽 PROM Description란에 자동적으로 등록이 된다.
3. 아래 그림에서 File -> PROM Properties을 선택하여 만들고자 하는 ROM File을 선택한다.



4. 여기서는 Xilinx 의 PROM을 선택한다.



5. 위의 그림에서 사용하는 ROM Writer을 선정하여 거기에 맞는 Data Format을 선택한다.  
여기서는 Intel Hex Format인 MCS-86을 선택한다.
6. 이 디자인은 Xilinx PROM인 XC1765D을 자동적으로 선택한다. 만일 Bitstream이 선택한 PROM이 담을 수 있는 용량보다 클 때에는 Split PROM을 선택하여 구분된 Data을 만든다.
7. 원하는 선택 사항을 만들면 확인 버튼을 누른다.



8. 위의 그림에서 Floppy Icon을 눌러 MCS 파일을 저장한다. (실제로 Floppy Icon을 누르면 저장할 파일명을 물어보는 윈도우가 뜨는데 그곳에서는 확장자가 PDR로 지정이 되어있다. 이는 PROM Property에서 선택한 옵션사항들이 저장되는 파일로서 이 파일을 저장하면 자동적으로 MCS File도 생성이 된다.)
9. 오른쪽 Directory Structure에서 저장된 MCS 파일의 Directory를 찾아서 필요한 mcs 파일을 ROM-Writer Program이 있는 directory로 복사하여 필요한 ROM을 굽는다.



## 후기

이상으로 회로 디자인에서 최종 동작까지 필수적인 과정들을 수행하였습니다.

좀 더 자세한 교육자료가 필요하시면 <http://support.xilinx.com/support/techsup/tutorials/index.htm>로 가셔서 Foundation F1.5 Watch Tutorial 9/98 (1580KB)과 Foundation Watch Tutorial Design Files (ZIP format), 9/98 (1646KB)을 받아서 업무에 적용하시길 바랍니다.

회로 설계시 의문사항이나 좀 더 자세한 내용을 알고자 하는 분은 현명전자 응용기술부의 F.A.E (Field Application Engineer)에게 도움을 받으시길 바랍니다.

하태욱 [twha@hmelec.co.kr](mailto:twha@hmelec.co.kr)

차영근 [ykcha@hmelec.co.kr](mailto:ykcha@hmelec.co.kr)

양창우 [cwyang@hmelec.co.kr](mailto:cwyang@hmelec.co.kr)

이경득 [kdlee@hmelec.co.kr](mailto:kdlee@hmelec.co.kr)

**1998년 3월 21일 Revision 2...**