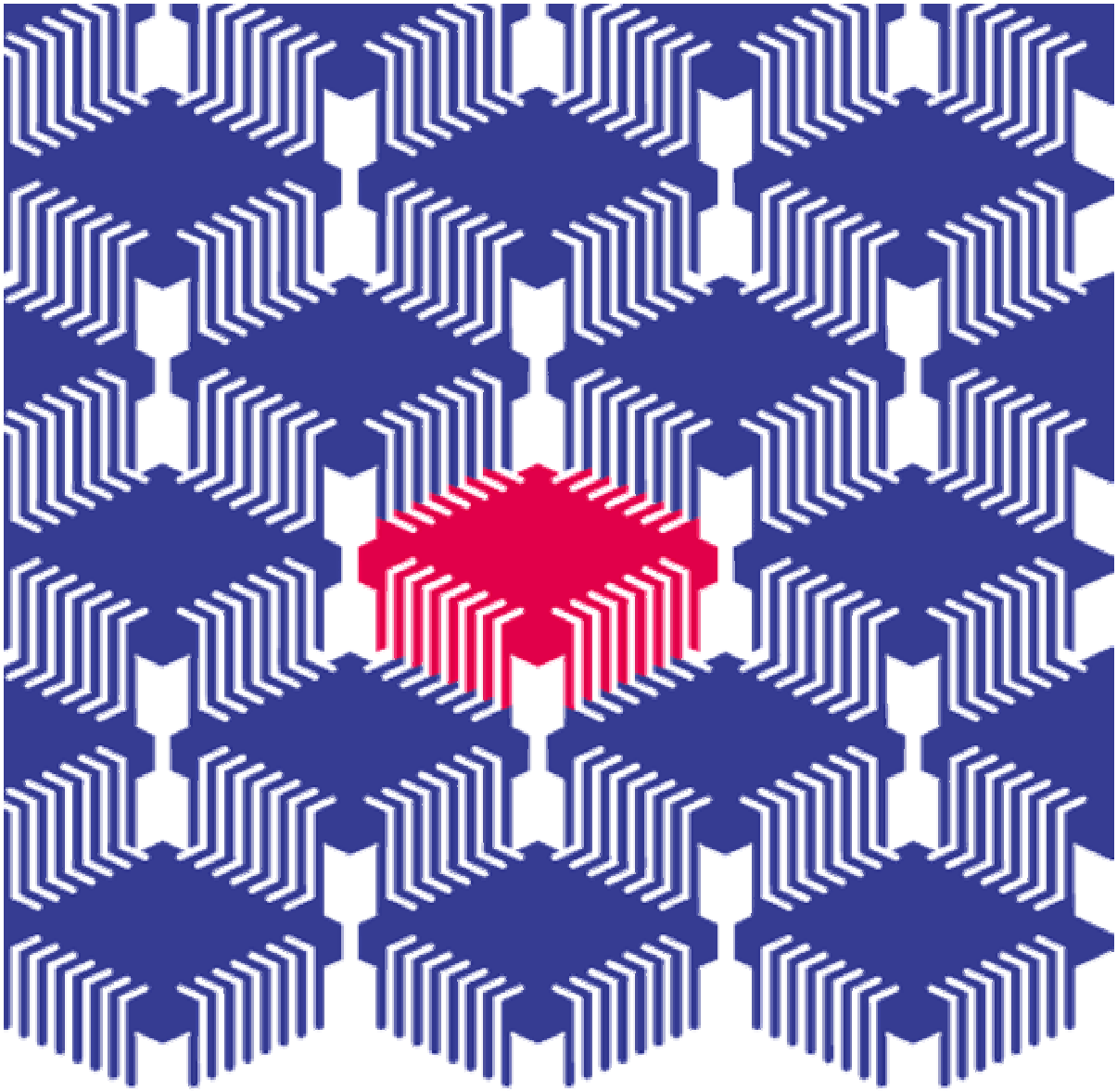


ALTERA®

MAX+PLUS® II GETTING STARTED

무작정 따라하기

PLDWorld.com

2001 년 10 월 4 일 / Version 2.0…

0

MAX+plus II 를 이용한 Digital 논리회로 설계

본 문서는 기존의 회로도 즉, Schematic Capture 를 이용한 디지털 논리회로를 설계하는 방법에 대해서 다룹니다... MAX+plus II 를 사용하면, 디지털 논리 시스템을 다루는데 흔히 사용되는 방법으로서 각종의 IC 를 만능보드에 설계하지 않고 직접 원하는 회로도를 입력하고 나서, 로직의 상태를 점검하는 컴파일과 시뮬레이션의 과정을 통해 원하는 기능이 나오는지를 간단히 확인할 수 있습니다... 이러한 일련의 과정을 마친 후에 CPLD 나 FPGA 를 이용하여 하드웨어적으로 디지털 로직 (Logic)을 구현합니다...

본 문서에서는 ALTERA 사의 PLD 인 FLEX10K Series 중 EPF10K10QC208-4 를 이용한 설계 과정을 다룹니다...

본 문서에서 설명되어지는 내용의 범위는 MAX+plus II 에서 Project 생성, Schematic 작성하기, 사용자 Design 의 Compilation, Simulation 의 수행까지 입니다...

이 이후부터는 편의상 경어의 사용을 생략합니다...

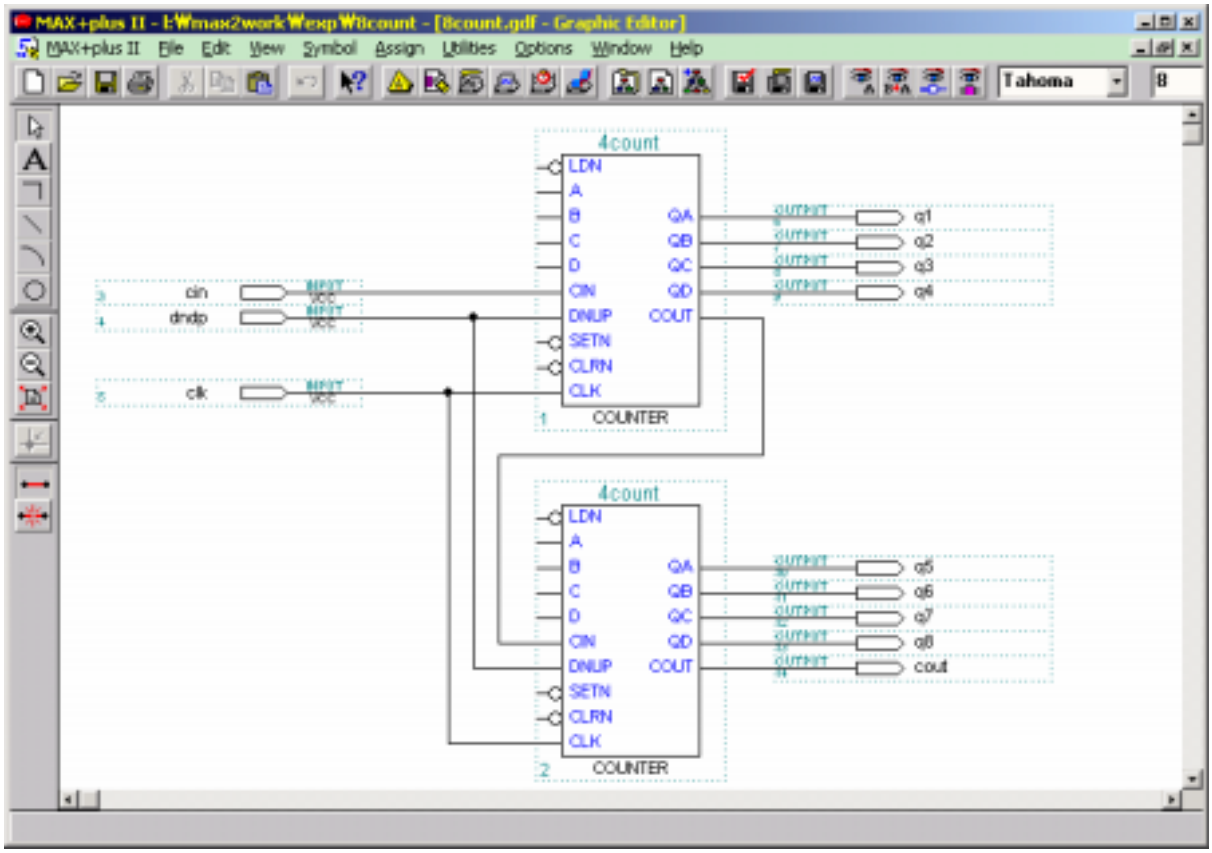
1

MAX+plus II 시작

본 장에서는 기본적인 MAX+plus II 의 시작에 대해서 다룬다. 이를 위해 임의의 작업 Directory 에 Folder 를 만든 다음 이를 MAX+plus II 의 작업 Directory 로 사용하기로 한다. 만약에 하드디스크 드라이브 『C:』 또는 『D:』가 준비되어 있으면 다음과 같이 즉, 『C:\max2work\exp』 또는 『D:\max2work\exp』라는 Directory 를 만든다. 본 과정에서는 『L:\max2work\exp』 라는 Directory 를 만들어 이를 작업 Directory 로 사용하기로 한다.

본 교재에서 작업할 회로는 **"4-Bit Binary Up/Down Counter with Synchronous Load (LDN), Asynchronous Clear, and Asynchronous Load (SETN)"**를 두개 사용한 간단한 8-Bit Binary Counter 이며 완성된 회로도는 다음장의 그림과 같다.

이후 설명되는 각종 사항은 본 회로를 기초로 이루어 진다.



이제 PC 의 바탕 화면에서 MAX+plusII 10.1 의 단축 아이콘이나 그림 1 과 같이 Windows 95/98/NT/2000 의 **시작** 메뉴 -> **프로그램(P)** -> **Altera** -> **MAX+plus II 10.1** 을 선택하여 MAX+plus II 의 초기 화면으로 들어간다.

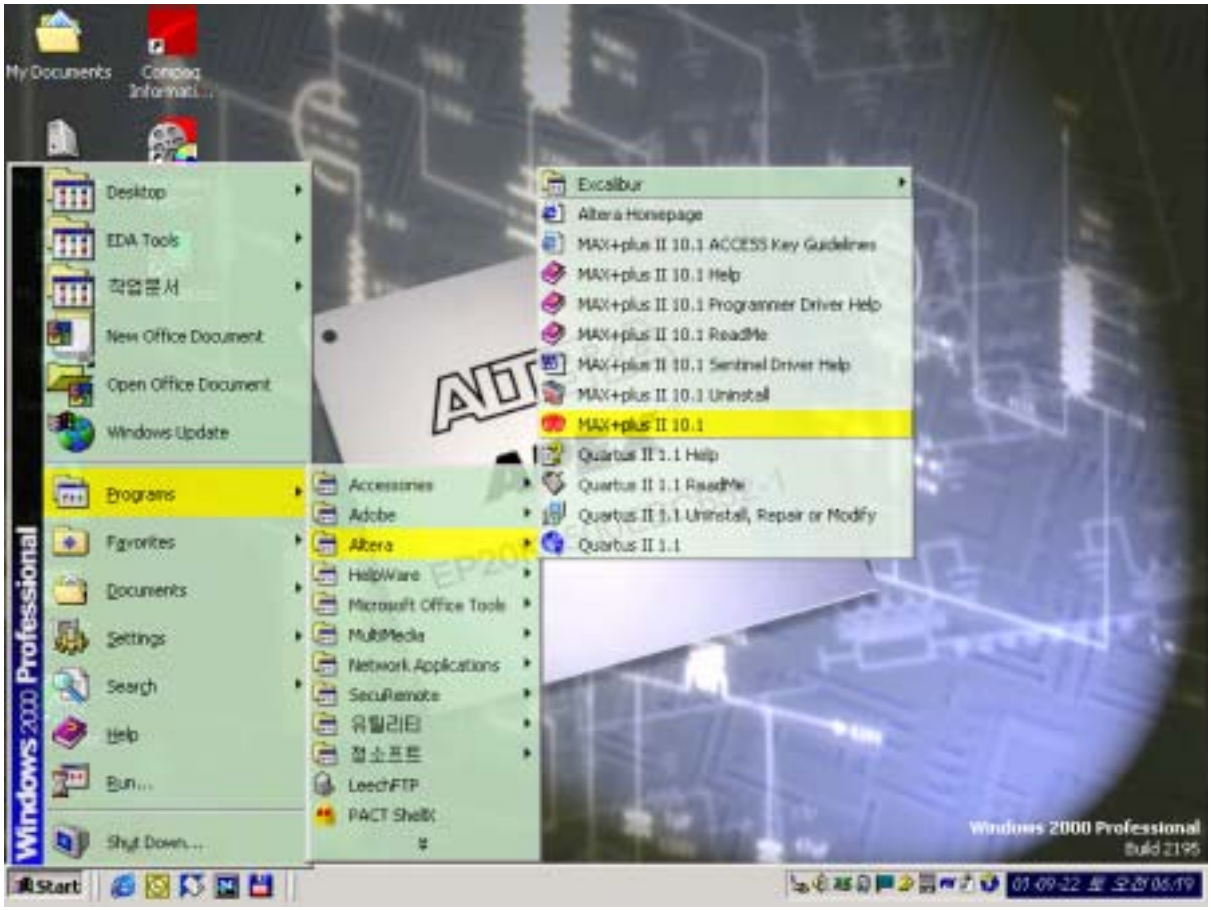


Figure 1 MAX+plus II 의 시작

앞에서 설명한 방법으로 MAX+plus II 를 실행하면 이에 대한 초기화면 구성은 그림 2 와 같이 표시된다. 이때 화면의 상단에는 현재 설정된 Project 의 이름이 표시되는데 그림 2 의 경우는 처음 시작의 상태이므로 "Untitled1"로 표시되어 있다.

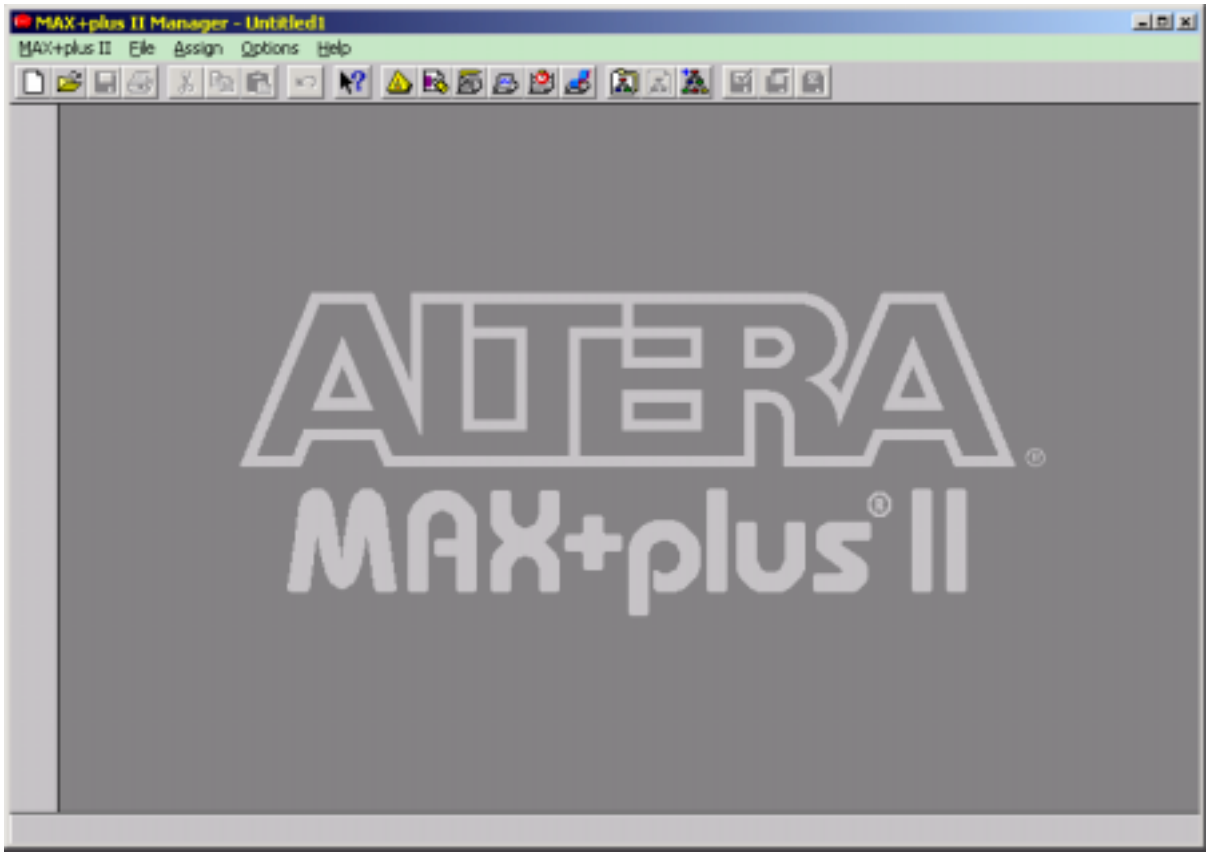


Figure 2 MAX+plus 초기화면

Toolbar 에는 모든 MAX+plus II 내에서 자주사용되는 각종 명령어들이 포함되어 있는데 그림 2 에서 화면 상단에 있는 것으로 다음과 같은 것을 Toolbar 라 한다.

Figure 3 MAX+plus II 에서 제공하는 Toolbar

2

Graphic Design File 의 생성

제 2 장에서는 회로도 (Schematic Capture)를 이용한 디지털 시스템의 설계 방법에 대해 다룬다. 이를 위해 Project 이름을 정한 후 Graphic Editor 를 이용하여 디지털 시스템을 설계한다. 이에 대한 설계 흐름은 다음과 같다.

1. **프로젝트 이름 (Project name)을 정한다.**
2. **새로운 파일 (New file)을 만든다.**
3. **새파일을 저장하여 Schematic 을 입력할 수 있는 환경을 설정한다.**
4. **Logic 함수에 대한 심벌 (Symbol)을 입력한다.**
5. **Symbol 을 원하는 위치로 이동한다.**
6. **입력 핀 (input pin)과 출력 핀 (output pin)을 입력한다.**
7. **Node 와 Bus 들을 연결한다.**
8. **해당 Pin 에 대한 이름을 입력한다.**
9. **회로를 완성한다.**
10. **Target Device 를 선정한다.**
11. **File 을 저장한 후 Compiler 를 통하여 기본적인 Error 를 Check 한다. (Design Rule Check 를 수행한다.)**
12. **File 을 Close 한다.**

이상과 같은 설계 흐름을 따라 한 단계씩 순서에 따라 진행한다. 이를 위해 먼저 그림 4 처럼 메뉴를 사용하여 새로운 Project 를 선택한다.

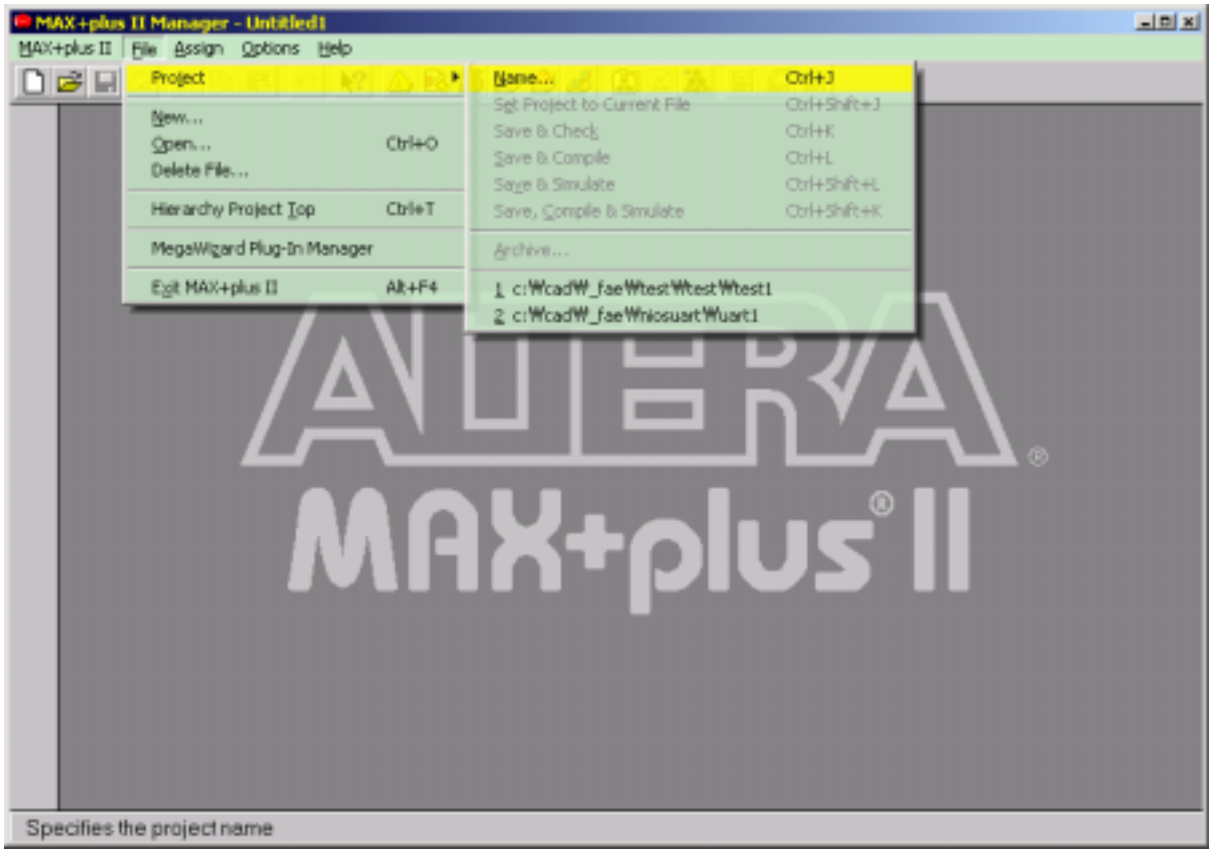


Figure 4 새로운 Project 메뉴 선택

그림 4 에서 Name...을 선택하면 새로운 Project 를 생성하여 작업함을 의미하며 그에따라 그림 5 와 같이 새롭게 작업할 Project 이름을 묻는 대화창이 나타난다.

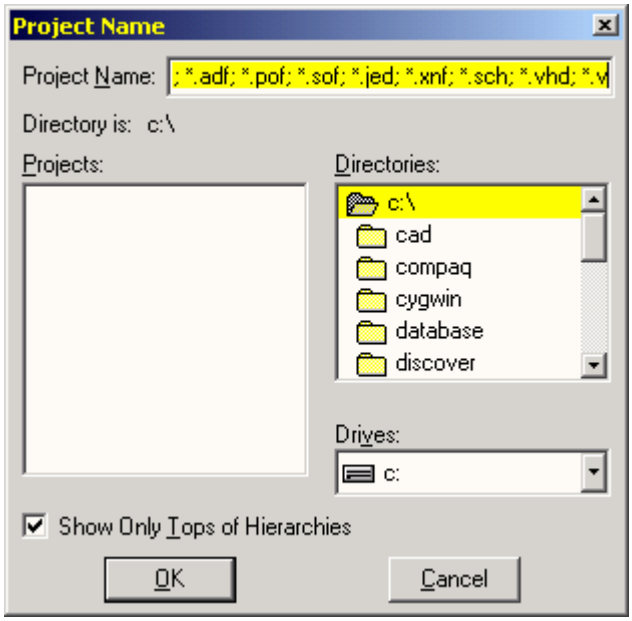


Figure 5 Project 명의 입력

사용자는 "Drives:"항목과 "Directories:"항목을 적절히 활용하여 서두에서 언급한 작업디렉토리 (L:\max2work\exp)로 이동을한다. 그다음으로 "Project Name:"항목에 여기서 작업할 Project 명인 8counter 를 입력한다.

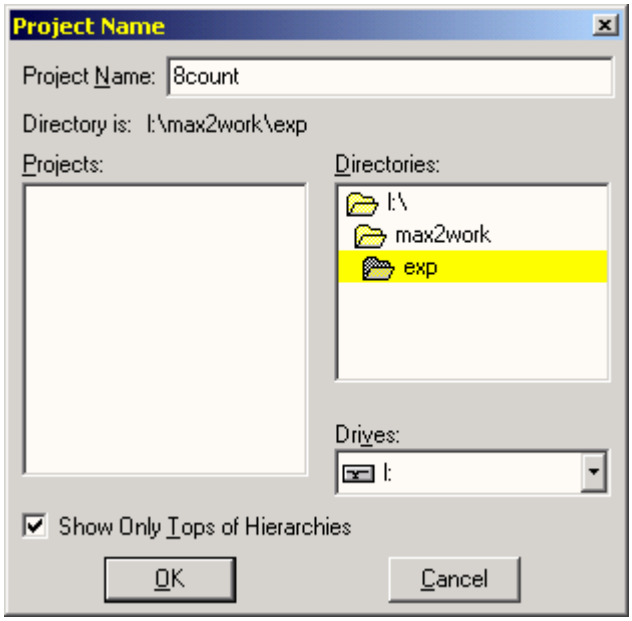


Figure 6 Project 명의 입력 (8count)

OK 버튼을 마우스로 클릭하면 그림 7 과 같이 타이틀바에 Project 명이 설정된다.

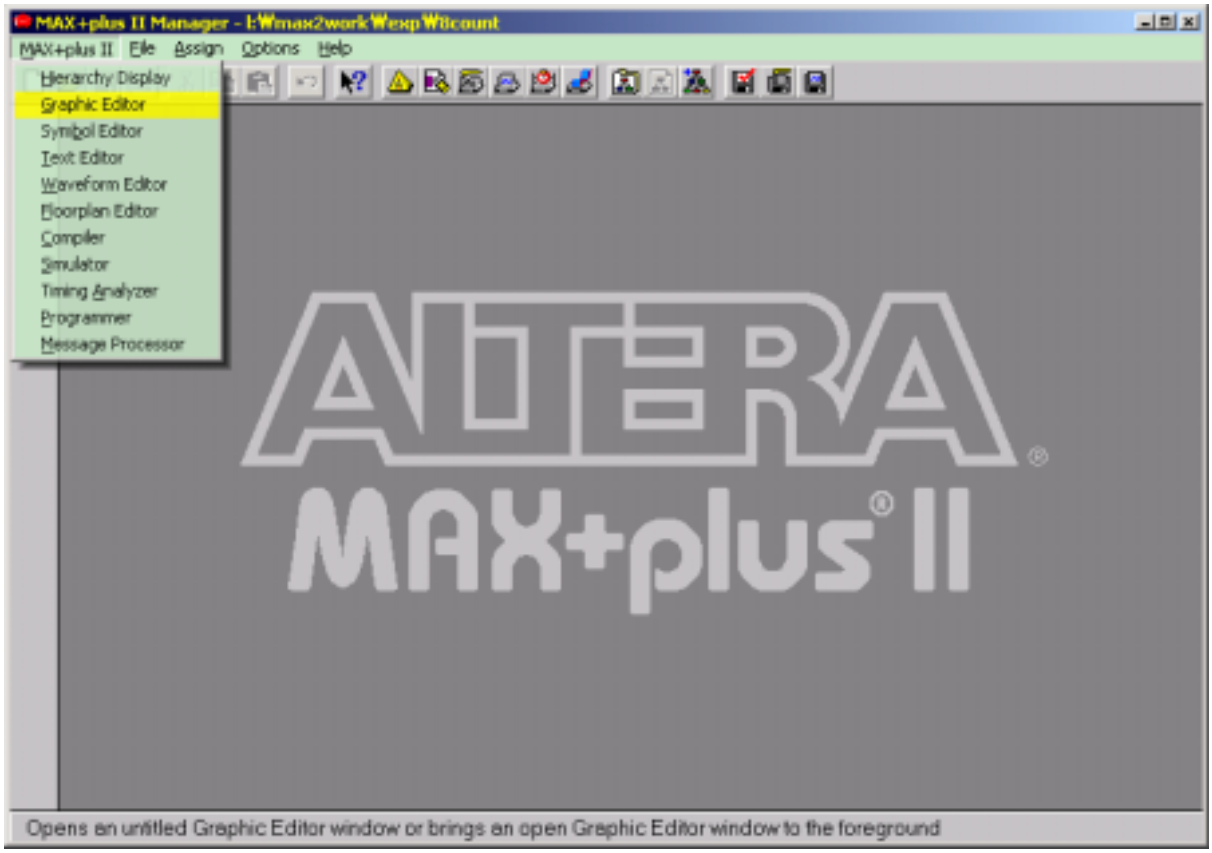


Figure 7 [MAX+plus II]의 Submenu 인 [Graphic Editor] 선택

그 다음 단계로 Schematic 을 그리기 위한 사전단계로 새로운 File 을 생성하는 단계이다. 그림 7 과 같이 MAX+plus II 의 Submenu 인 Graphic Editor 를 선택하여 그림 8 과 같은 새로운 파일을 만든다.

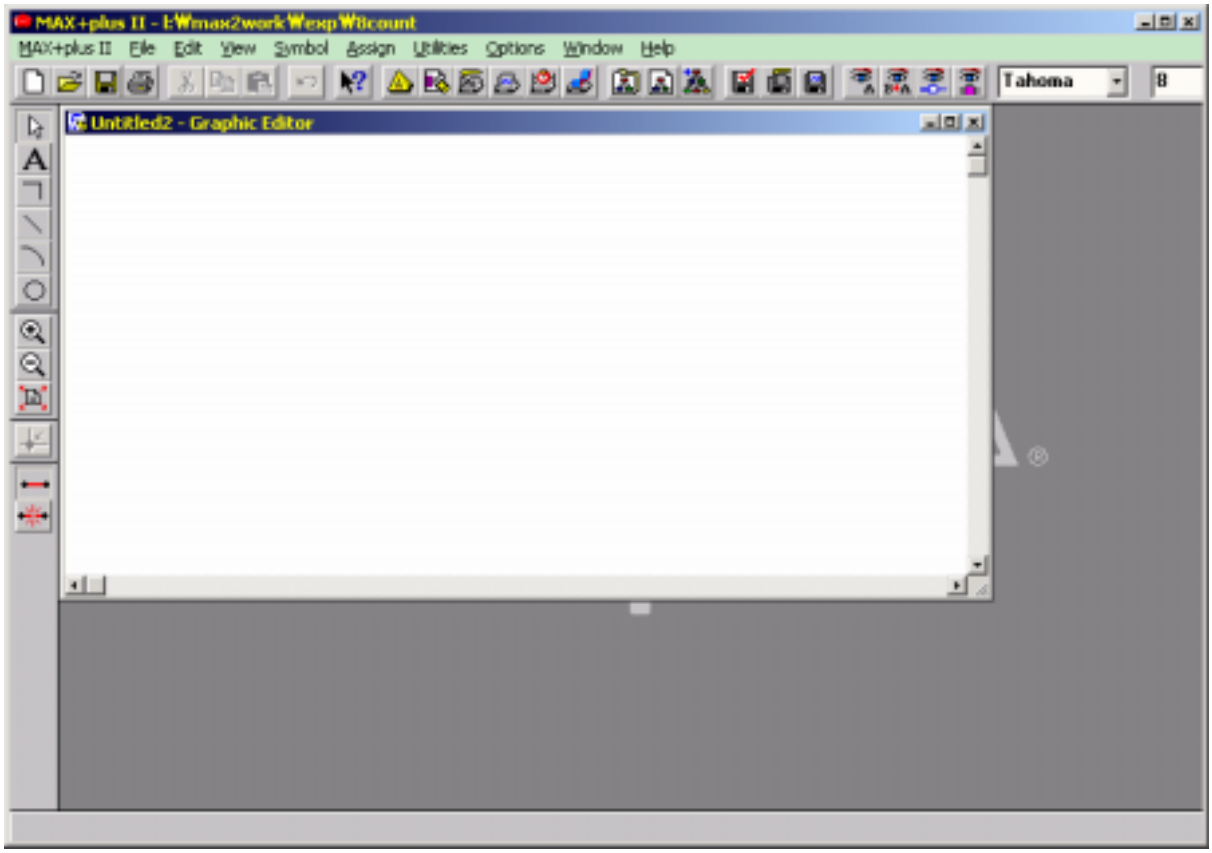


Figure 8 Graphic Editor 화면

Graphic Editor 가 화면에 나타나면 파일명의 저장을 위하여 풀다운 Menu 중에서 [File]의 서브메뉴인 [Save As….]를 선택한다. 파일명을 저장하기 위한 화면이 나타난다.

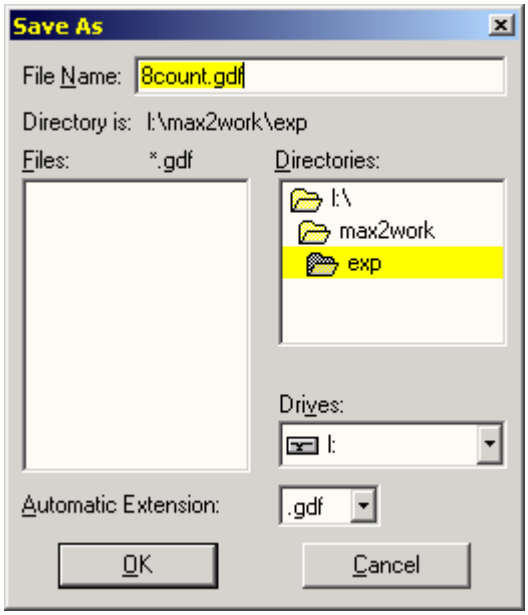


Figure 9 새롭게 열린 GDF File 을 저장한다.

파일명을 저장하기 위한 화면이 나타나면 앞서 설정한 Project 명이 자동적으로 저장할 파일명으로 나타나며, 확장자도 gdf 로 자동설정된다. 위의 예에서는 8count.gdf 로 자동 설정이 되어있슴을 확인할수 있다.

화면에서 OK 버튼을 마우스로 클릭하면 그림 10 과 같이 사용자가 설정한 파일명이 Graphic Editor 화면에 나타난다.

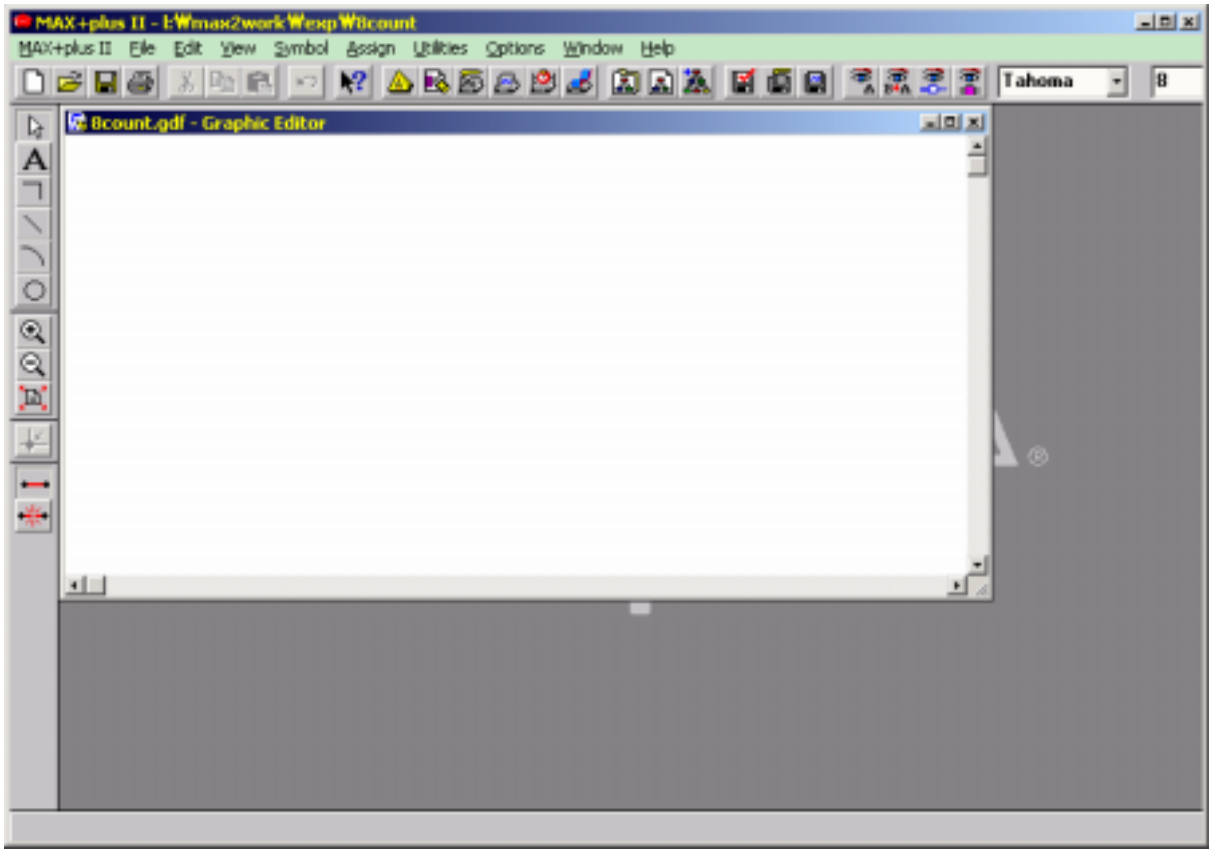


Figure 10 8count.gdf 로 저장된 Schematic 편집화면

Graphic Editor 화면을 Symbol 입력및 배치의 편의를 위해 화면 전체크기로 확대한다. 다음 사용자가 설계할 심볼을 Graphic Editor 화면에 불러오기 위하여 그림 11 과 같이 풀다운 메뉴에서 [Symbol]의 서브메뉴인 [Enter Symbol…. Double–Click]을 선택하거나, Graphic Editor 화면의 임의의 지점에 마우스 포인터를 위치해 놓고 마우스의 왼쪽 버튼을 더블클릭하면 심볼명을 입력하는 화면이 나타난다.

4-Bit Binary Up/Down Counter 를 불러내기위해 Symbol Name 에 4count 라고 기입후 Enter Key 를 치거나 OK 버튼을 클릭한다.

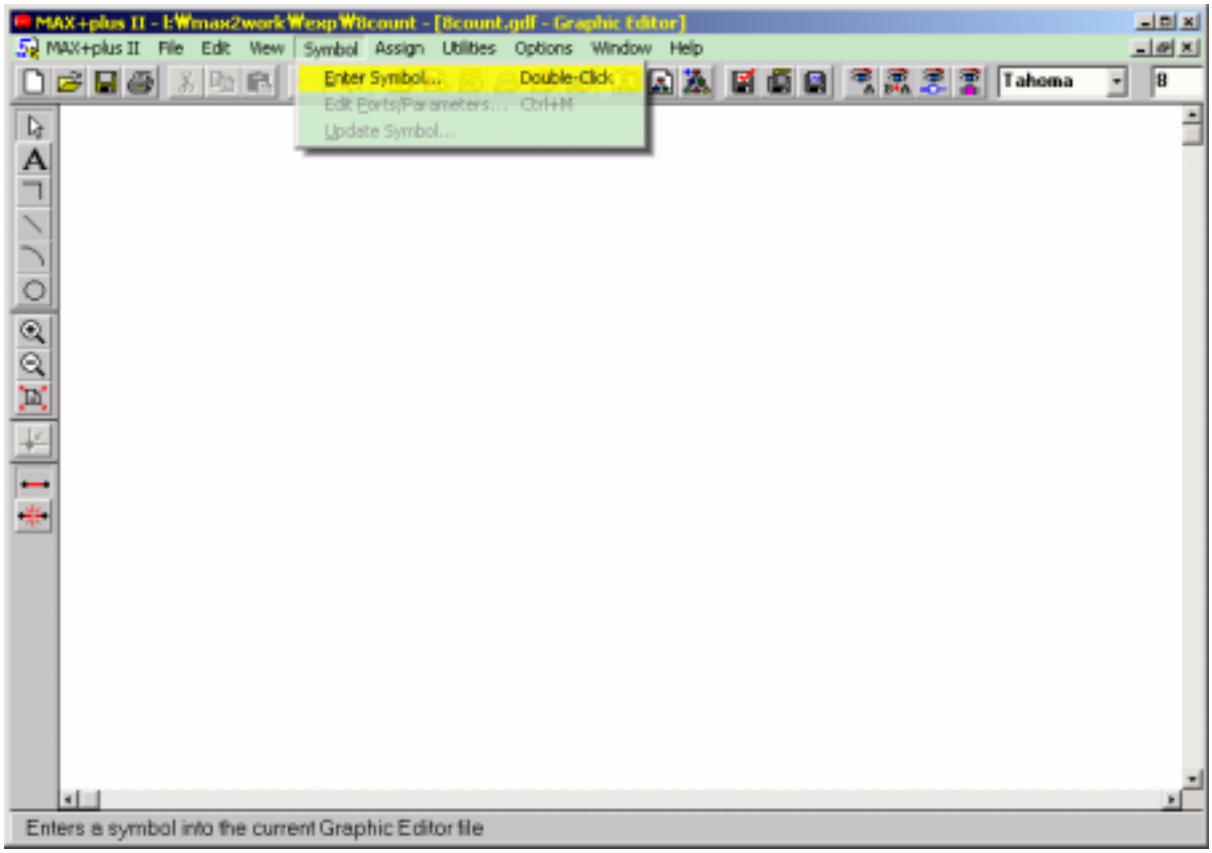


Figure 11 [Symbol] 메뉴의 [Enter Symbol… Double-Click]메뉴 선택화면

초보자를 위한 Enter Symbol 의 추가 설명

만약 처음으로 MAX+plus II 를 사용하는 사람은 이러한 Symbol Name 의 입력에 대해서 생소할 수밖에 없기 때문에 여기서는 이에 대한 입력을 Symbol Libraries 를 이용해 입력하기로 한다. 만약에 어느 정도의 Symbol Name 을 알고 있다면 아주 쉽게 Symbol Name 을 직접 입력하여 원하는 심벌을 추가할 수 있을 것이다.

만약 입력 하고자 하는 Gate 의 이름을 모를 경우에는 그림 12 에서 Symbol Libraries 내에 있는 **\maxplus2\max2lib\prim** 이나 **\maxplus2\max2lib\mf** 또는 **\maxplus2\max2lib\mega_lpm** 등의 Symbol Libraries 분류명들을 마우스로 더블 클릭하면 각각의 Symbol Libraries 내에 있는 Symbol Name 이 Symbol Files:에 나타나는데 이들 이름을 보고 원하는 Symbol 을 입력할수도 있다.

각 분류들에 들어있는 전체 symbol 들에 대한 자세한 설명은 MAX+plus II 의 HELP Menu 중 Megafunctions/LPM, Old-Style Macrofunctions, Primitives 항목을 참조하면 된다.

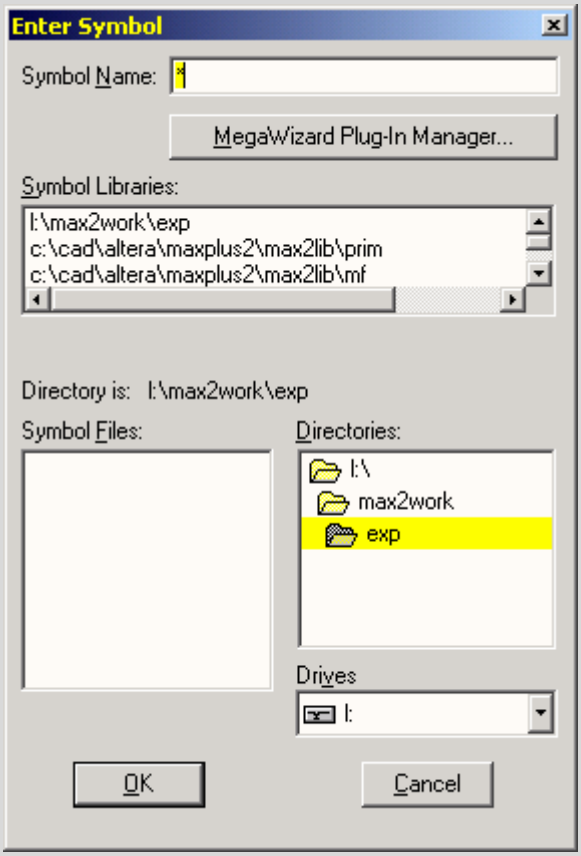


Figure 12 Symbol Name 의 입력

그림 12 에서 **\maxplus2\max2lib\prim** 을 선택하여 더블클릭을 하면 그림 13 과 같이된다.

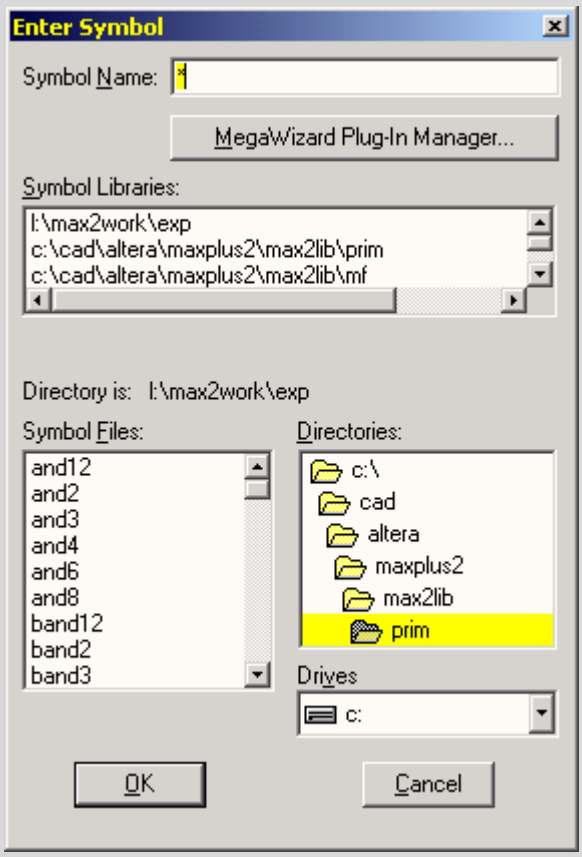


Figure 13 Primitives Libraries 에 포함된 Symbol 들

그림 13 에서 Symbol Files 내에 있는 원하는 Symbol 을 찾기 위해서는 Symbol File Box 의 우측에 있는 스크롤을 마우스로 누르고 이동하면 원하는 Symbol 을 찾을 수 있고 이때 마우스로 원하는 Symbol 을 선택한 후에 OK 를 선택하면 Graphic Editor 화면 상에 Symbol 을 추가할 수 있다.

그림 13 의 Symbol Files Box 내에는 여러 가지의 각종 Gate 들이 있으며 **\maxplus2\max2lib\prim** 에 포함된 symbol 들은 다음과 같이 여러 종류가 있다.

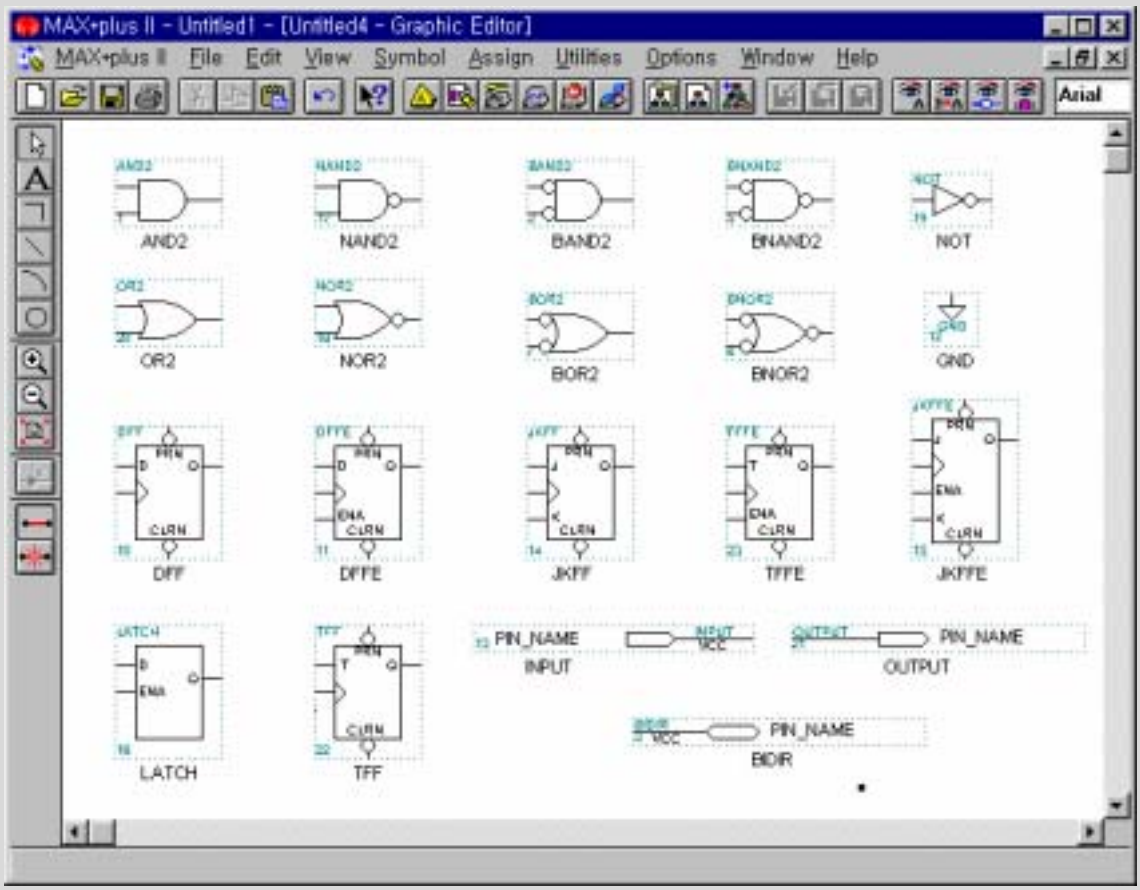


Figure 14 "\maxplus2\max2lib\prim"에 포함된 symbol 들

그림 14 에서 AND 뒤의 숫자는 입력 Port 의 수를 나타낸다. 예를 들어 2 입력 AND Gate 는 AND2 로 표현하며, 2 입력 OR Gate 는 OR2 로 표현된다. 또한 그외의 기호들도 우리가 일반적으로 생각하는 용어들과 크게 다리지 않다. 예를 들어 D Flip-Flop 은 DFF 로 표현하고, JK Flip-Flop 은 JKFF 로 표현하는것이 그것이다.

이와 같은 심벌을 여러 번 입력하다 보면 요령이 생기게 되는데 이것은 그림 12 나 그림 13 에서 Symbol Name 의 *위치에 Symbol 명을 바로 입력한 후 OK 를 선택하면 쉬운 방법으로 Symbol 을 입력할 수 있다. 그러나 이와 같은 방법은 여러 번 Symbol 을 입력하는 과정에서 터득되므로 사용자는 이와 같은 Symbol 의 입력을 많이 해보기 바란다.

그림 14 에서 INPUT 은 최종적으로 구현되는 PLD 의 외부 PIN 에 연결되는 입력 Port 를 나타내며 OUTPUT 은 PLD 의 외부 PIN 에 연결되는 출력 Port 를 나타낸다. 그러나 BIDIR 은 Data bus 와 같은 경우에 사용되는 것으로써 High, Low, High Impedace 와 같이 TRI state 가 필요할 경우에 사용되는 것으로 PLD 의 외부 PIN 에 연결되는 입출력 PIN 을 나타낸다.

GND 는 로직적으로 "0"을 나타내며 VCC 는 로직적으로 "1"을 나타낸다. 아울러 XOR 는 배타적 논리합을 나타내고 TRI 는 TRI state buffer 를 나타낸다.

또한 그림 3 의 Symbol Files Box 내의 **\maxplus2\max2lib\mf** 에 대한 기호는 그림 15 와 같이 고기능의 Symbol 이 많이 있다.

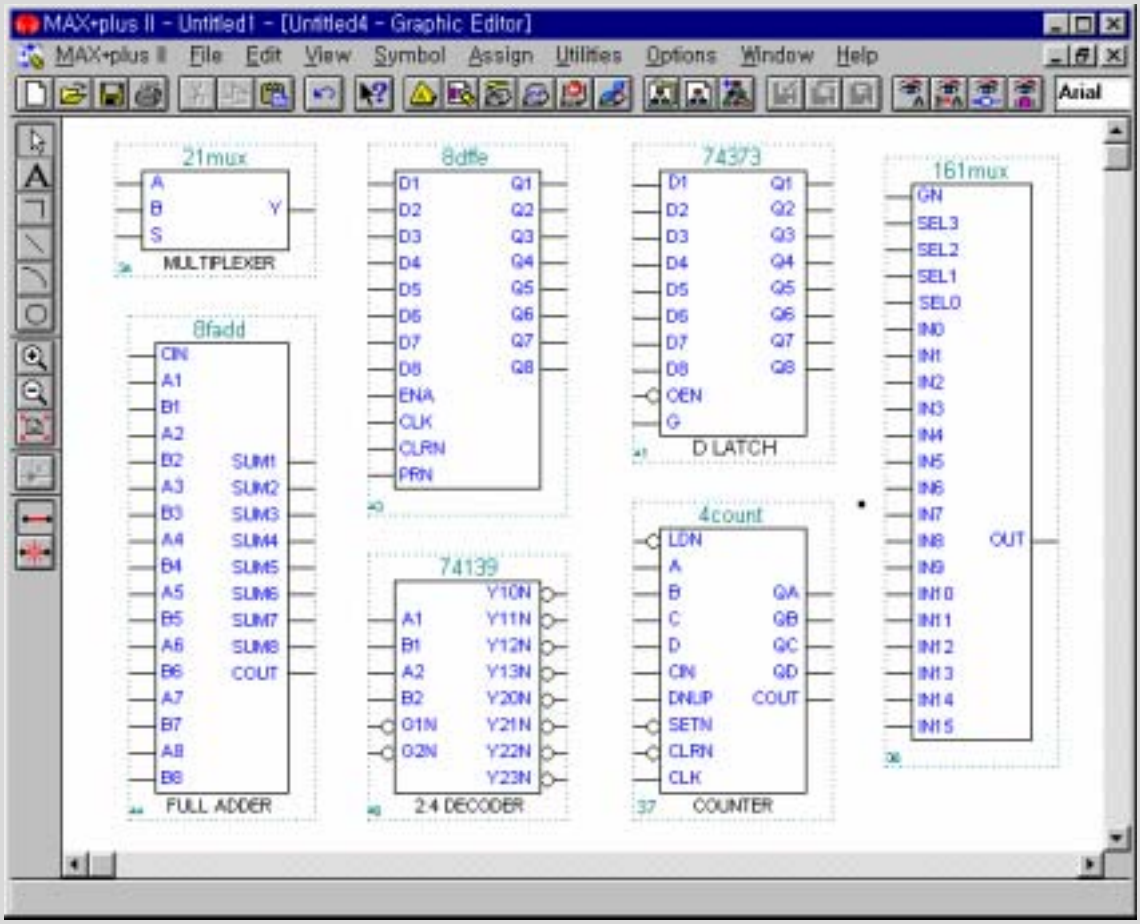


Figure 15 "\maxplus2\ max2lib\mf"에 포함된 symbol 들

Symbol Name 에 4count 를 입력한후 OK 버튼을 누르면 그림 16 과 같은 그림이 된다.

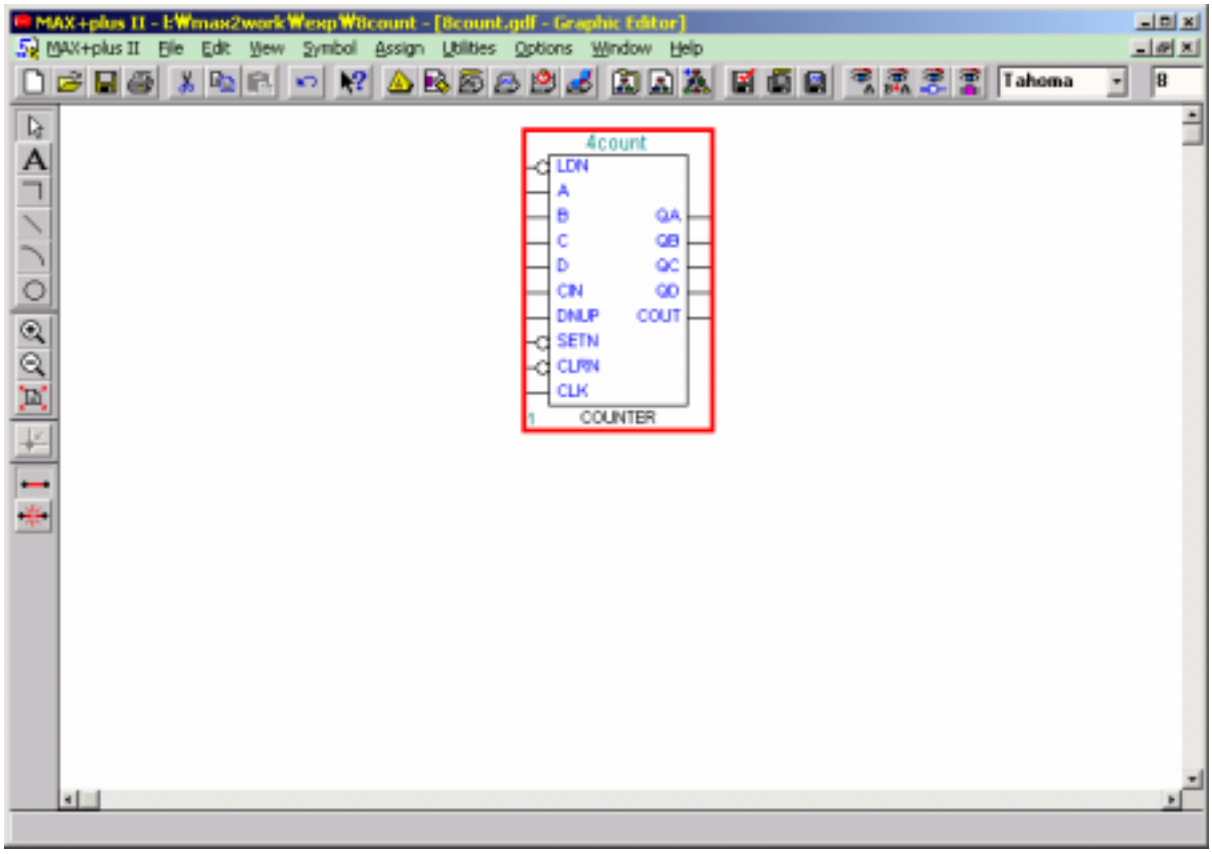


Figure 16 화면상에 나타난 4count 심볼

화면상의 4count 심볼에 마우스의 포인터를 위치해놓고 마우스의 오른쪽 버튼을 클릭하면 Copy 라는 메뉴가 나타난다. 이메뉴를 클릭한 다음 다시 마우스를 임의의 지점에 위치한 다음 오른쪽 마우스 버튼을 클릭하면 Paste 라는 메뉴가 나오고, 이메뉴를 클릭하면 지금의 4count 가 **복사된다**.

위의 방법외에도 Ctrl Key 를 누른 상태로 4count Symbol 을 마우스의 왼쪽버튼으로 클릭, Drag 를 하여 원하는 위치에서 왼쪽버튼을 띄면 위의 방법과 동일한 현상을 보여준다. (복사가 된다는 얘기.)

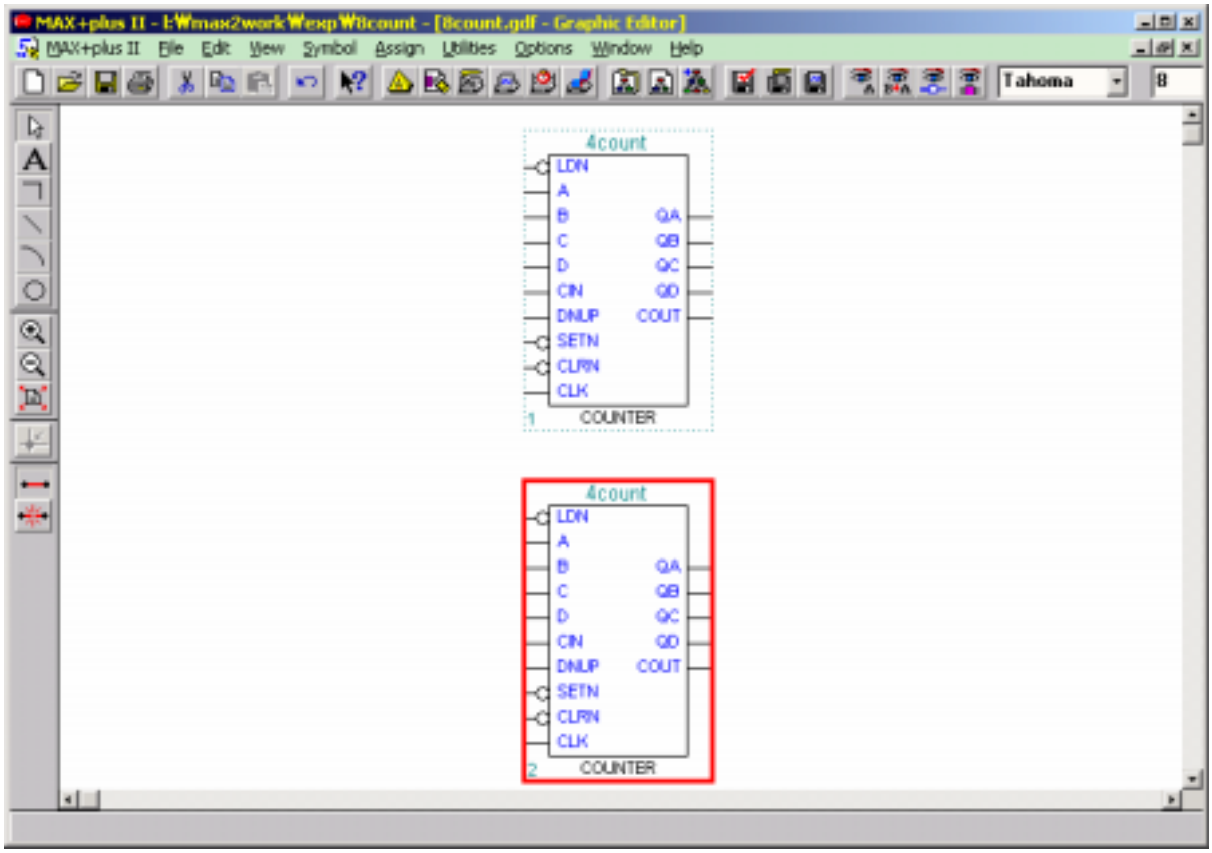


Figure 17 복사된 4count

지금의 과정을 반복하여 input 3 개와 output 9 개를 만들어 보자. 그림 18 은 지금까지 진행된 화면이다.

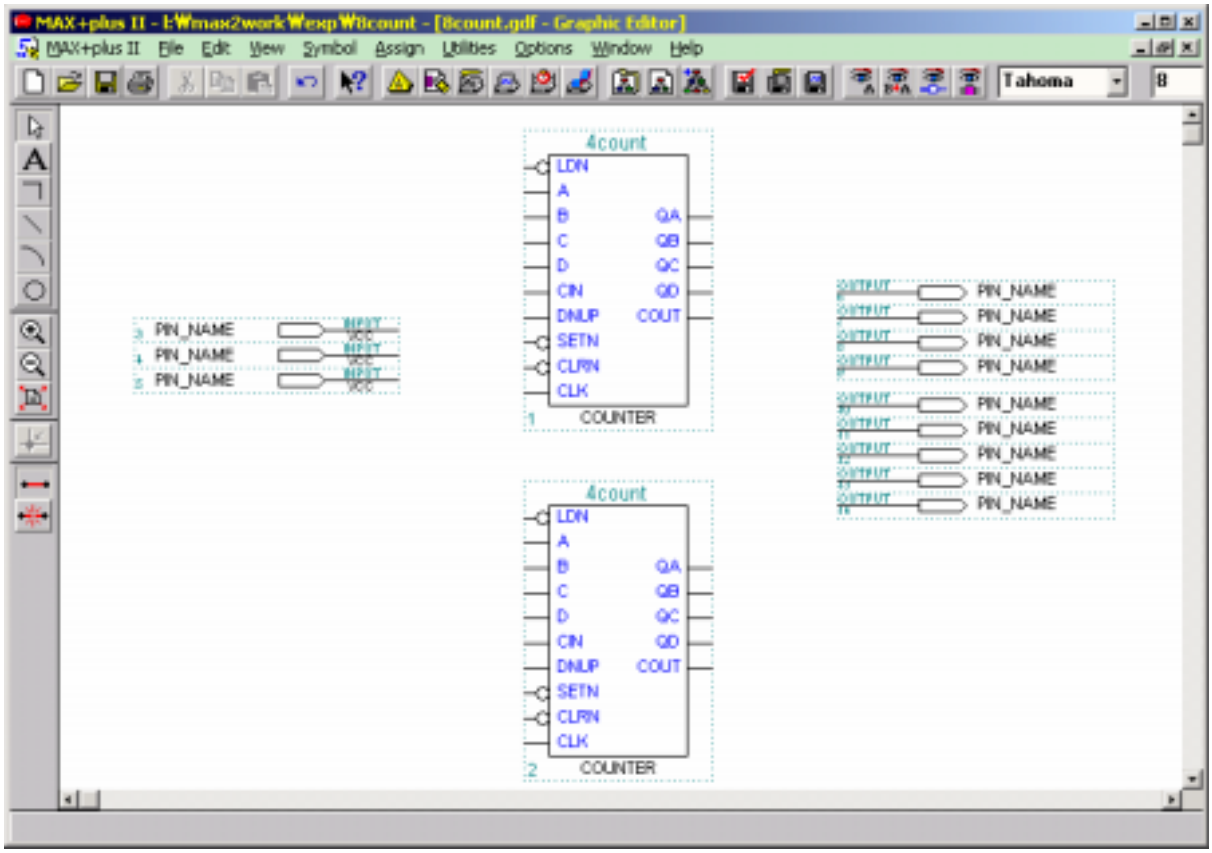


Figure 18 Line 을 연결하기전의 8count

앞에서 표시한 심벌과 아래에 설명된 편집 도구를 이용하여 그림을 그린다. 이때 마우스를 심벌의 입출력 Port 위치로 이동하면 심벌 위에『+』표시가 나타나는데 이때 마우스의 왼쪽 버튼을 누른 상태에서 마우스를 이동하면 빨간 색의 실선이 나타나 선으로 연결되고 있음을 보여주고 있다. 만약에 어느 지점에서 끝내고 또 다른 곳에서 새로운 선을 그리고 싶을 때는 마우스의 왼쪽 버튼을 떼고 난 후 다시 마우스의 왼쪽 버튼을 누르고 원하는 위치로 이동하여 선을 그릴 수 있게 된다. 이때 마우스의 이동시에 점선으로 된『+』표시가 나타나 화면의 편집을 도와준다. 이와 같은 방법으로 그려진 화면은 그림 19 와 같다.

-  - 화면 확대
-  - 화면 축소
-  - 전체화면 표시
-  - 연결점(Connection dot)의 삽입 또는 삭제
-  - Symbol 의 이동시 연결선도 같이 이동 (Rubberbanding function on)
-  - Symbol 의 이동시 연결선은 그대로 분리된 상태로 유지 (Rubberbanding function off)

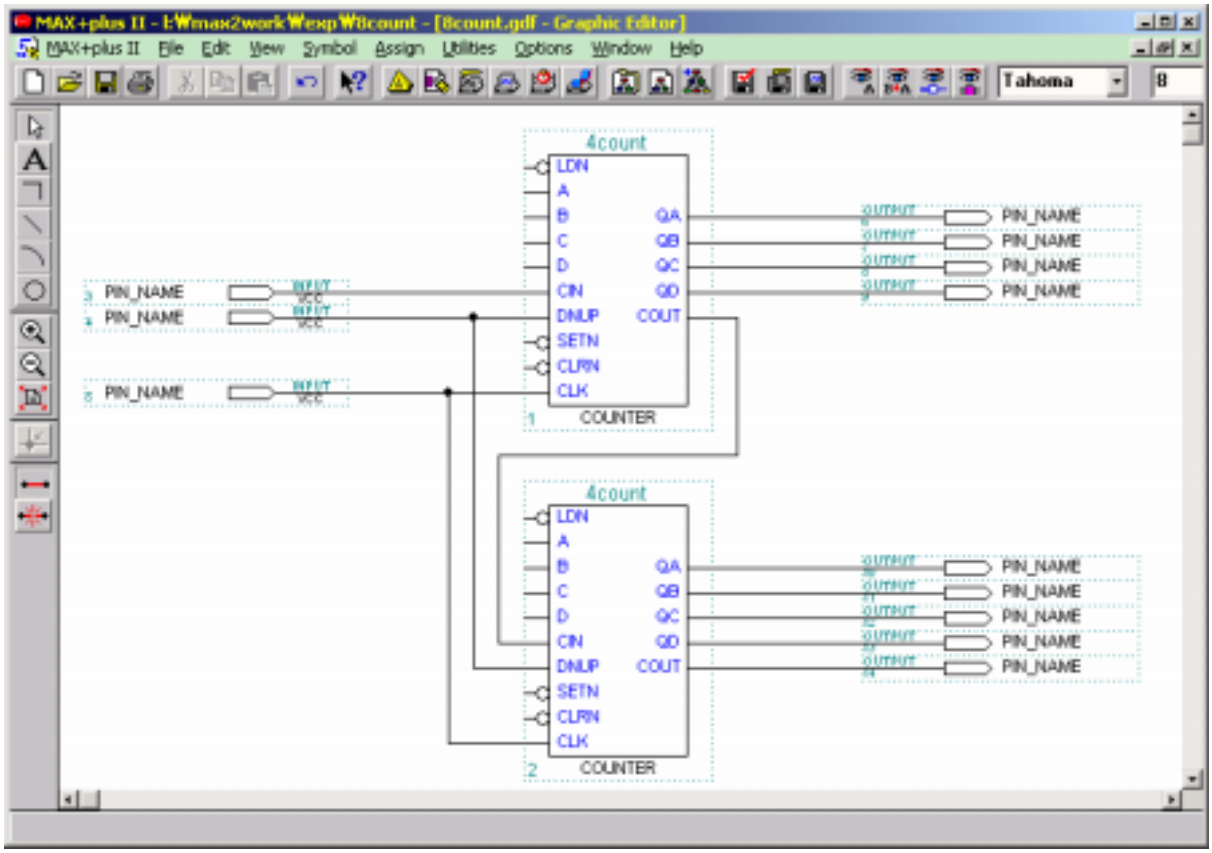


Figure 19 심볼을 이용한 Graphic Editor 완성

위의 그림 19 는 아직 INPUT PORT 와 OUTPUT PORT 의 이름이 정의되어 있지 않는 상태이므로 이름을 정의하기위한 작업이 필요하다.

해당 입·출력 Port 의 PIN_NAME 위에 마우스의 왼쪽 Button 을 두 번 클릭 하면 PIN_NAME 위에 검정색의 바탕이 생기며 여기에 원하는 이름을 입력한다. 여기서는 실험을 위해 입력에 대해서는 cin, dndp, clk 로 설정하며 또한 출력은 q1, q2, q3, q4, q5, q6, q7, q8, cout 으로 각각 이름을 정한다. Pin 이름을 입력하는 다른 방법은 그림 20 과 같다.

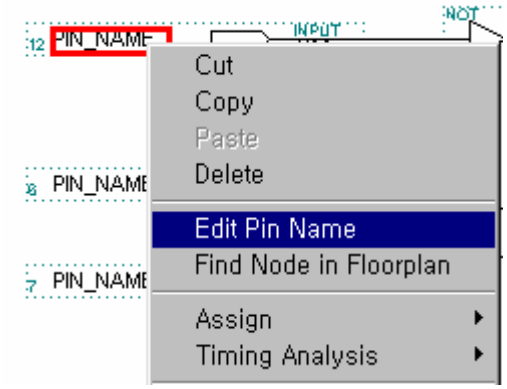


Figure 20 Pin Name 입력

여기서 알 수 있듯이 먼저 마우스로 해당 PIN_NAME 을 선택한 후 오른쪽 마우스를 누르면 그림 20 이 나타나며 여기서 Edit Pin Name 을 선택하면 해당 PIN_NAME 의 위치가 검정색 바탕으로 되어 원하는 Pin Name 을 입력할 수 있다.

이와 같은 방법으로 입력된 PIN_NAME 및 완성된 회로는 그림 21 과 같다.

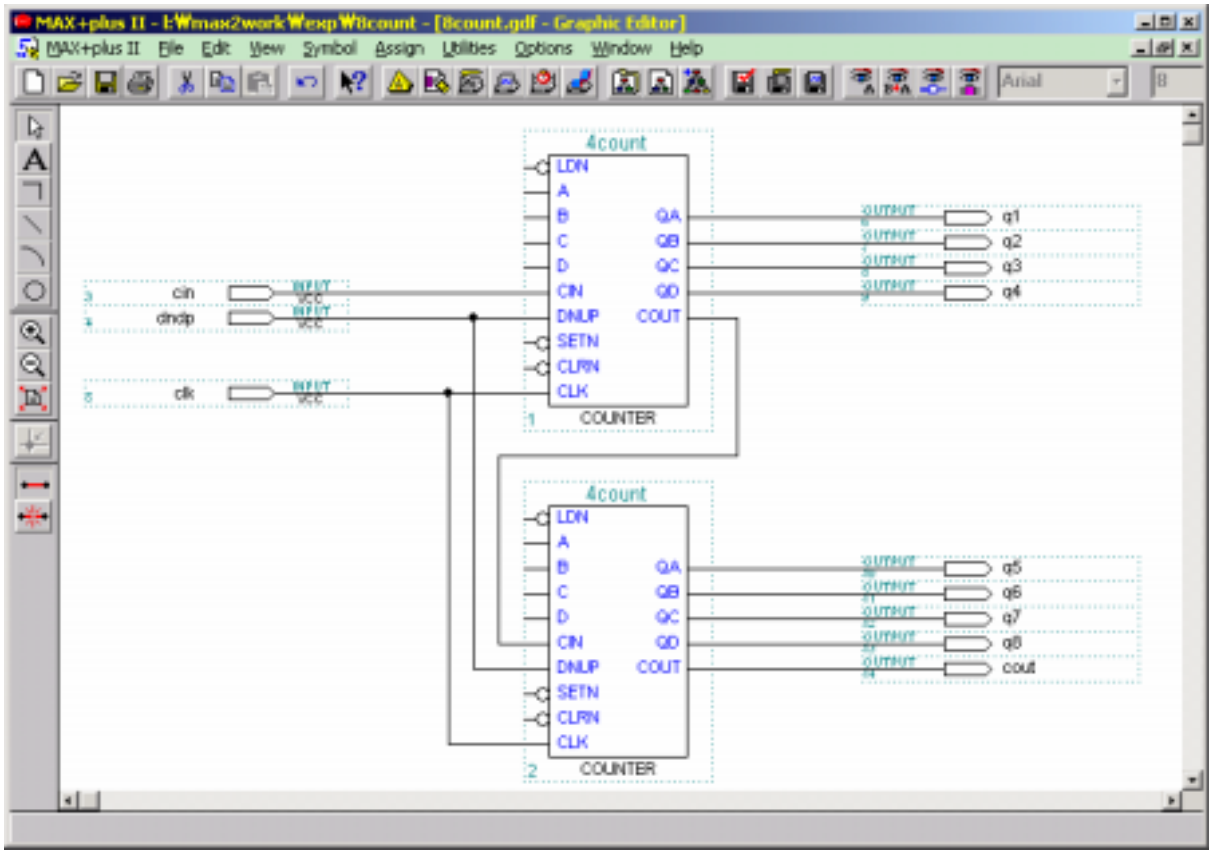


Figure 21 전체 Pin Name 입력

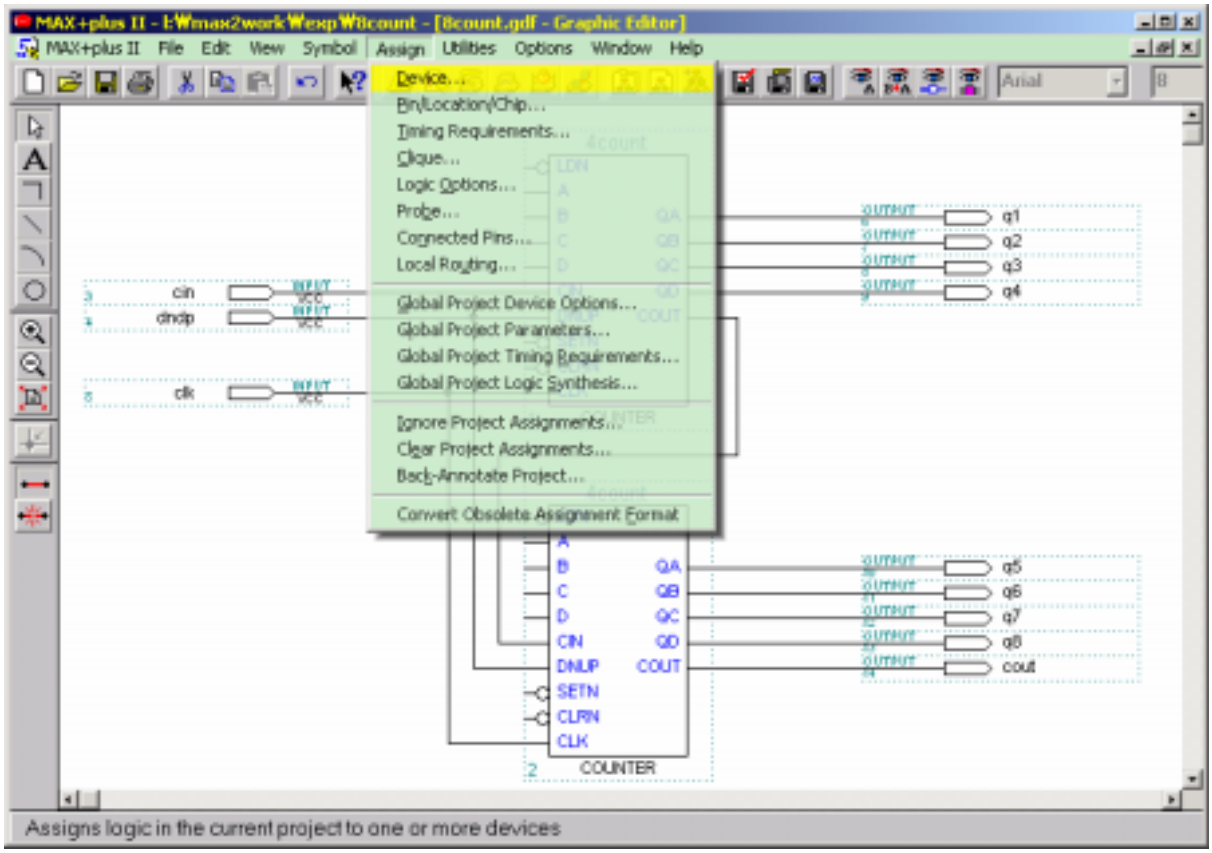


Figure 22 PLD DEVICE 선택을 위한준비

현재 상태는 아직 사용할 Device 를 선택하지 않았으므로 Device Family 가 AUTO 로 미리 설정되어 있다.

이를 위해 그림 22 와 같이 Assign Device 를 선택하면 그림 23 과 같은 Device 선택 화면이 나타나며 이때 FLEX10K Series 중 EPF10K10QC208-4 를 선택한 후 OK 를 선택한다.

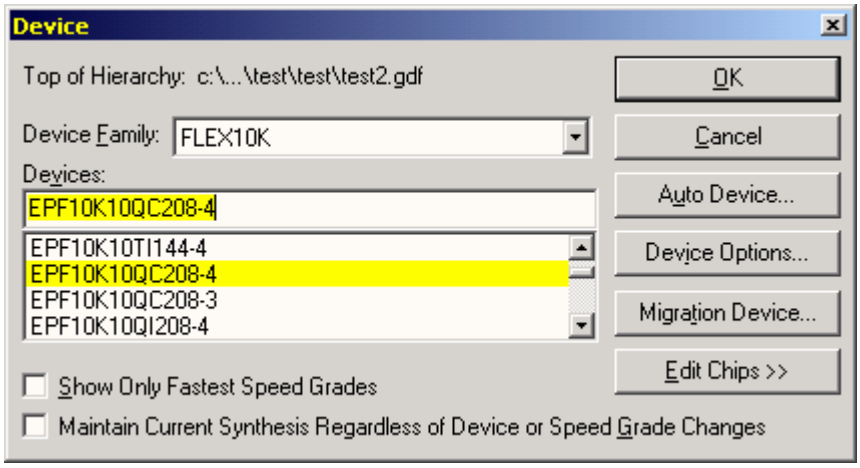


Figure 23 DEVICE 선택

그러나, 그림 23 에서 다음과 같이 선택되어 있으면 EPF10KQC208-4 를 선택할 수 없고 다만 EPF10KQC208-3 밖에 보이지 않을 것이다. 이것은 그림 24 와 같이 Fastest Speed Grades 만을 보여주도록 기본적으로 선택되어있기 때문이다.

Figure 24 속도가 빠른 Device 만 보이도록 선택하는 Option 항목

이 경우에는 그림 23 에서 "**Show Only Fastest Speed Grades**"를 Disable 시키면 EPF10KQC208-4 가 나타나며 이때 EPF10KQC208-4 를 선택한 후에 OK 를 누른다.

참고적으로 FLEX10K10 Series 에서는 EPF10K10QC208-3 이 가장 Speed 가 빠르며 그다음은 EPF10K10QC208-4 이다. 상대적으로 속도가 빠르면 회로 전반적인 Performance 에는 이득이 되지만 반대로 Chip 의 가격이 상승하는 요인이 된다.

이번 단계는 완성된 Schematic 에 잘못된 곳이 없는지는 검사하는 Design Rule Check 단계이다. 그림 25 를 보자.

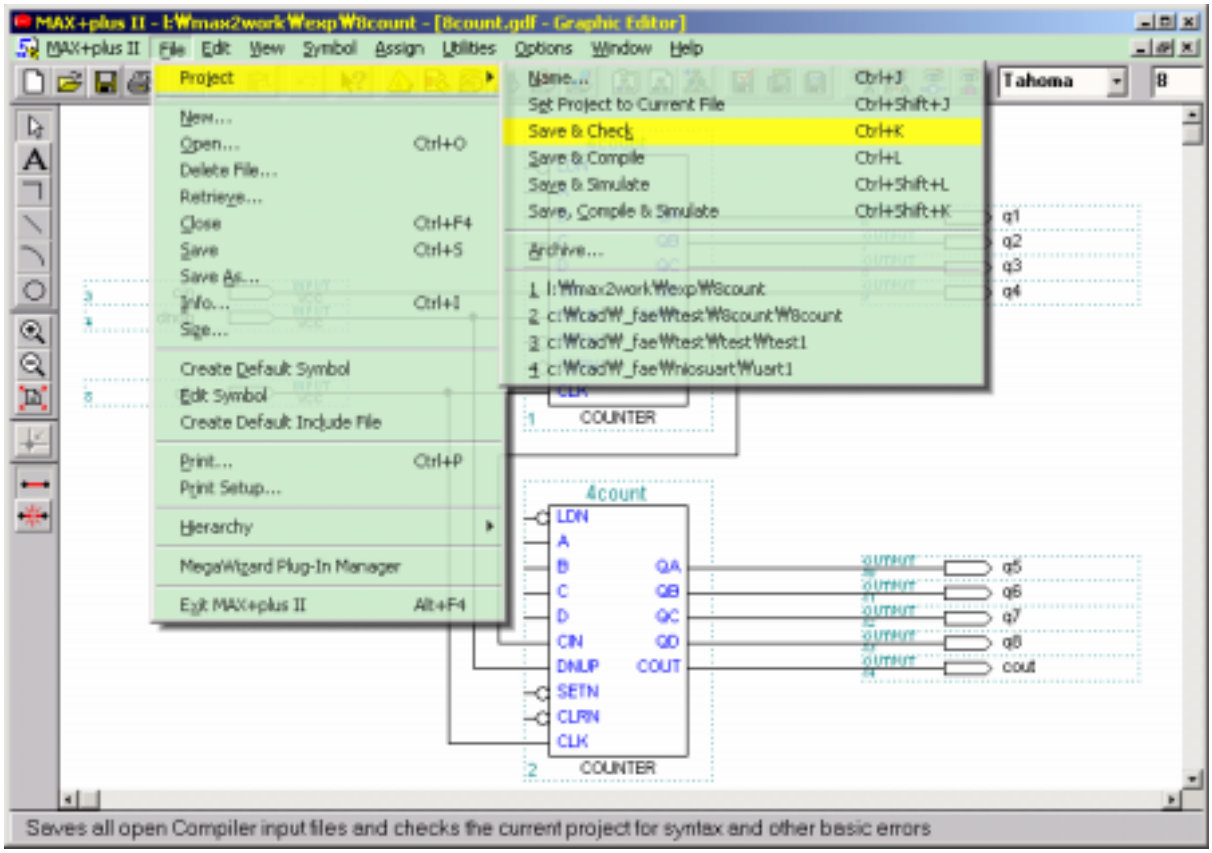


Figure 25 Design Rule Check

File -> Project -> Save & Check 를 차례로 선택하면 지금까지 그린 회로에 어떠한 이상이나

잘못된 점이 있는지는 검사하여 우리에게 알려주는 기능을 수행한다. 그림 26 과 27 은 진행과정을 나타내주는 그림들이다.

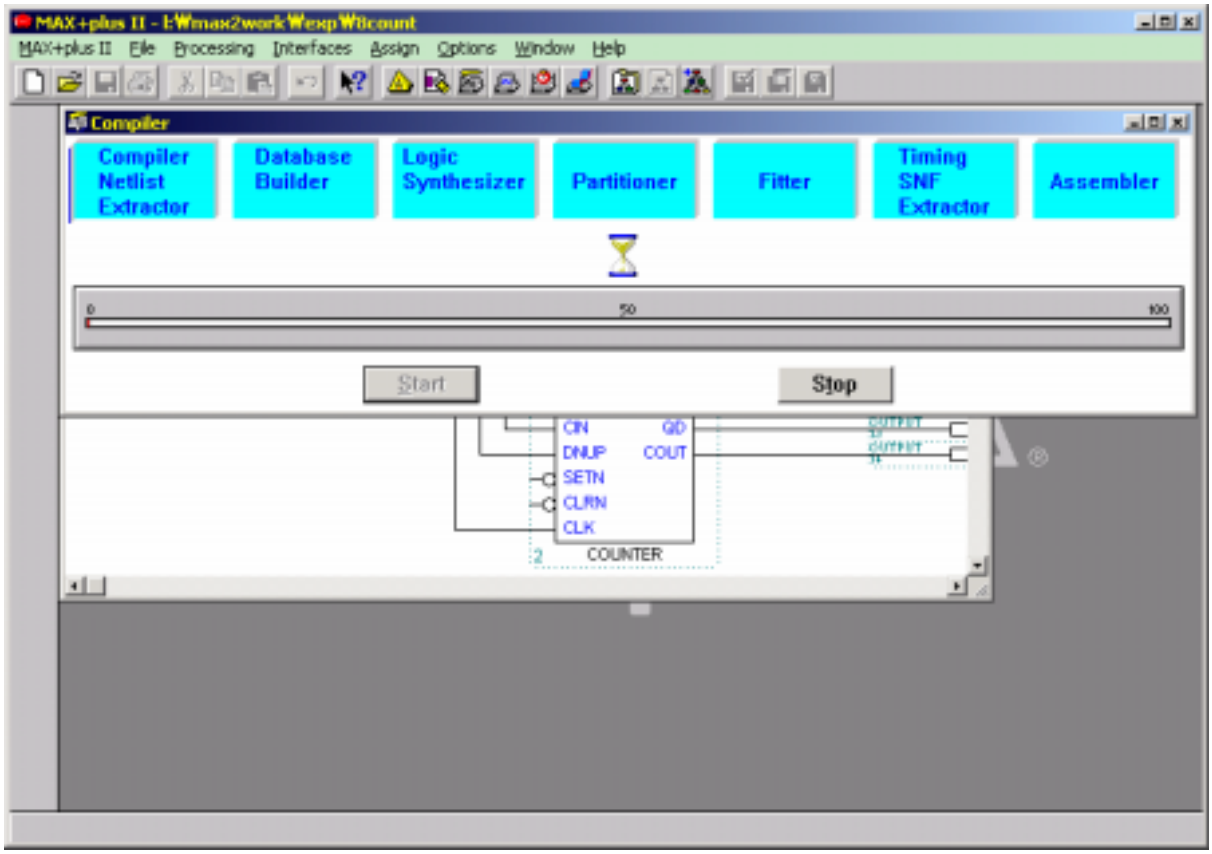


Figure 26 진행중인 Design Rule Check

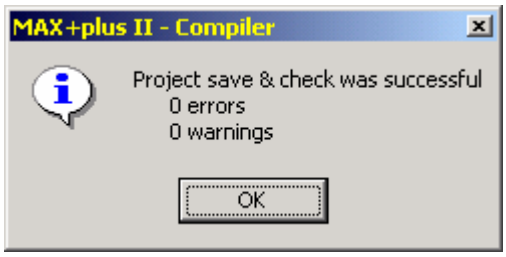


Figure 27 이상없이 Design Rule Check 가 완료되었슴을 보여준다.

만약 이단계에서 어떠한 Error 나 Warning Message 를 만난다면 Message Processor 창에 나타나는 Message 를 참고하여 이상한 부분을 수정하는 작업이 필요하게 된다.

참고로 Error 는 반드시 해결을 해야 다음단계로 진행을 할수있으므로 반드시 제거를 해야하는 부분이고, Warning 은 해당 Message 를 보고 불필요한 Warning 인지 고려대상에 포함될 Warning 인지를 스스로 판단하여 해결을 해야한다.

3

Compiler 를 통한 Design 상태 점검 및 Fitting

앞장에서 설명한 Graphic Editor 상에서 설계된 회로를 검증하기 위하여 다음과 같이 컴파일 모드로 들어간다.

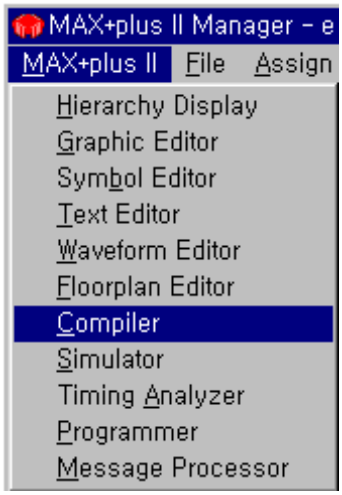


Figure 28 Compiler 실행하기

MAX+plus II 컴파일러는 일련의 모듈 (Module)과 Utility 로 구성된다. 이와 같은 것은 Project 에 대한 오류 (Error)를 검출하며 Logic 을 Synthesize 하여 하나의 Project 를 하나 이상의 ALTERA Chip 으로 Fitting 하는 과정을 행하게 된다.

Project 를 컴파일 하기 위해서는 Design File 이 필요한데 이 File 은 앞장에서 만들었기 때문에 바로 컴파일 과정에 들어갈 수 있다.

이에 대한 Compiler 의 초기 화면은 그림 29 와 같다.

Figure 29 Compiler 초기화면

위의 그림에서 Start 를 클릭하면 현재의 Project 에 대한 모든 입력 파일을 자동적으로 컴파일하게 된다. 이러한 Compiler 는 다음과 같은 모듈과 Utility 를 이용하여 Compile 을 한다.

- Compiler Netlist Extractor - 현재의 프로젝트에 있는 각각의 Design 파일을 하나 이상의 Binary Compiler Netlist 파일(.cnf)로 만들게 된다.
- Database Builder - 현재 프로젝트의 전기적인 연결 상태를 나타낸다.
- Logic Synthesizer - 로직 상태를 점검하여 연결되지 않은 node 는 삭제한다.
- Partitioner - 설계되는 Project 가 하나의 Device 에 들어갈 수 있는지의 여부를 알아보고 불가능할 경우에는 여러 개의 Device 로 나누어준다.
- Fitter - 각각의 Logic function 을 최상의 Logic cell 로 할당한다.
- Timing SNF Extractor - 완전히 최적화된 프로젝트에 대한 Timing data 를 포함하는 Timing Simulation Data File 이 만들어진다.
- Assembler - Fitter 의 로직셀(Logic cell), Pin 등을 해당 Device 에 대한 하나 이상의 Programmer Object File (.pof)이나 SRAM Object File (.sof)이 만들어진다.

-  - 이 표시는 현재 Compiler 의 진행 과정을 나타내고 있는 표시이다.

컴파일 되고 있는 상황을 그림 30 에 나타내었다.

Figure 30 Compiler 의 진행 과정

그림 30 을 보면 현재 Compiler 는 Partition 의 과정을 수행하고 있음을 알 수 있다. 각각 Module 의 진행이 완료되면 그림 31 과 같은 Windows 가 나타난다. 이와 같이 표시됨으로써 작성된 회로가 정상적으로 설계되었음을 알려준다.

Figure 31 Compile 완료

그러나, 다른 경고 (warning)나 에러 (error)가 발생하면 ALTERA 의 Help 를 이용하여 틀린 회로를 바로잡고 다시 위의 과정을 되풀이하여 그림 31 과 같이 경고나 에러가 없으면 다음의 과정으로 넘어간다. **일반적인** 경우 앞장에서 행한 Design Rule Check 가 정상적으로 진행이 되었다면 Compile 과정은 별 무리없이 넘어간다.

참고로 간단한 모드 전환 방법을 다음에 보인다.

- 

- Compile 완료 후 이 Timing SNF Extractor 의 아래쪽 Icon 을 더블클릭하면 바로 시뮬레이션 모드로 들어갈 수 있다. (물론 Simulation 을 위한 Stimulus 가 미리 준비된 상태에서나 가능한 얘기이다.)

- 

- Compile 완료 후 이 Assembler 의 아래쪽 Icon 을 더블클릭하면 바로 PLD 에 Download 또는 Configuration 을 할 수 있는 Programmer 모드로 갈 수 있다.

4

Timing Simulation 실행하기

설계된 디지털 시스템의 로직을 검증하기 위하여 시뮬레이션을 행한다. 이러한 시뮬레이션을 하기 앞서 그림 32 와 같은 방법으로 시뮬레이션을 위한 파형을 만들기 위해 Waveform Editor 를 선택하면 그림 33 과 같이된다.

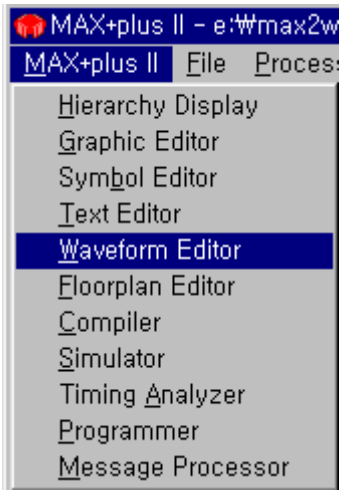


Figure 32 Waveform Editor 로 들어가기

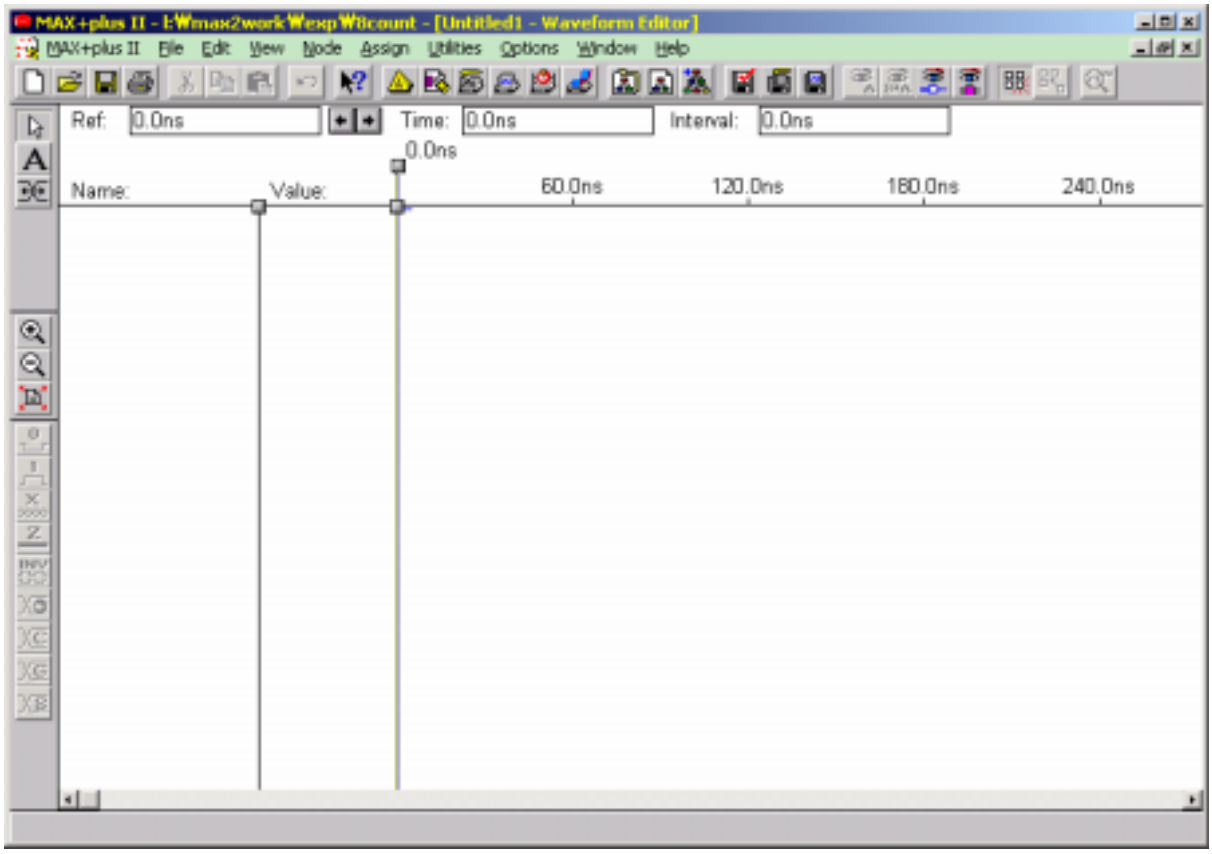


Figure 33 Waveform Editor 화면

위의 화면에서 우리가 보고싶은 입출력 Node 들을 입력하게 된다. 입력 방법은 첫째, Node -> Insert Node ; Node Name 을 모두 알고 하나씩 선택하는 방법과 둘째, Node -> Enter nodes from SNF ; SNF file 의 내용을 참조하여 Node 를 선택하는 방법이 있는데 본 설명서에서는 둘째 방법을 이용한다. 둘째 방법에 의하며 그림 35 와 같은 Window 가 생성된다.

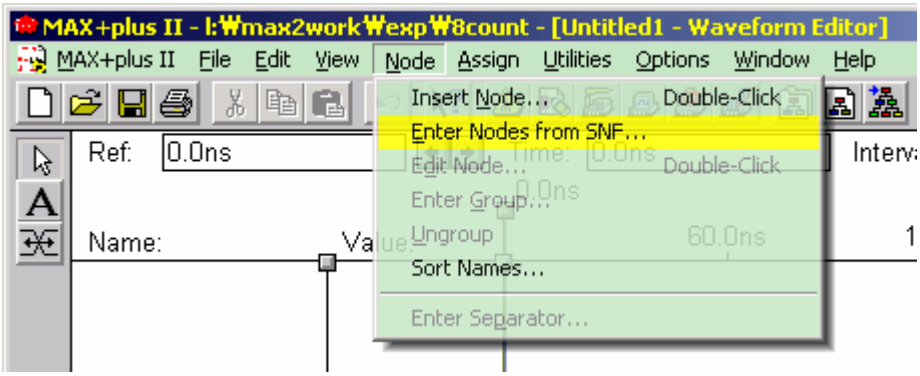


Figure 34 Node 의 서브메뉴인 Enter Node From SNF…선택

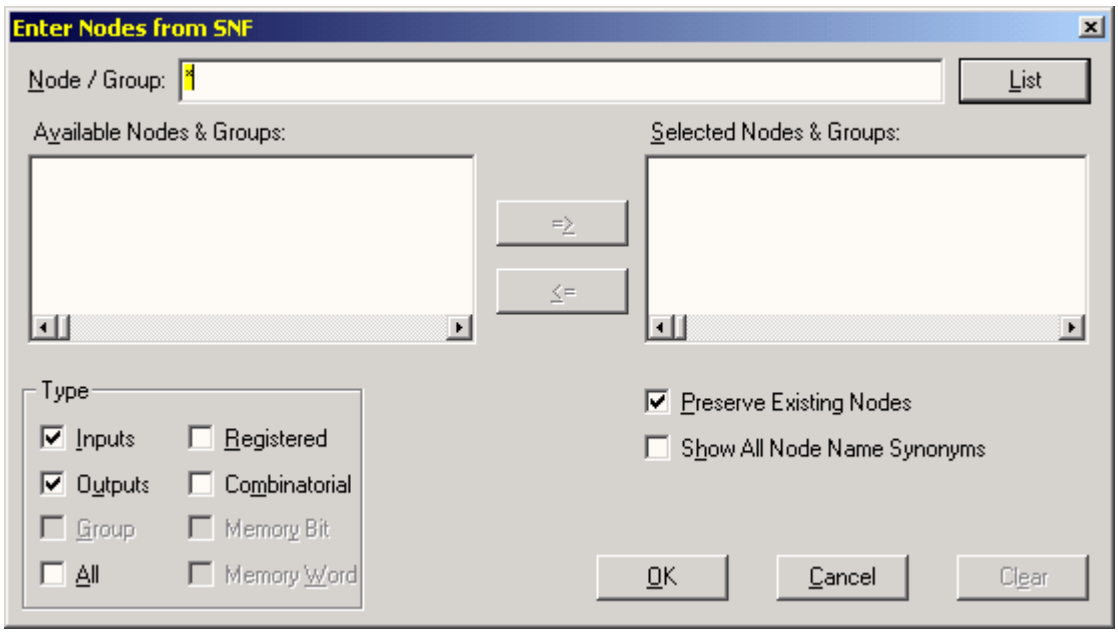


Figure 35 Enter Node From SNF 창

이와같은 Window 에서 List Button 을 누르면 회로 내에 있는 모든 node 들을 볼 수 있는데 이때 "=>" Button 을 선택하여 Node 를 선택한다. 위의 모든 과정이 끝나면 OK Button 을 누르고 Waveform 에 나타날 Node 의 설정을 마친다.

현재의 설정상태는 입출력 Port 만을 찾도록 되어있으므로 Available Nodes & Groups: 항목에 입출력 Pin 들만이 나타난다. 만약 회로 내부의 Node 들에 대한 결과도 Simulation 상에서 보고 싶다면 Type 항목에서 원하는 부분을 Check 후 재검색 (List 버튼)을 통해 해당 Node 를 추가하면 된다.

본 예제의 경우 확인하려는 Port 의 갯수가 적기때문에 이러한 방법을 사용한다. 하지만 실제 설계시 이렇게 할 경우보다는 원하는 Port 만을 선택하여 "=>" Button 으로 추가하는 방법을 사용하게 된다.

본 예제의 경우 모든 Node 가 선택되었다.

File menu 의 "**Save as**"를 선택하여 파일을 저장한다. 파일이름은 기본적으로 ***<Project Name>.scf*** 가 된다.

원하는 In/Out Node 가 설정된 Waveform Editor 에 대한 Window 는 다음과 같을 것이다.

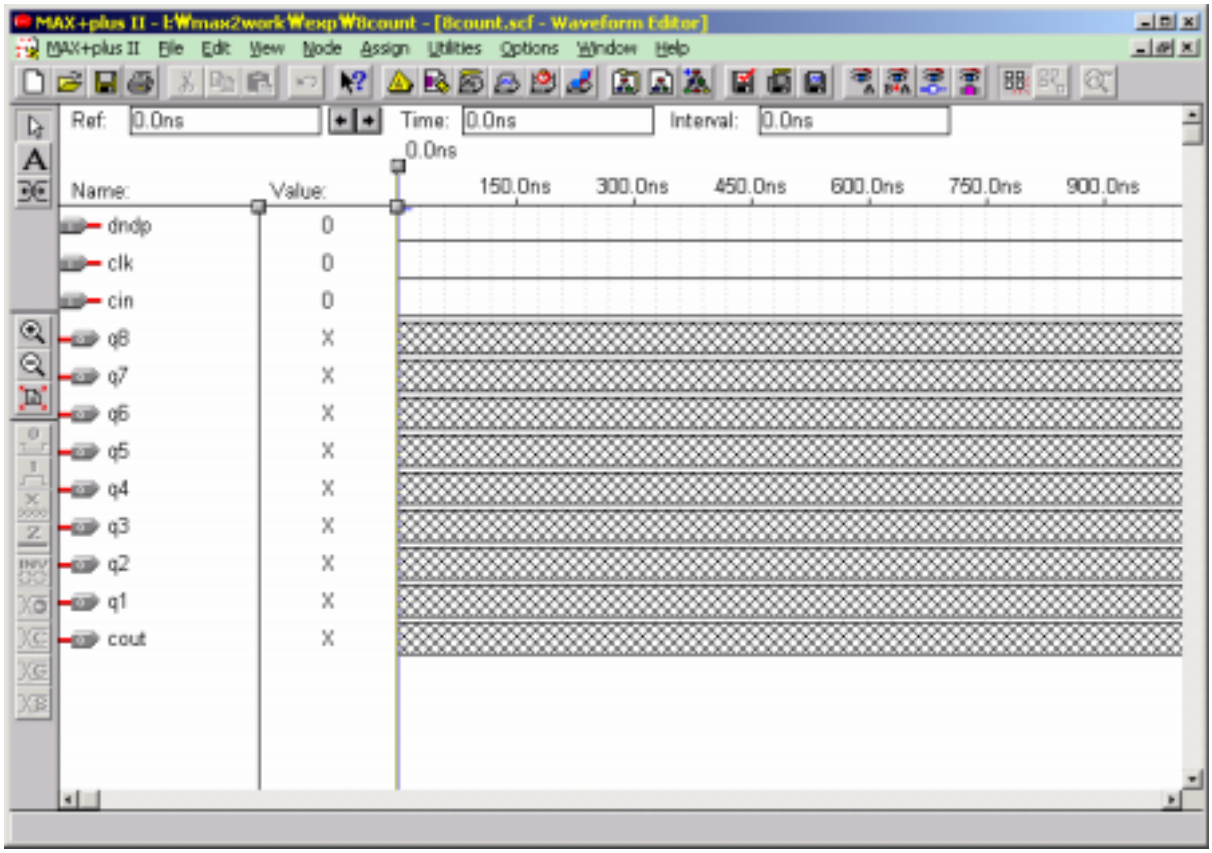


Figure 36 입/출력 노드 삽입

왼쪽에 있는 Icon Toolar 를 이용하여 입력 Pin 에 대한 시간에 따른 변화 파형을 입력하므로써 Stimulus Vector 를 생성하도록 하는데 각 Icon 의 기능을 간략히 설명하면 다음과 같다.

화면확대

화면축소

화면크기 맞춤

"0" 입력

"1" 입력

Unknown 입력

High Impedance 입력

현재입력된 값을 반전

Clock 입력

Increase/Decrease Count Value 입력

Multi Bit Signal 을 Bus Group 으로 만들기

Bus Group 을 Single Bit 로 풀기

Option 메뉴의 Grid Size...에서 grid size 를 30ns 로 할당하고, 위의 아이콘들을 이용하여 그림 37 과 같은 Waveform 을 생성한다.

기본적으로 Waveform Editor 에서 Stimulus 의 입력은 한칸의 Grid 씩 입력이 되도록 되어있다. (임의의 Block 을 선택하여도 Grid 단위로 선택되며, Clock 의 주기도 한 Grid 가 ½ Period 가 된다.) 이는 MAX+plus II 의 Option 메뉴중 "Snap to Grid"라는 항목이 선택되어있기 때문인데, 만약 사용자가 Grid 간격에 구애받지않고 Clock 등의 입력을 하고 싶다면 "Snap to Grid" 항목의 Check 표시를 꺼야한다.

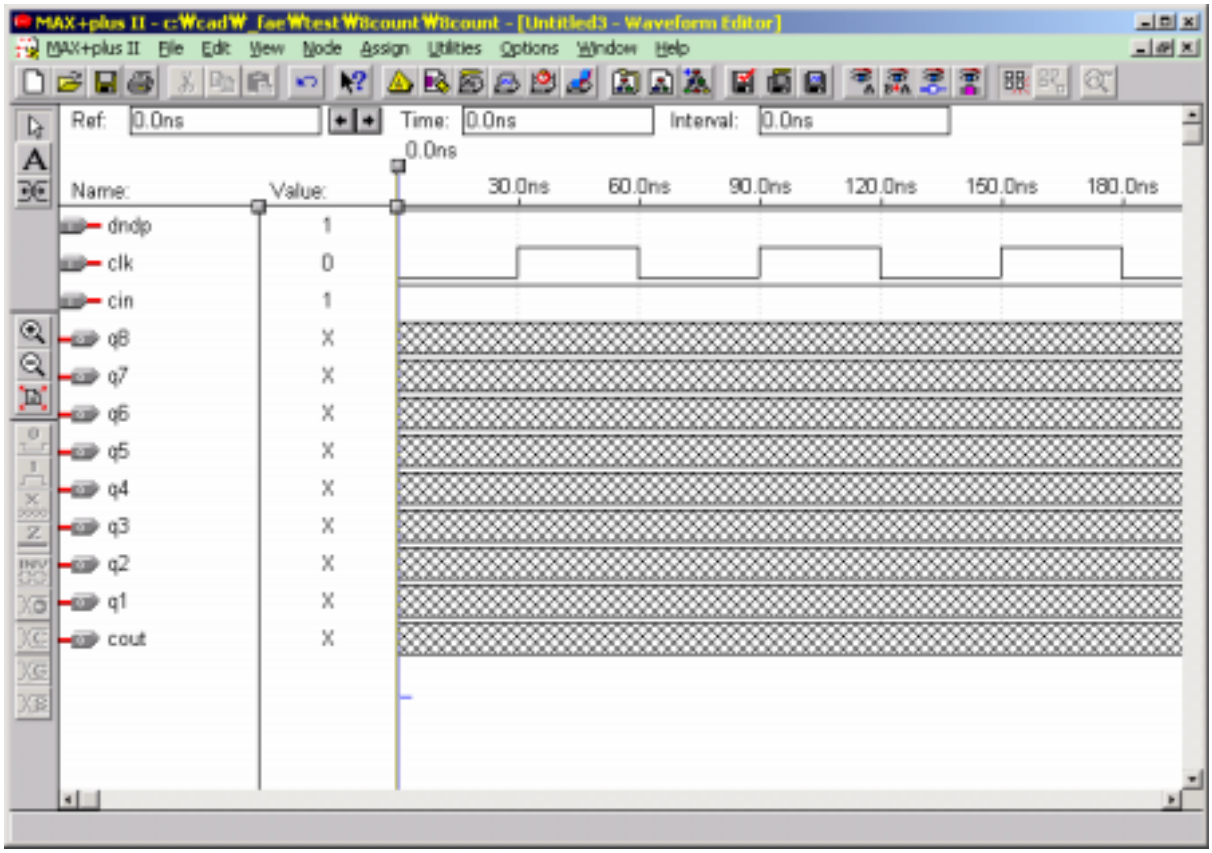


Figure 37 입력신호에 대한 Stimulus 입력

최종적으로 Waveform 의 Editing 이 완료가 되면 다시한번 해당파일을 저장한다.

여기까지가 Simulation 을 하기위한 준비단계로 입력 Waveform 의 설정과정이며, 실제로 Simulation 의 수행은 MAX+plus II S/W 의 Main Menu 중 MAX+plus II -> Simulator 를 선택하여 Simulation 을 실시한다.

그림 38 을 보면 현재 행할 Simulation 은 Timing Simulation 이고, 방금전 작성한 Waveform File 이 Simulation Input: 항목에 자동적으로 선택이 되어있다. 또한 Waveform 의 내용을 분석하여 Start Time:과 End Time:이 설정이 되어있다.

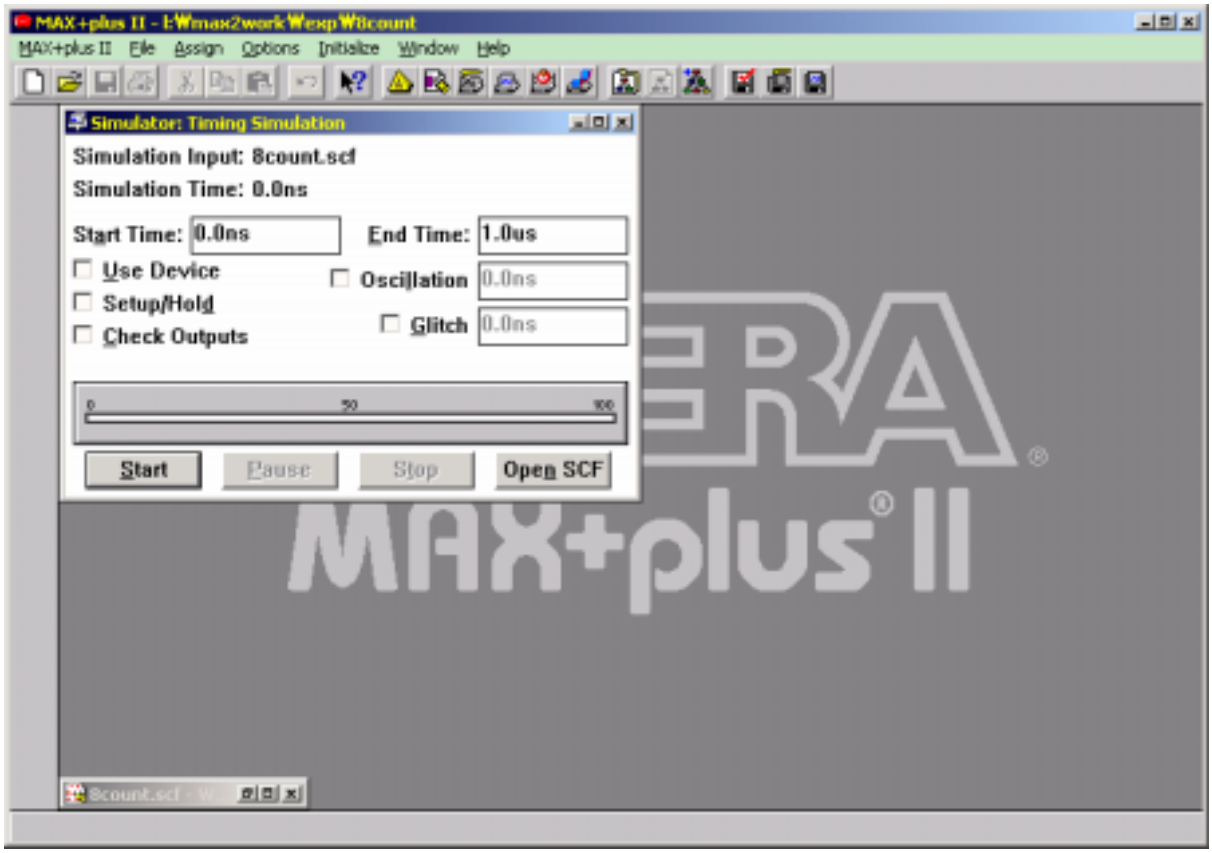


Figure 38 Simulator 의 실행

Start Button 을 누르면 실제 Simulator 가 작동, Simulation 을 수행한다. Simulation 이 끝나면 **Open SCF** button 을 눌러 Simulation 결과를 Waveform 으로 확인할 수 있다.

그림 39 는 Simulation 결과를 보여주는 그림이다. 우리는 Timing Simulation 을 수행하였으므로 출력신호의 결과들이 Target Device 의 Delay 정보와 연동되어 나타남을 알 수 있다.

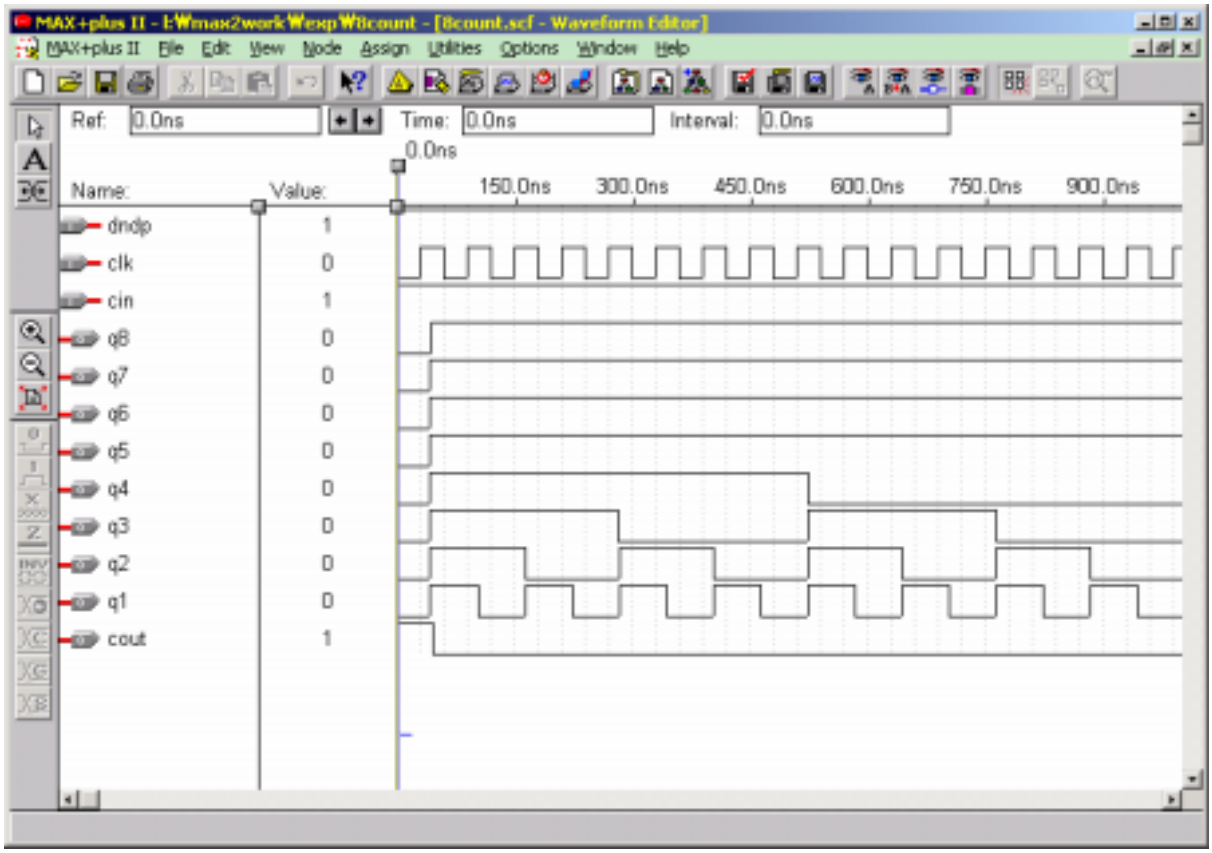


Figure 39 Simulation 결과

5

마치며...

이상으로 MAX+plus II 를 사용한 디지털 논리회로 설계를 마칩니다... 이 이후에 다루어져야 하는 부분은 Compile 된 결과를 직접 Hardware 상에 장착된 FPGA 나 CPLD 에 Configuration 혹은 Program 과정을 거쳐 동작을 확인하는 과정이 필요하나 별도의 Hardware 가 준비되지 않은 상황에서 단지 문장만으로 표현하는데는 다소 무리가 있다고 판단되어 생략되었습니다...

본 문서가 ALTERA PLD 와 S/W 를 처음 접하시는 모든 분들께 약간이나마 도움이 될수 있기를 바랍니다...

Revision History

- 2000-12-4 - Ver 1.0: Initial Release... (최초 작성: 박태응)
- 2001-10.4 – Ver 2.0: 오자, 탈자 제거... 불필요한 군살제거 및 부실한 내용 추가등 전반적인 손질... 그림 작업 몽땅 새로함... (수정: 양창우)