CPLD Software Tutorial

FOR

ALTERA MAX+PLUS II

Note: This material was developed by William Kleitz for inclusion in his textbook Digital Electronics: A Practical Approach 6th edition, (Prentice-Hall 2002).
http://vig.prenhall.com/catalog/academic/product/1,4096,0130896292,00.html
The software used for this CPLD development was downloaded from the Altera website (www.altera.com).

We will learn the basics of CPLD design and simulation by building a solution to the

Boolean equation **X = AB(C+D)**

Start the Altera MAX+PLUS II program. The main screen is shown in Figure E-1.

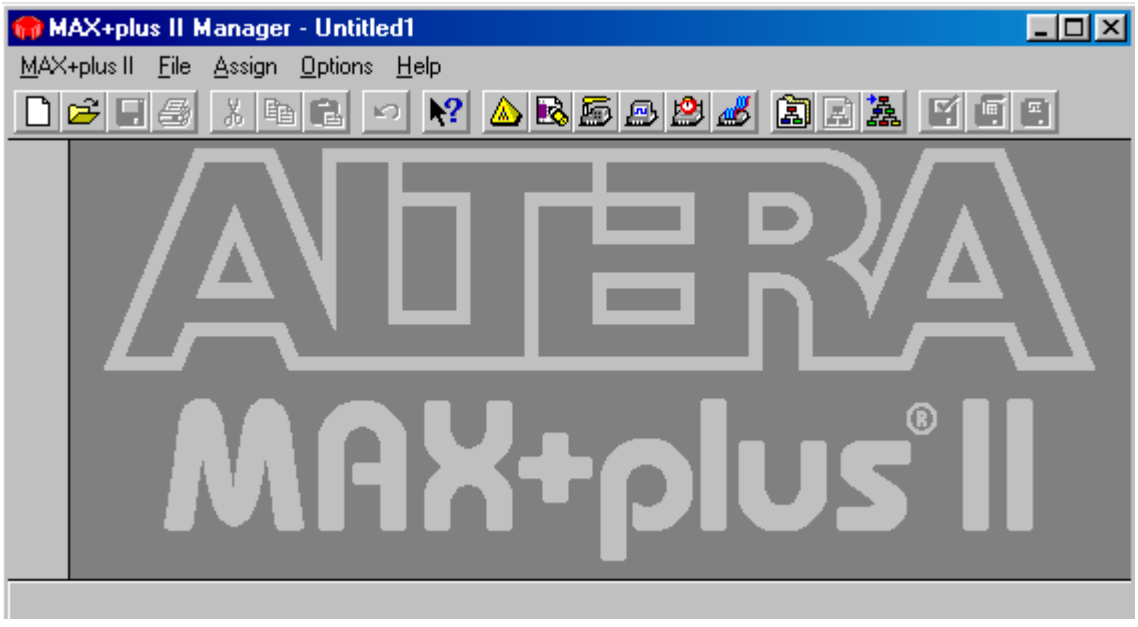

Figure E- 1

Create a Graphic Design File (.gdf*)**

1. To draw the logic circuit for our Boolean equation we will use the graphic editor to create a Graphic Design File:

Choose **File > New > Graphic Editor file > .gdf**

Then press **OK** (See Figure E-2)

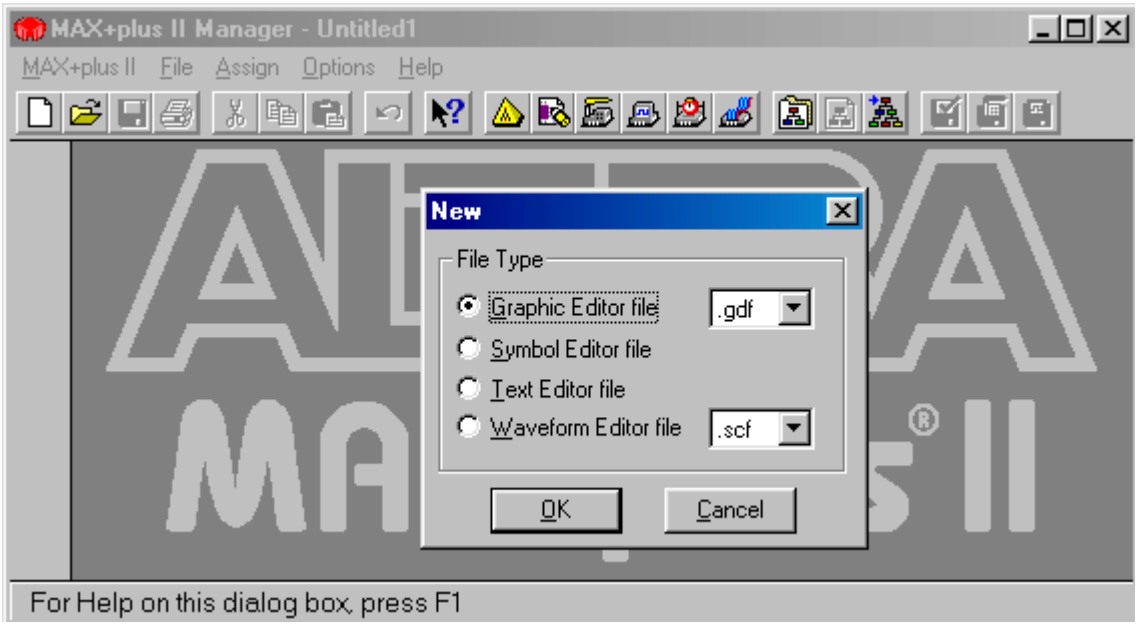

Figure E- 2

2. You then want to assign a name to your file:

Choose **File** > **Save As**

File Name **C:\maxplus2\kleitz*boolean1***

Then press **OK** (See Figure E-3)

When asked if OK to create new directory, press **YES**

[The extension will be *.gdf.* Ask your class instructor, but you will probably want to use your name as the subdirectory in place of kleitz.

The complete pathname **C:\maxplus2\kleitz*boolean1*** creates the subdirectory kleitz and only needs to be given the first time that you save a file. Subsequent times will only require the file name (like *boolean1*) as long as you are careful to have the correct subdirectory name highlighted in the Directories

column.)

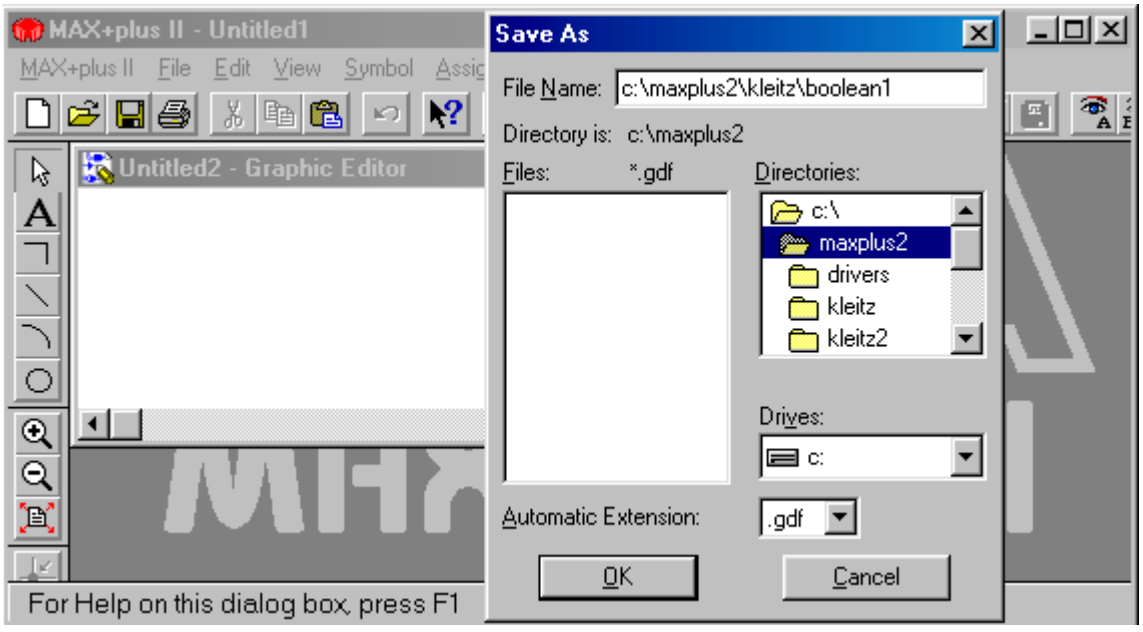

Figure E-3

3. Next we need to specify a Project name:

Choose **File** > **Project** > **Set Project To Current File** (See Figure E-4)

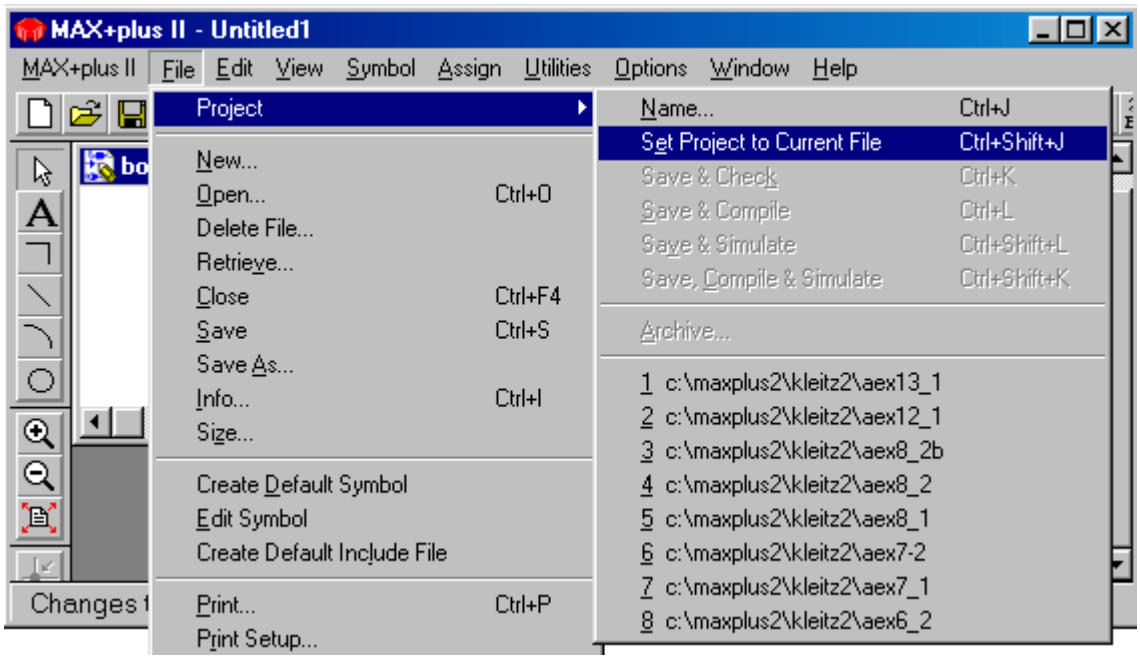

Figure E-4

4. Now you are ready to draw the logic circuit. To add a logic symbol to the graphic editor workspace :

Double click anywhere in the workspace. Enter the symbol name for the first logic gate (a 2-input AND gate) by typing **AND2** then press **OK**

5. Repeat for the second AND gate and for the 2-input OR gate (**OR2**)

6. Input and output pins must now be added to the circuit:

Double click somewhere near the input to the first gate and type the symbol name **INPUT** then press **OK** . Double click on the **PIN_NAME** and change it to **A.**

7. Repeat for the other inputs: **B, C, D** and the output **X**. (The results of steps 4-7 are shown in Figure E-5.)

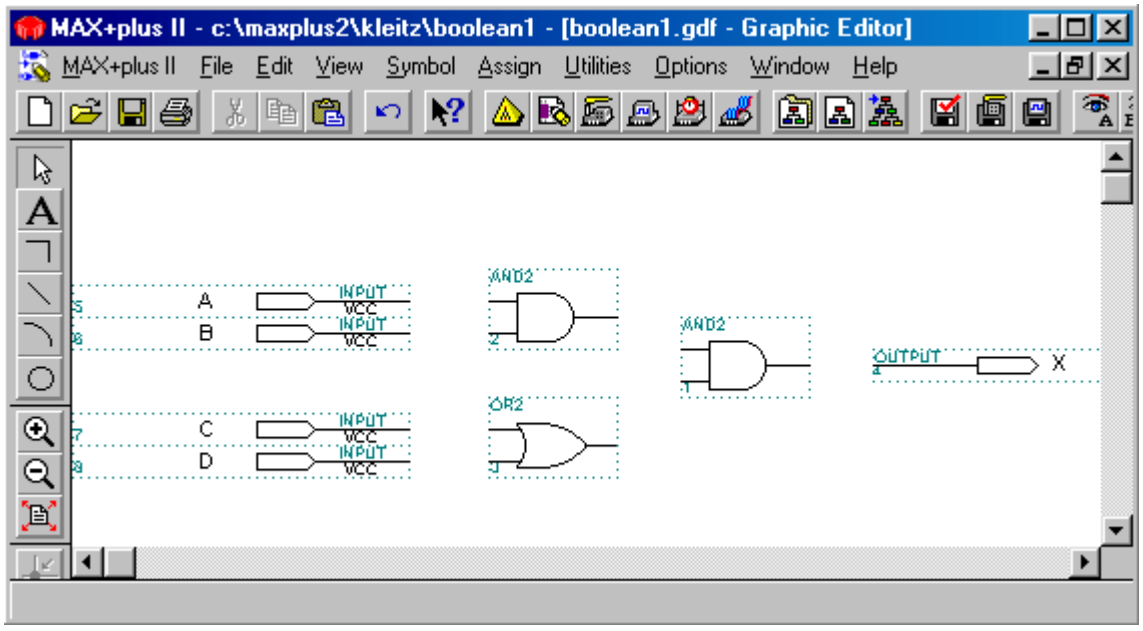

Figure E-5

8. Next you want to connect the symbols by drawing lines between them. First, use the left mouse button to drag the input/output pinstubs so they line them up with the appropriate logic gate. Then use the left button to draw a line from the end of each pinstub to the beginning of each corresponding logic gate input. If you make a mistake highlight the line and press the Del key. (See Figure E-6.)

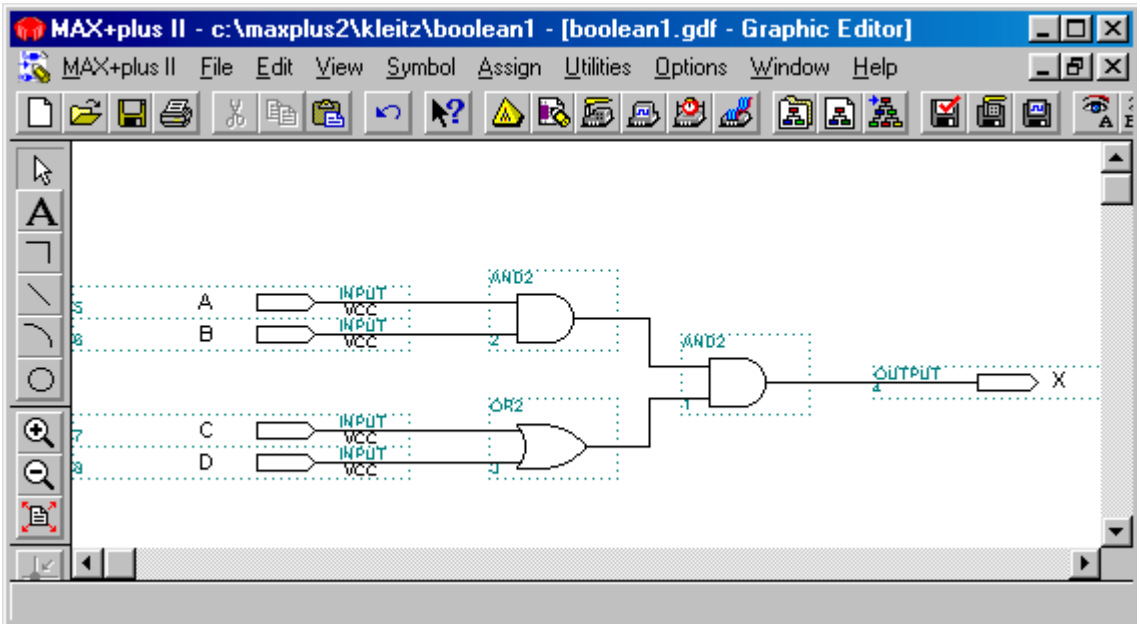

Figure E-6

9. We are done creating the graphic design file and can now assign an Altera device to be used to implement the design.

To assign a device to the project:

Choose **Assign** > **Device**

The CPLD that we will be using is in the Device Family **MAX7000S**

In the devices column highlight **EPM7128SLC84-7** then press **OK. (**Be sure that "Show Only Fastest Speed Grades" is not checked.) (See Figure E-7.)

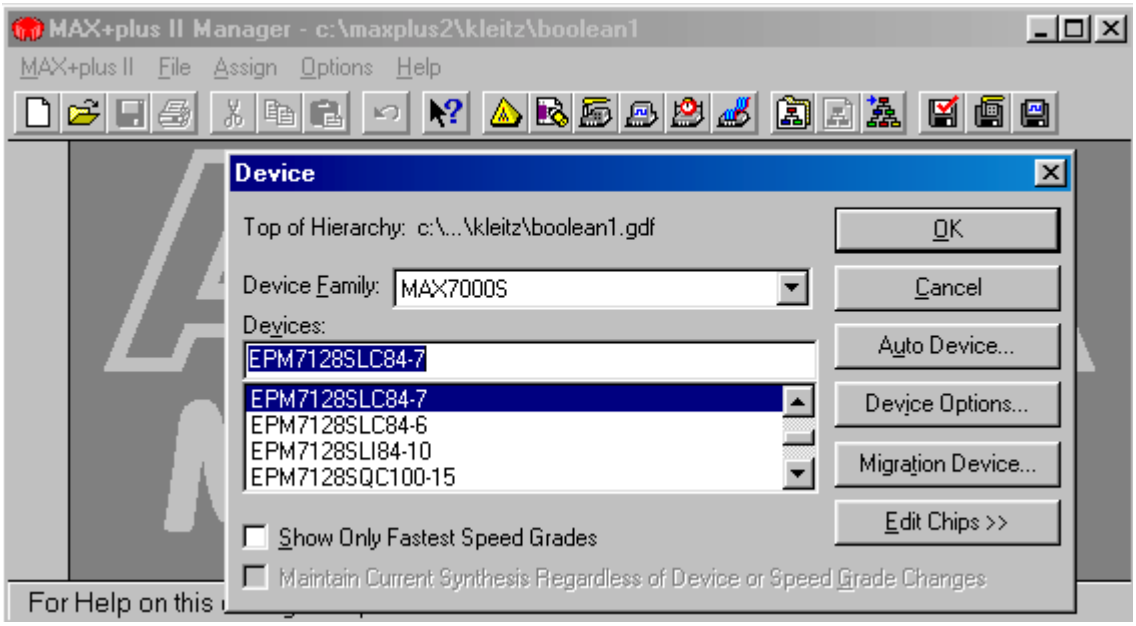

Figure E-7

10. Finally you want to save the project and compile it to check for errors.

Choose **File > Project > Save and Compile**

The compiler message should display 0 errors, Press **OK** (See Figure E-8.)

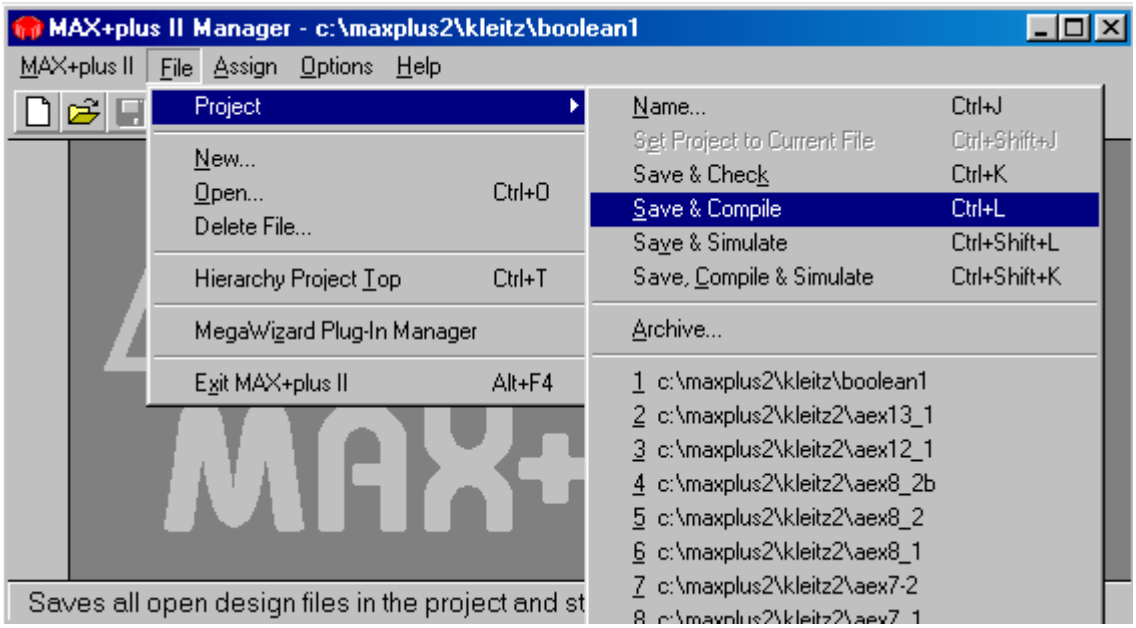

Figure E-8

Create a Simulator Channel File (**.scf*) for Waveform Simulation

Next, we will use the waveform editor to create a Simulator Channel File to test and simulate our logic design. To test our logic we need to simulate all 16 possible combinations of inputs at A, B, C and D.

1. To create a new Simulator Channel File:

Choose **File** > **New** > **Waveform Editor file** > **.scf**

Then press **OK** (See Figure E-9.)

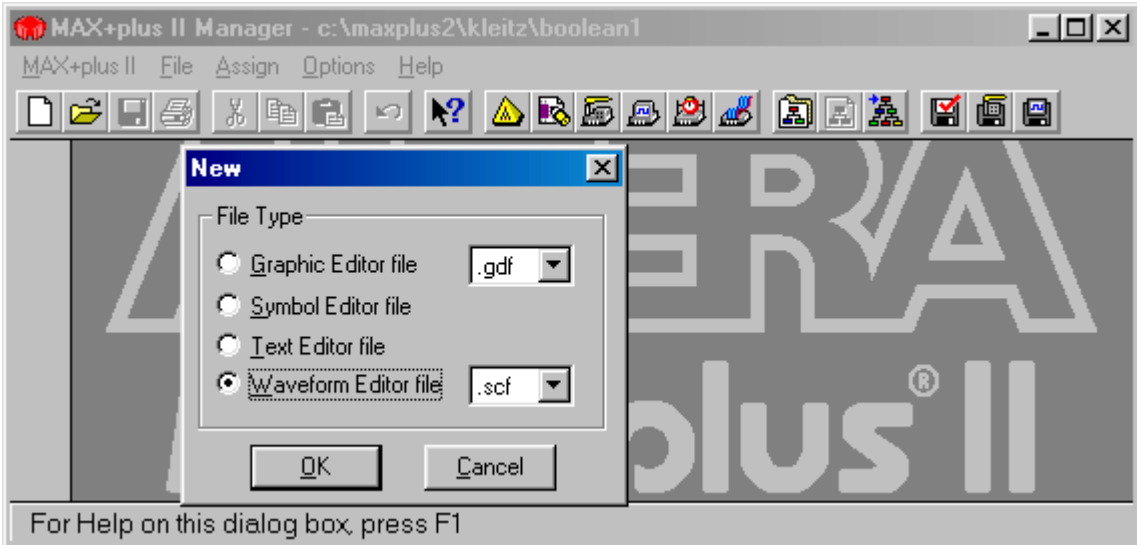

Figure E-9

2. You then want to assign the same name to this file:

Choose **File > Save as > *Boolean1***

Then press **OK**

(the file extension will be *.scf*. Again, be sure that the correct subdirectory name is highlighted first.) (See Figure E-10.)

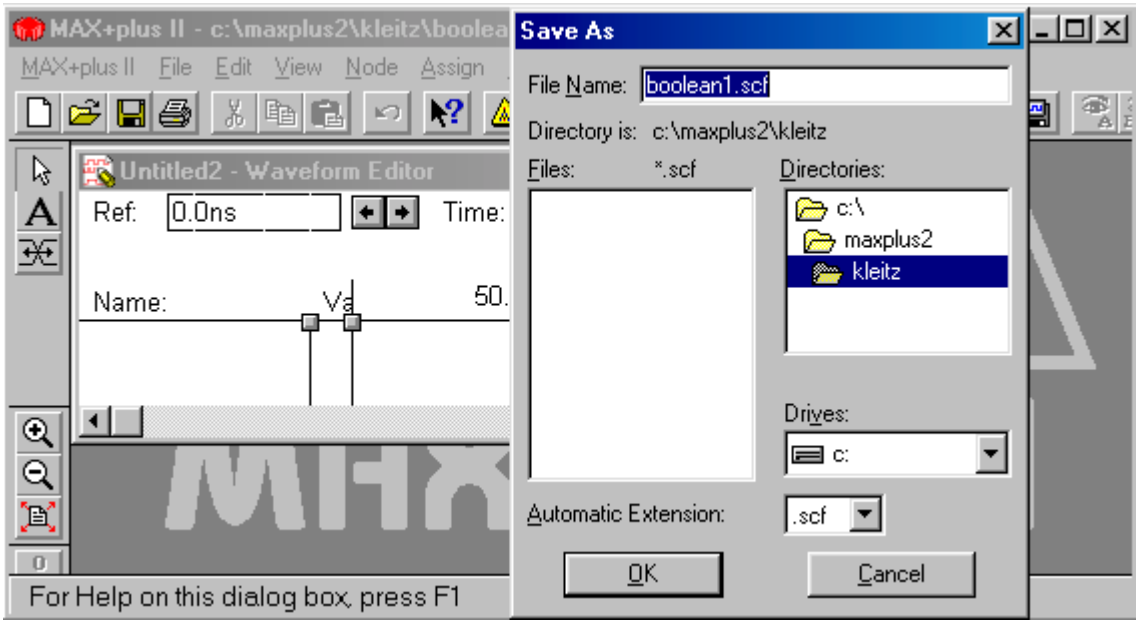

Figure E-10

3. You should now be looking at a blank Waveform Editor screen. The first thing we need to do inside the waveform editor is create node names for the 4 inputs and 1 output. To do this, position the cursor in the area just below the word *Name:* and then double click. This brings up the Insert Node menu.

Now to enter the Node Name for the first input:

Enter **A** then choose **Input Pin** then press **OK** (See Figure E-11.)

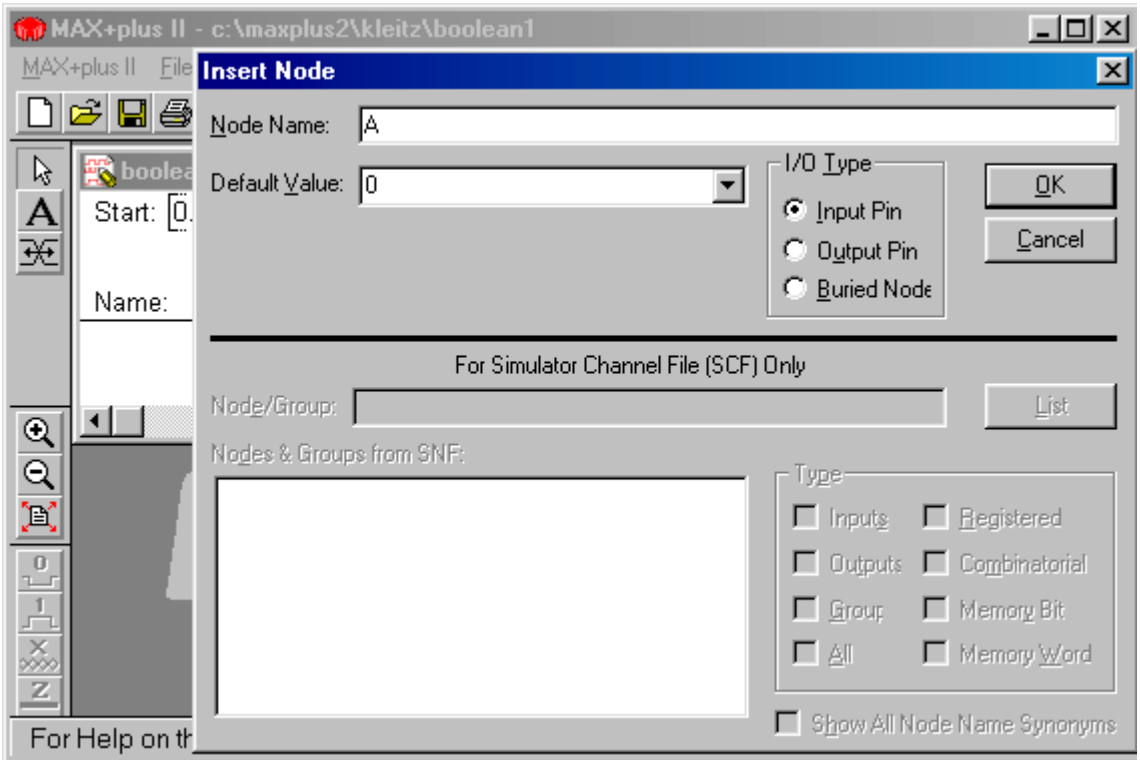

Figure E-11

4. Repeat step 3 for the other three inputs: B, C, and D

5. The fifth node name will be for the output pin named X:

Enter **X** > **Output Pin** Then press **OK** (See Figure E-12.)

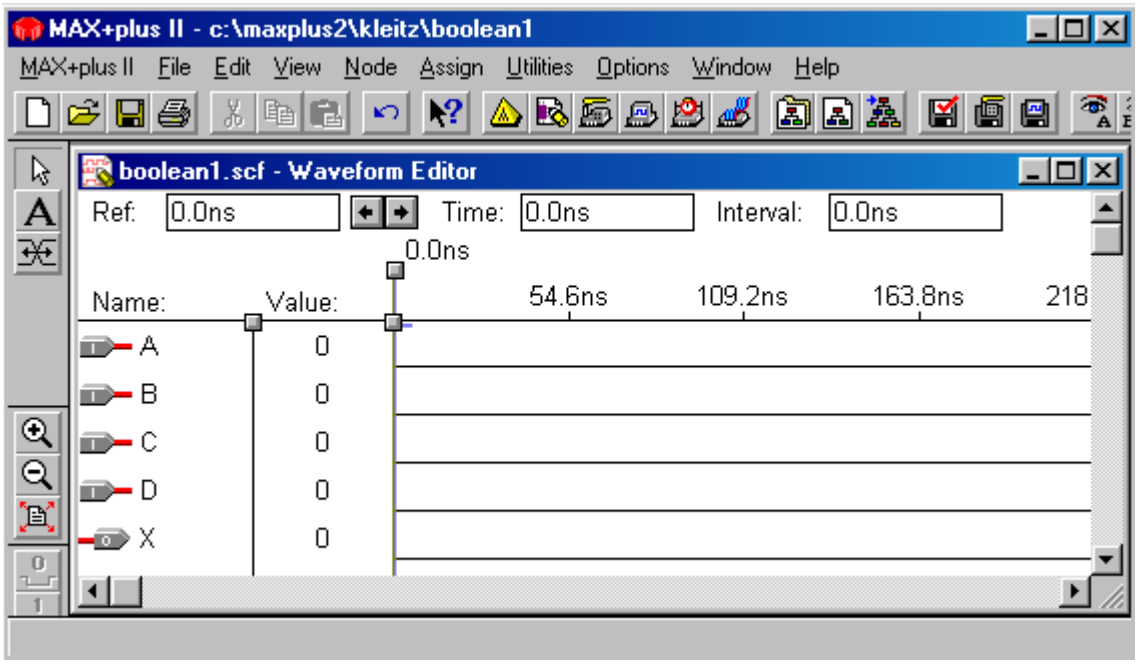

Figure E-12

6. Now we need to develop four input waveforms that will cover all 16 possible combinations of inputs. To set up these waveforms perform the following steps (The parameters listed below are recommended to make your display match the figures in this tutorial.):

Choose **File** > **Endtime** > **16us**

Choose **View** > **Fit In Window**

Choose **Options** check **Show Grid**

Choose **Options** > **Gridsize** > **1us**

7. Next, we want to draw a sequence of waveforms that count from 0000 up to 1111.

To draw the first waveform for the **A** input make the following entries:

Left click on node name **A** then

Choose **Edit** > **Overwrite** > **Clock**

In the Overwrite Clock screen:

Choose **Multiply by 1** (This will draw clock pulses with a period of **1 X** 1uS.)

Then press **OK** (See Figure E-13.)

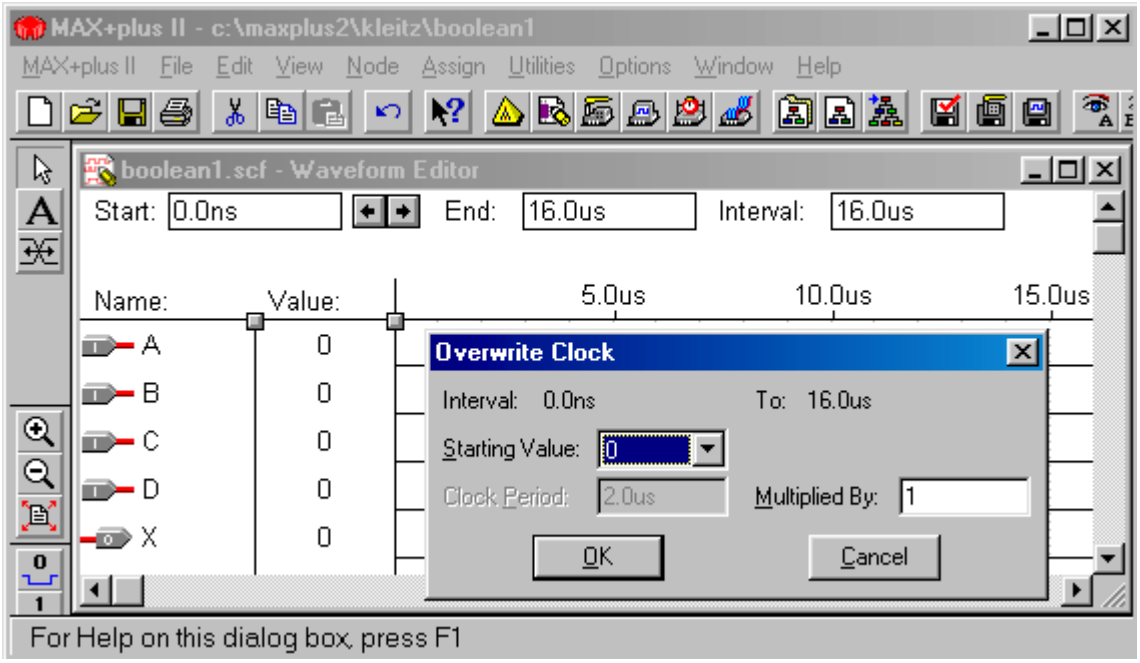

Figure E-13

8. Then draw the remaining waveforms:

Left click on node **B** then

Choose **Edit > Overwrite > Clock**

In the Overwrite Clock screen:

Enter **2** in the **multiply by** field (This will draw clock pulses with a period of **2 X** 1uS.)

Left click on node **C** then

Choose **Edit > Overwrite > Clock**

In the Overwrite Clock screen:

Enter **4** in the **multiply by** field (This will draw clock pulses with a period of **4 X** 1uS.)

Left click on node **D** then

Choose **Edit > Overwrite > Clock**

In the Overwrite Clock screen:

Enter **8** in the **multiply by** field (This will draw clock pulses with a period of **8 X** 100nS.)

The final Waveform Editor file will show waveforms counting from 0000 up to 1111

To save your file press **File** > **Save** (See Figure E-14.)

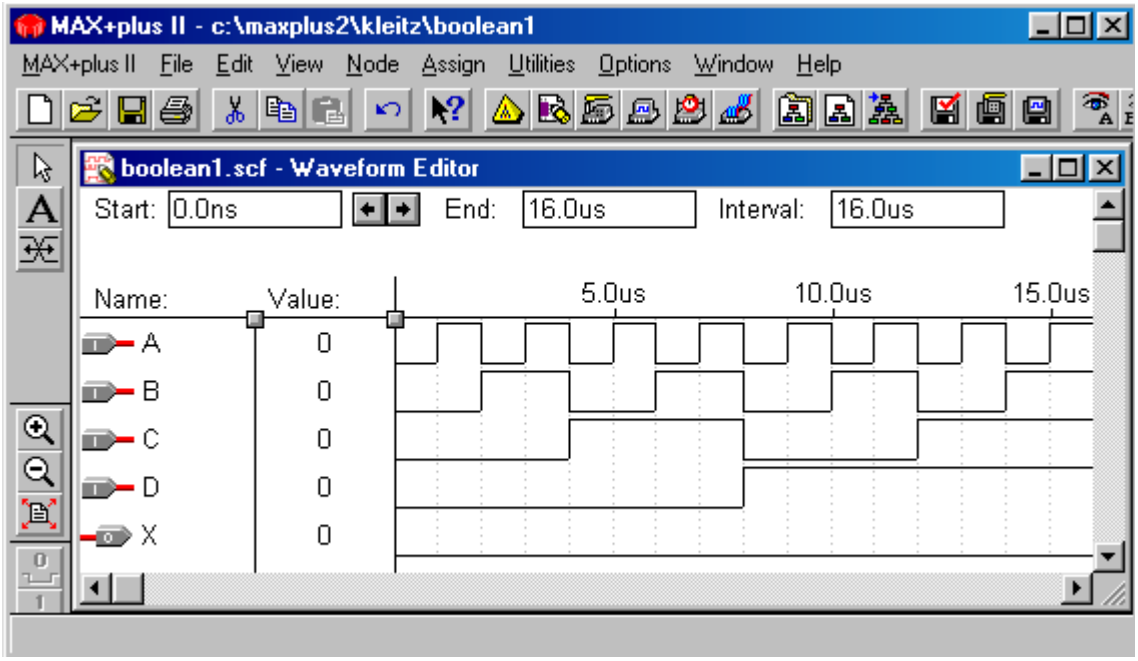

Figure E-14

Waveform Simulation

The waveform simulator will now use the Simulator Channel File (*.*scf*) to simulate the inputs to the logic circuit in the Graphic Design File (*.*gdf*) and produce the output at **X**.

Choose **File** > **Project** > **Save, Compile and Simulate**

It should return *0 errors.* Press **OK**

In the Timing Simulation window press **Start**

It should return *0 errors.* Press **OK**

To see the results of the timing analysis press **Open SCF**

The output waveform at **X** should satisfy the Boolean equation for **X=AB(C+D)**

Programming Altera PLDs Using The UP-1 Education Board

The UP-1 education board has two PLDs and all of the support circuitry required to download our logic designs so that they can run on the actual PLDs they were designed for. You will need to refer to the UP-1 Users Guide for all of the operating detail but its' basic operations will be described in the paragraphs that follow.

The EPM7128S Device

This CMOS EEPROM-based PLD is a member of the Altera high-density MAX 7000 family. It is an 84-pin PLCC package containing 128 macrocells. Each macrocell has a programmable-AND/fixed-Or array for implementing combinational logic. It also has a configurable register for implementing sequential circuits like counters and shift registers.

We will use this part for most of our designs. We will download our design to the IC and then test it in a stand-alone configuration. Because of its non-volatile nature our design will remain on the IC even when power is removed and reapplied. Also since it is an Electrically Erasable PROM (EEPROM) we can reprogram it over and over to test our other designs.

Connecting Inputs and Outputs To The PLD

The EPM7128S PLCC is socket-mounted on the UP-1 board. Female headers are on the board providing wiring access to each of the 84 IC pins. We will use solid 22 to 24 gauge hook-up wire to connect to the 16 switches and 16 LEDs provided on the board. The switches are pulled-up through 10-KΩ resisters and thus provide a logic 1 when the switch is open (OFF) and a logic 0 when ON. The LEDs are pulled up with a 330-Ω resistor and thus are illuminated with a logic 0. (Refer to the UP-1 Users Manual for the header pin numbers and the wiring description of the switches and LEDs.)

The UP-1 board has jumpers that need to be set to configure its operation. DC power must be applied to the board via a user-supplied power supply. A connection to the PC's parallel port is made with the cable included with the UP-1 board. The UP-1 Manual provides all of the steps required to configure the board and make the hardware connections to the PC.

Instructions For Programming The PLD

In this section we will download our design to the EPM7128S CPLD. After the successful download we will test the operation of the IC by connecting the on-board switches and LED to the IC and step through several combinations of input levels to be sure the output responds correctly.

Setting the On-Board Jumpers for EPM7128S Programming

Refer to the UP-1 Users Guide to determine the jumper settings required to program the correct IC on the board (The EPM7128S in this case). Make the jumper changes specified.

Connecting The Download Cable From The PC To The UP-1 Board

Refer to the UP-1 Users manual to determine the correct connection points for the supplied download cable. Make the proper connections now.

Apply Power To The UP-1 Board

Refer to the UP-1 Users Guide to determine the voltage level and connection point for powering up the UP-1 board. Make the proper connections now.

Download Your Circuit Design

If not already open, then open your **.gdf* file and set the project to the current file by performing the following steps:

Choose **File** > **Open** > ***Boolean1.gdf*** > **OK** then

Choose **File** > **Project** > **Set Project To Current File**

Now to Download the design:

Choose **MAX+plus II** > **Programmer**

If this is the first time programming a device a “Hardware Setup” window will appear.

Choose the correct **Hardware Type** and **Port** for your UP-1 then press **OK**

Back in the Programmer window press **Program**

After a successful program download (Programming complete) press **OK**

Testing The Downloaded Design On The UP-1 Board

The logic circuit is now programmed into the EPM7128S CPLD. The female headers next to the IC provide a means to connect to switches or push buttons to exercise the inputs. (Remember, as the user guide states, putting the switch in the ON position places a LOW at that input.) The output of the design will be wired to one of the active-LOW LED indicators on the board to monitor the output result.

Determine The CPLD Input And Output Pin Numbers [1]

We need to determine which pins on the CPLD were selected by the software for the inputs and output by studying the project's **.pin* file. To view the pin numbers:

Choose **File** > **Open** then enable the circle in front of **All Files**

Highlight the file ***Boolean1.pin*** then press **OK**

This will display a text file that lists all of the pin assignments for your programmed CPLD. Make note of the input pin numbers and the output pin number.

[1] Instead of allowing the software to select the I/O pins, you can assign specific pin numbers to your project. However, several pin numbers are not allowed as specified in the UP-1 manual, so be careful. To assign a pin number, right-click on an input or output and choose **assign** > **pin/location/chip.**

Re-compile and save these new assignments.

Connect Wires To The Input Stimulus Switches And Output LED

The EPM7128S CPLD is an 84 pin IC. You will need to refer to the pin layout table provided for it in the UP-1 Users Guide to find the appropriate header to connect your inputs and output to. The layout for the headers is also shown in Figure E-15.

Make the necessary connections from the header to the switches and LED using 22 or 24 gauge solid wire.

Exercise the CPLD by testing all possible combinations of inputs as you watch the output on the LED. Remember that an ON switch = '0' and that the LED is active-LOW. [In the future, you may want to add an inverter (Called a "NOT" gate) to the output so that the LED acts like an active-HIGH.]

```
      11 9  7  5  3  1 83 81 79 77 75
      x  10 8  6  4  2 84 82 80 78 76
12 13                              x  74
14 15                              73 72
16 17                              71 70
18 19          ALTERA              69 68
20 21          EPM7128S            67 66
22 23          Headers             65 64
24 25                              63 62
26 27                              61 60
28 29                              59 58
30 31                              57 56
32  x                              55 54
      34 36 38 40 42 44 46 48 50 52  x
      33 35 37 39 41 43 45 47 49 51 53
```

Figure E-15

Optional VHDL Design Entry

In this section, instead of designing the *boolean1* logic circuit using the graphic editor, we will use the VHDL Text Editor.

1. To define the boolean equation using VHDL :

Choose **File** > **New** > **Text Editor file**

Then press **OK** (See Figure E-16)

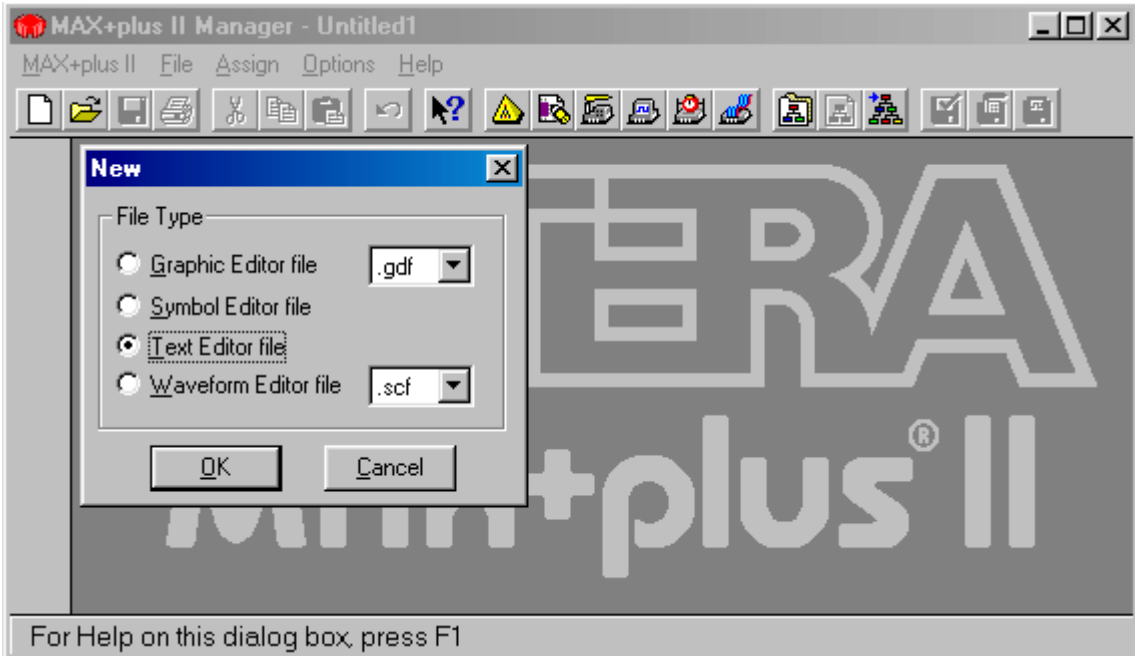

Figure E-16

2. Type in the VHDL program for $X = AB(C + D)$ shown in figure E-17.

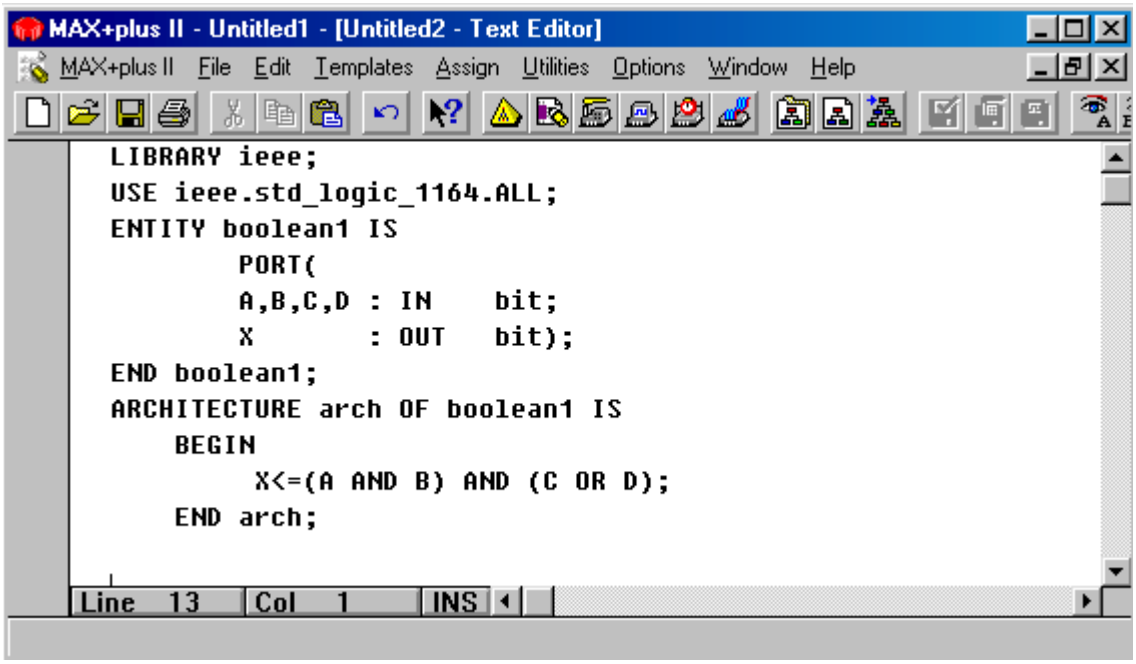

Figure E-17

3. Save the VHDL file:

Choose **File** > **Save as** > ***boolean1*** > **automatic extension** > **.vhd**

(Be sure that the correct sub-directory is highlighted as shown in Figure E-18.)

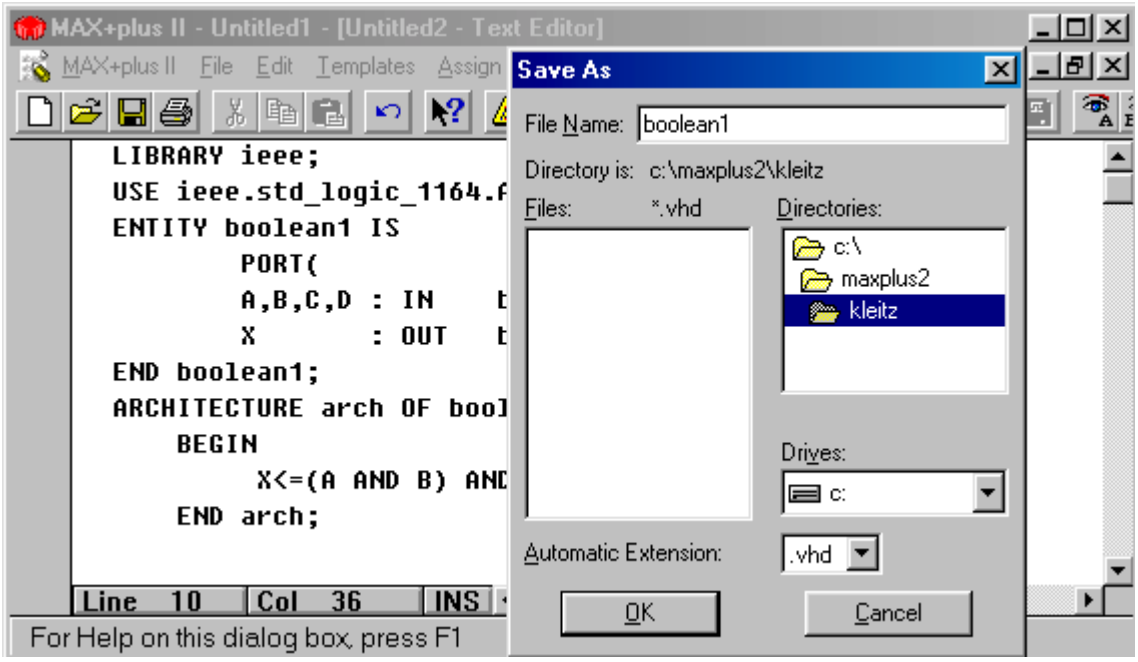

Figure E-18

4. To compile and test for errors in the VHDL program:

Choose **file** > **project** > **set project to current file**

Then choose **file** > **project** > **save and compile**

The compiler message should show *0 errors*. Now you can follow the steps previously outlined to build a Simulator Channel file *(*.scf)* to test your design.