*the* ***Technology Interface / Winter98***

**Getting Started With Altera VHDL**

**a tutorial exercise**

by

Jeffrey S. Beasley

[jbeasley@nmsu.edu](mailto:jbeasley@nmsu.edu)

Department of Engineering Technology

New Mexico State University

and

William B. Hudson

[wbhudson@eece.ksu.edu](mailto:wbhudson@eece.ksu.edu)

Department of Electrical and Computer Engineering

Kansas State University

**Abstract**

*This VHDL tutorial has been prepared to acquaint the student with VHDL programming using the student edition of the MAX+PLUS II Version 7.21 Student Edition PROGRAMMABLE LOGIC DEVELOPMENT SOFTWARE. The tutorial provides a step by step procedure for implementing a simple VHDL program in the Altera system.*

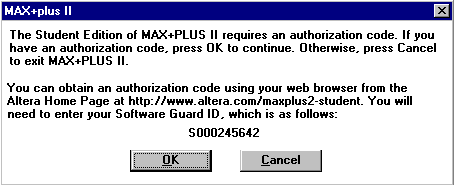
**Introduction**

Altera Corporation now provides a student edition of their powerful programmable logic development software through their University Program. Any university can apply to participate in the program The URL for the Altera University program is [http://www.altera.com/html/univ/info.html .](http://www.altera.com/html/univ/info.html)

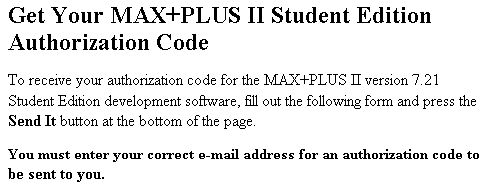
This suite of tools provides an option for the development of complex digital systems using their VHDL (**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage) compiler. The student edition is very powerful and contains most of the features of the full version except the student version only compiles to a limited set of Programmable Logic Devices (PLDs), these being the

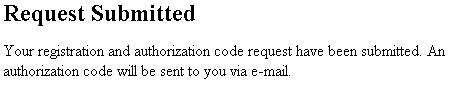
* 84-pin MAX 7000 EPM7128SLC84-7 (2,500 usable gates)
* 84-pin FLEX 10K EPF10K20RC240-4 (15,000 to 63,000 usable gates

The installation procedure for using the software is simple. It requires 33 MB of hard disk space and 8 MB or RAM. The student edition does require that the user registers on-line with Altera. Following software installation, the user is given a software guard ID number and instructed to register this number with Altera. The number is registered through the Altera WWW site at <http://www.altera.com/maxplus2-student>. Once this number is registered with Altera, the user is emailed an authorization code number. The user must enter the authorization code before using the software. Fig. 1 shows the Altera prompt for the registration number followed by the instructions for obtaining the authorization code, and acknowledgement of the request for registration and authorization code.

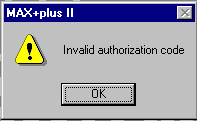


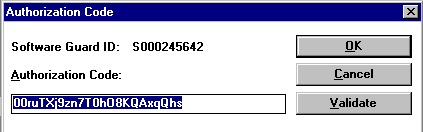
**Fig. 1 The Altera prompt for the registration number.**





It may take up to ½ hour to get your authorization code from Altera. Sometimes you will get your code number back in a few minutes. Once you have the authorization code start your Max+plusII software. Your software should prompt you that you have an invalid authorization code as shown in Fig. 2. Click on OK and you will get another screen as shown in Fig. 3 prompting you to obtain a proper authorization code from Altera. Click on OK and enter the authorization code. You can also change your authorization code by selecting **Options / Authorization Code.** You will get the screen shown below.

  
**Fig. 2 Prompt indicating an invalid authorization code.**

  
**Fig. 3 Prompt for the proper authorization code**

This tutorial assumes that the student has been briefly introduced to VHDL syntax. The tutorial contains a simple VHDL description of a TTL 7408 2-input AND gate. The VHDL program will run, as is, with the Altera package.

**Altera VHDL Programming Procedure**

**Step 1**. Select **MAX+plus II / Text Editor** to create the VHDL file.

**Step 2**. Type the VHDL listing as shown below. This is the VHDL listing for a 2-input AND gate comparable to one gate in a TTL7408 package. For this exercise, the entity has been called ttltest.

**entity ttltest is port -- The entity declaration is used to identity each Input /**

**-- Output Signal. These are external pins only. Internal**

**-- wiring is handled differently.**

**(**

**a,b: in bit; -- Defines inputs "a" and "b" as a bit type. This means 1 and 0 values only.**

**y: out bit -- This defines the output (y) as a bit type.**

**);**

**end ttltest;**

**architecture behavioral of ttltest is -- set of statements describing the**

**-- function of the device**

**begin**

**process (a,b) -- this defines the process to be implemented**

**-- (a,b) indicates that the process is sensitive to changes in**

**-- either input a or b**

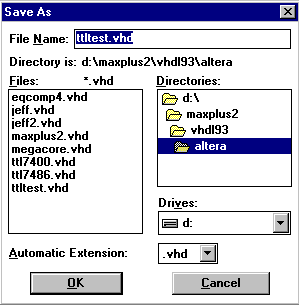
**begin**

**y <= a and b; -- y is assigned the value of the logical operation a and b**

**end process;**

**end behavioral;**

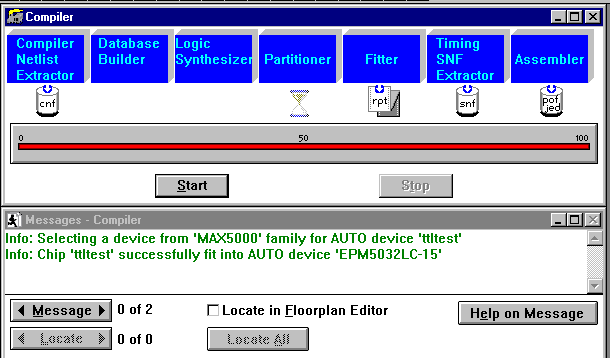
**Step 3.** Once you have completed typing the file select **File /Save As** and select the **.VHD** extension as shown in Fig. 4.



**Fig. 4 The file Save As window and the .VHD extension**

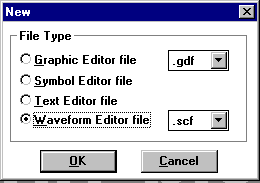
**Step 4.** At this point you need to make this file the current project. You can do this by clicking on **File / Project / Set Project to Current File**.

**Step 5.** Compile your file by selecting **File / Project / Save and Compile** or you can press **Ctrl L**. You will get a screen that looks like the Fig. 5. Note: Make sure the entity name and the file name are the same for now. Altera expects matching names.



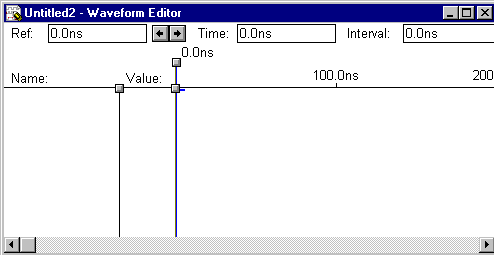
**Fig. 5 The Compiler Window**

**Step 6.** In the menu bar, click on **File / New.** You will get a screen which looks similar to Fig. 6. Select the **Waveform Editor file** option. Press **OK** when done.



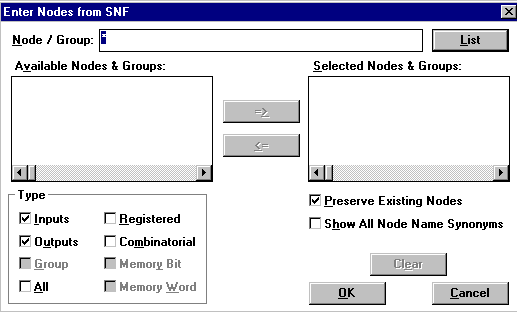
**Fig. 6 The New file window.**

You should get a new screen which looks like Fig. 7.



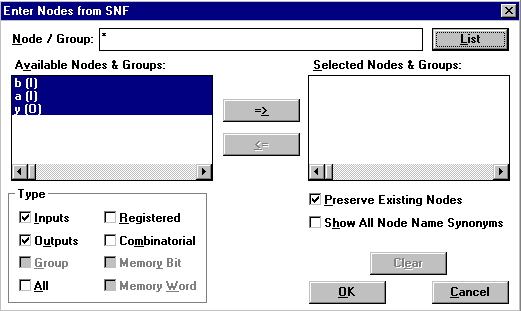
**Fig. 7 A new Waveform Editor window.**

**Step 7.** Click on **Node / Enter Nodes from SNF**. You will get a Enter Nodes from SNF screen which looks like Fig. 8.



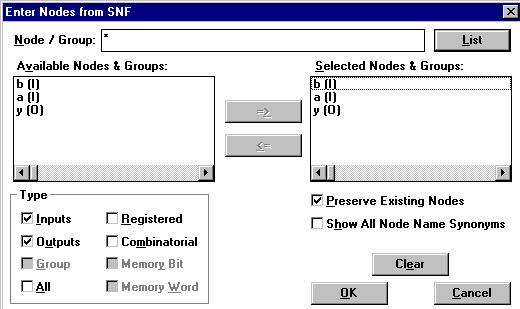
**Fig. 8 The Enter Nodes from SNF window.**

Click on the List button. Your screen should look like the Fig. 9.



**Fig. 9 The listing of the available nodes.**

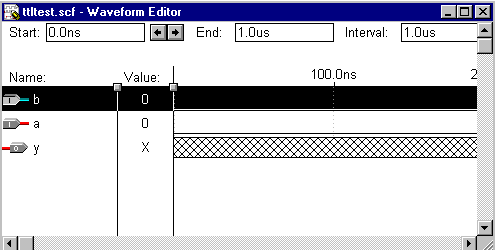
Click on the => button to move all of the selected values to the Selected Nodes and Groups window as shown below. Press **OK** when done. Your screen image will now look like Fig. 10. Notice that a list of Selected Nodes is now listed.



**Fig. 10 The list of Selected Nodes**

**Step 8.** Click on **File / Save** to make the Waveform Editor visible.

**Step 9.** In the next step, you will want to modify the input signals to some desired clocking sequence. The initial waveforms are shown in Fig. 11.



**Fig. 11 The waveforms for use in the simulation.**

You can modify the inputs as needed to fit your needs. Fig. 12 shows some options that are avaiable.

* + Click on **Options** and remove the **Snap to Grid**. This allows you to vary clock duty cycles, etc.
  + Select **File / End Time** to set the simulation end time.

These are the buttons available for altering the clocking.

**http://engr.nmsu.edu/%7Eetti/winter98/electronics/beasley/image30.gif**

**Set inputs to a "0"**

**Sets input to a "1"**

**Overwrites a selected waveform with an "X" value**

**For specifying a high impedance value**

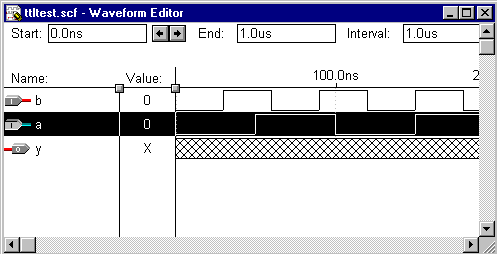
**Inverts waveforms**

**Overwrites a node with a clock waveform**

**For specifying a sepecific count sequence**

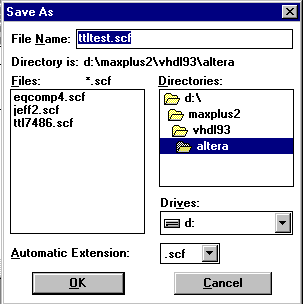
**Fig. 12 The function of some of the buttons available for altering the clocking**

Experiment with these options. Your waveform Editor should look like similar to the Fig. 13 when you are done.



**Fig. 13 The finished waveform editor file.**

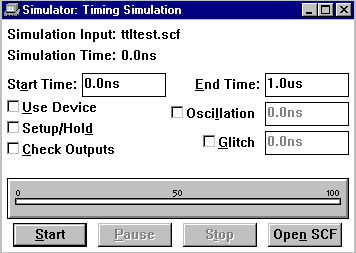
Click on **File / Save As** to save your Waveform Editor files as shown in Fig. 14.



**Fig. 14 Window for saving the waveform editor (.scf) file.**

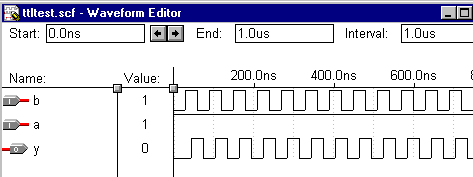
This creates the file name ttltest.scf. Click on **OK** when done.

**Step 10** To run the simulation, click on **Max + plus II / Simulator**. Your screen will look like Fig. 15.

.

**Fig. 15 The simulation startup window.**

Change the desired starting and ending times as needed. Click on **Start** when you are ready to run your simulation. You will get a prompt on your simulation window to alert you of any problems. Fig. 16 shows the results of thof the simulation run.



**Fig. 16 Result of the ttltest entity simulation.**

**Conclusion**

This paper has presented a short tutorial for using the Altra VHDL option. The information presented provides a brief introduction. Additional VHDL Sample Routines are provided in this issue of the Technology Interface for those wishing to gain more experience with VHDL.

**Suggested References**

Skahill, K., ***VHDL for Programmable Logic*** Addison Wesley, 1996

Bhasker, J., ***A VHDL Primer***, revised edition Prentice Hall, 1995

Ashenden, P., ***The Designer's Guide to VHDL***, Morgan Kaufmann, 1996

***Altra MAX+PLUS II VHDL***, Altera Corporation, San Jose, CA, 1996

***VHDL Made Easy***, David Pellerin and Douglas Taylor, Prentice Hall PTR, 1997

***VHDL Starter's Guide***, Sudhakar Yalamanchili, Prentice Hall, 1998