

A Beginner's Guide to MAX+plus II

Author: Jacinda Clemenzi - Last Updated: December 20, 2001

Last Revised: Frank Honoré - January 23, 2004

The purpose of this guide is to provide a tutorial that will help you to become familiar with how to use the software provided by Altera to edit Verilog, simulate designs, and program the FLEX 10K FPGA devices. It will be assumed that you have a basic understanding of Verilog.

Skip to the end of this document for a [QUICK REFERENCE](#)

Altera provides a software package for programming their devices. This package is called MAX+plus II which is available on the PCs in the Digital Lab. Launch MAX+plus II by clicking on the Desktop icon.

In addition MAX+plus II can be run on certain Sun Ultra machines running Solaris 8.

MAX+plus II is a fairly complete software package. It provides an editor, compiler, programmer, waveform generator, and simulator. Each of these functions can be chosen from the MAX+plus II menu option. This guide gives a walk-through of the different steps necessary to program a device. At the end is a step-by-step tutorial with an example project.

[Starting a Project](#)

[Working with Verilog Files](#)

[Configuring the Device](#)

[Compiling](#)

[Viewing the Report](#)

[Assigning Pinout](#)

[WARNING - BURNOUT](#)

[Simulation](#)

[Programming](#)

[Tutorial: A Simple Example](#)

[Appendix: Differences between MAX+plus II and Warp](#)

Starting a Project

A project consists of all the files necessary to program one device. MAX+plus II works with one project at a time. When you first open MAX+plus II you will have to create a project. The name of the project must be the name of the top level Verilog file. The

recommended method for starting a new project is to open an existing top level Verilog file or create a new one through the File menu option:

File > New...

or

File > Open...

Then set the name of the project to the current file by doing the following:

File > Project > Set Project Name to Current File

It is easy to change the name of the project and top level file at any time in the future.

All of the components you call in your top level Verilog file need only to be in the same directory for the compiler to find them.

When you open MAX+plus II next time, it will automatically load the last project.

Working With Verilog Files

MAX+plus II provides an editor that uses color coding much like Emacs. When you open a Verilog file in MAX+plus II it is the default editor. You can access it directly through the MAX+plus II menu option. You may use any editor to modify files. XEmacs is provided on the Lab PCs.

Configuring the Device

Once you have defined a project, you will need to configure the device.

Assign > Device...

Now select the FLEX 10K device from the pull-down menu. This will give you a number of selections in the Devices scroll menu. For this tutorial choose EPF10K10LC84-3. (If you later use the right hand chip, choose EPF10K70RC240-2.) From this same window click the **Device Options** button. From the pull-down menu next to the keyword *Configuration Device* under Device Options select EPC2LC20.

Compiling

Compile the code by selecting the compiler from the MAX+plus II menu option. Or click on the "chip with smokestack" icon. A compiler window is brought up to the front.

To compile the file click Start. The process will take some time, especially fitting it to the device. Be patient. Any errors or warnings will appear in a window that pops up during the process.

Viewing the Report

Several files are generated by the fitter after your project is compiled. The two most useful files are the report file and pinout file. The report file will include the pinout close to the bottom.

The report file is *project_name.rpt*
The pinout file is *project_name.pin*

Assigning Pinout

Now that you have compiled the project once and let MAX+plus II assign the first pinout, you can set it so it remains constant in three ways.

First, you can directly edit the configuration file *project_name.acf*. If you edit the configuration file then you will have to set the project name again for the changes to take effect.

Second, you can use the graphical interface provided by MAX+plus II to modify this file:

Assign > Pin/Location/Chip

Enter the name of the node and the corresponding pin assignment. Be sure to select the **Add** button after each assignment. See the [information](#) on the pin number mapping between the FPGA and the interfaces to the kit.

The last and easiest option to assign the pinout is to let MAX+plus II back assign the pin numbers automatically to the configuration file for the project:

Assign > Back-Annotate Project...

Select the *Chip, Pin & Device* option. However, you may need to modify this automatic assignment to ensure correct operation with the kit pin assignments.

WARNING - BURNOUT

A number of students have burned out FPGAs. The FPGAs that have been fried are the 10K70 (the right hand one). The 10K70 is surface mounted and it is expensive to replace. In addition, the pc board is often damaged in trying to replace the 10K70. We have a limited number of pc boards!

Altera states that unused pins **MUST** be unconnected. This is only possible for the pins connected to the /AD bus if you do not connect anything to the /AD bus. This means that you must not have any wires in /AD0-31 for unused pins to the Altera FPGA and also that you must not have NUSW grounded if any unused pins on the Altera FPGAs are connected to /AD16-31. In other words, remove the jumper which connects NUSW to ground.

It is ok, perhaps even desirable, to have NUHEX grounded and /CLK wired to /LHEX and /HHEX so that the hex leds always display the /AD bus.

Another way to avoid these contention problems is to use all pins connected to the /AD bus. You, of course, could use them as inputs to the FPGA. If you wish to use them for other purposes, then you must explicitly declare them as outputs and explicitly tristate them.

Unused pins must be specifically listed in the top level entity and specifically tristated. See the example files tristate.acf and tristate.vhd in the directory.

If in doubt, read the [handout](#) that describes the kit wiring. All of these /AD bus pins must be used as inputs, outputs, or specifically tristated. In addition, all pins that are connected to a 50-pin connector (see below) should be used as inputs, outputs, or specifically tristated.

Please use the 10K10 (on the left) until you are familiar with the use and programming procedures for Altera FPGAs. This device is in a socket, and at least we can replace it if you burn it out.

Simulation

MAX+plus II provides a simulator and waveform generator for verifying that your code performs the way you expect it to. To simulate your design, you must first create a waveform file that includes input waveforms and specifies which output waveforms to watch. Do this by using the waveform editor:

```
MAX+plus II > Waveform Editor
```

Now select the input and output nodes from your project:

```
Node > Enter Nodes from SNF...
```

This will allow you to select multiple nodes at once by highlighting the names and clicking the arrow button to move them into the box on the right. Click **OK** to add these nodes to the Waveform Editor. Now you will want to define the input waveforms by selecting each waveform and using the toolbar along the left.

Set your maximum simulation length by:

File > End Time

Once you have done this be sure to save as *project_name.scf*.

Now you are ready to run the simulator.

MAX+plus II > Simulator

Deselect all options if any are selected and press **Start**. You can view the results in the Waveform Generator. If you have already closed this window, then click **Open SCF**.

Programming

Set up your lab kit next to a PC with a ByteBlaster programmer in the Digital Lab. Be sure to plug the programming cable into the programming port next to the chip you wish to program. Each port programs the chip closest to it. Be sure to align pin 1 by matching the **red** marks when plugging in this cable. The red stripe should be on the left. Power on your kit. Now open the programmer from MAX+plus II:

MAX+plus II > Programmer

Program your FPGA by using Altera's MAX+plus II software.

If this is the first time that you are using the programmer, a hardware configuration window will appear. Likely the appropriate hardware options will be displayed on the computer case. If not, see the beginner's guide or the handout ``How to Use Max+plus II''. You can edit the hardware configuration at any time as follows:

Options > Hardware Setup...

To speed up the programming process, you should turn off the automatic blank-check and verify. Select the following menu option:

Options > Programming Options...

A pop-up window will appear. If any of the options are selected, de-select them. Click **OK**.

Now you have to specify the programming method. Make sure that Multi-Device JTAG Chain is selected from the JTAG menu.

JTAG > Multi-Device JTAG Chain

If this is the first time that you have used this option, the configuration window will pop up automatically. From the Device Name pull-down menu select EPC2LC20. Click **Select Programming File...** and choose the file called *project_name.pof*. Click **Add**. Click **OK**.

Now you are ready to program the device. To do this, click **Program** from the main programming window. Programming will take a while. When the programmer is done, it will return a "Programming Successful" message.

You will have to turn the power to your kit off and on to get the new program to load into the FPGA.

Appendix: Differences between MAX+plus II and Warp

MAX+plus II is the software provided by Altera for writing Verilog, compiling and programming their FPGAs. Cypress provides a similar tool for their CPLDs called Warp. If you already have experience with Warp, you may be interested in the specific differences between the two programs.

Working With Projects

In MAX+plus II projects are handled a bit differently. The project is defined and set to the same name as the top level file. Each Verilog file must have the same name as that of a component defined in them. Furthermore, all components referenced by a top level Verilog file must be in the same directory as the top level file.

Assigning Pinout

Unlike Warp, the pinout can not be specified in the Verilog file. The *.acf* file holds all the device configuration information including pin assignments, device, configuration device, and logic block assignments. This file can be edited directly, but Altera also provides a way of editing each of these options through the graphical user interface.

QUICK REFERENCE

Opening a Project

File -> Open
Select top level .v file, and click on OK
File -> Project -> Set Project to Current File

Compiling a Verilog File

MAX+plus II -> Compiler
Assign -> Device Set Device to FLEX10K (regardless of whether you are using

the 10K10 or the 10K70)

Set Devices to EPF10K10LC84-3 if you are using the 10K10,
and to EPF10K70RC240-2 if you are using the 10K70.

Click on the "Maintain Current Synthesis...".

Click on Device Options.

Under the Configuration pull-down menu choose EPC2LC20.
To Compile click on Start.

Simulating your Project

MAX+plus II -> Waveform Editor
Node -> Enter Nodes from SNF
Click on List, and move signals you want to view to "Selected Nodes"
File -> End Time
Set your maximum simulation length.
Right click on the signal you want to modify. (You can highlight part of the waveform and edit parts of the signal as in Galaxy.)
(Option 1) Select the Overwrite submenu. Choose either Low, High or Clock (for bits) or Group (for busses). For Clock, leave the settings as they are.
(Option 2) Use the toolbar to select 0, 1, X, Z, invert, clock, count, or group.
File -> Save
MAX+plus II -> Simulator
Click on Start. The Waveform Display window will change to display the simulation. To zoom in or out, use the +/- magnifying glass icons on the toolbar.

Programming your Device

First you MUST tristate (blank) the device you are not using. See the WARNING at the beginning of this handout. Program this device using the appropriate file. The files are compiled and ready to program.

MAX+plus II -> Programmer
Options -> Hardware Setup
Select ByteBlaster for Hardware.
Select lpt1 for Parallel Port.
JTAG -> Multi-Device JTAG Chain Setup
Click on Delete All
Click on Select Programming File
Select your .pof file generated by the compiler
Click on Add

Plug the programming cable into the left port of your FPGA board if you are programming the 10K10, and into the right port of the FPGA board if you are programming the 10K70. Make sure that the programming cable is not plugged in backwards. The red stripe should be on the left with the front of the kit towards you.

(Optional) Options -> Programming Options
You can select/deselect Blank-Check, Verify, Test. It is suggested that

you Verify the first few times you program.
Click on Program.