VHDL MIPS Synthesis and Simulation using Altera CAD Tools and the UP1 CLPD Board

Introduction

The purpose of this lab is to introduce the student to Complex Programmable Logic Devices (CPLD's), Logic Synthesis using VHDL and modern Computer Aided Design, CAD, tools for logic synthesis and simulation. VHDL is a hardware modeling language that has syntax similar to ADA or PASCAL. The entire process will be defined and illustrated in a step-by-step fashion in order to familiarize the student with the design process in the shortest possible time.

A MIPS model in VHDL for the Altera CAD tools with a demo program is available on the web. The VHDL synthesis tools automatically convert a VHDL description into a hardware design. This tool runs logic and state minimization so there is no need for K Maps or espresso. Altera has donated a CDROM student version of this tool, which you can run on your home PC or you can use the professional versions installed in the lab. If you use the student version at home, your PC should have 16-64M memory and around 100M of disk space open. After installing the student version it generates a unique code and you must request a matching secret code via email to enable the software. If you reinstall or change PCs another code is required. This authorization process could take a day or two to get the email back, so start early if you want to run the student version at home. The professional version of this tool costs $5,000 a copy.

Your VHDL MIPS computer model, top_spim.vhd, is combined with a larger VHDL model, top_flex.vhd which simulates and generates the VGA video display of the MIPS computer system. The CPLD chip itself generates the VGA signal. A PC is used only to download the chip. The CPLD board has an Altera FLEX 10K20 CPLD chip that can be used for up to 20,000 gate designs. You can simulate your MIPS design without the video interface and would normally simulate it only since the video output slows down the simulations quite a bit.

You can download a zip file with the MIPS VHDL computer models at:

www.ece.gatech.edu/users/hamblen/ALTERA/mips.zip

Unzip it to drive:/max2work/mipsrgb, where drive is the disk drive on which you installed the student edition. Free unzip software can be found on the web at locations such as:

www.win95.com

Some additional information on the software and the board can be found at:

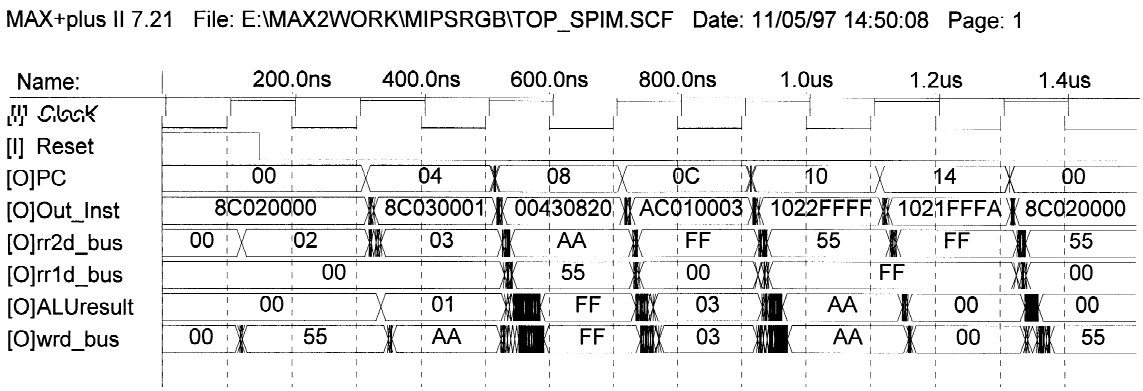
www.ece.gatech.edu/users/hamblen/ALTERA/altera.htm

Simulation Vector file for State Machine Simulation

This file controls the simulation and tests the MIPS computer by setting up a clock, reseting and running for a few instructions. You may need to change the input patterns to test future versions of your MIPS computer. The numbers left of the ">" are the time at which the patterns change.

```
% Simulation Test Vectors for MIPS Machine - File: top_spim.vec
%
START 0us;
STOP 1500ns;
% Generate Clock %
INTERVAL 100ns;
INPUTS ClocK;
PATTERN
0 1;
% Reset MIPS Machine %
INPUTS Reset;
PATTERN
0>1
150>0;
OUTPUTS PC Out_inst rr2d_bus rr1d_bus ALUresult wrd_bus;
```

Altera Simulation of Top_spim.vhd using Top_spim.vec vector file



Running the MIPS Simulation on the Altera CAD tools

After installing the software, code and mips.zip file, make Top_spim.vhd the current project with *File -> Project -> Name.* If you switch between the student version and the full version in the lab you will need to ignore errors in the acf file and replace the acf file when prompted. This is because the lab version is newer than the student version. This will only happen once when you switch versions.

Then find and select Top_spim.vhd.

Select *File-> Project -> Save Compile and Simulate.*
The simulator will run automatically if there are no compile errors. Select Open SCF to see the timing diagram display of your simulation. Whenever you change your VHDL source you need to repeat this step. If you get compile errors, clicking on the error will move the text editor to the error location. The Altera software has extensive on-line help including VHDL syntax examples.

Make any text changes to Top_spim.vhd or Top_spim.vec (test vector file) with *File - > Open* . This brings up a special editor window – note that the menus at the top of the screen change depending on which window you are currently in.

To update the simulation with new test vectors from a modified Top_spim.vec file Select *File -> Open* Top_spim.scf or select the timing display window if it is already open. If you have changed Top_spim.vec update the scf file with *File -> Import Vector File*. You can then hit Start on the simulation window to rerun the simulation with new test vectors.

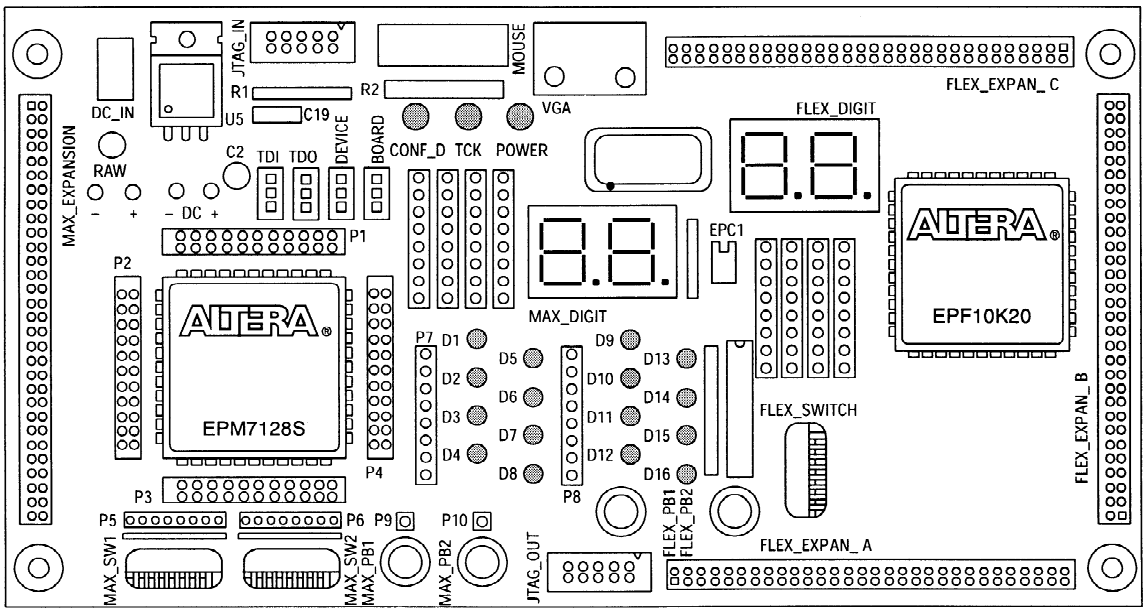
Running the Video MIPS Computer System on Real Hardware (After Simulation Works)

Make Top_flex.vhd the current project with *File -> Project -> Name* Then find and select Top_flex.vhd. Top_flex.acf must be in the project directory since it contains the FLEX chip pin assignment information. Any changes made in the top_spim.vhd (top level signals) must also be made in the top_flex.vhd file.

Select *File-> Project -> Save and Compile*. Top_flex.vhd will link in your top_spim.vhd file that is in the same directory, when it is compiled. This is a big program so it will take a few minutes to compile. Try running on a fast machine with 32M or more of memory.

Select *MaxPlus -> Programmer*. Pull JTAG menu at top to see that Multi-Device is turned on. In *JTAG-> multi device JTAG chain setup* select EPF10K20 as the device and top_flex.sof as the programming file to download. See UP1 board manual pages 20-22 for details a copy is included on the following page. Under *options -> hardware* select byteblaster. Note: Jumpers must be set for flex only on UP1 board and cable must be plugged into printer port. When everything is setup, the configure button in the programming window should highlight. To download the board click on the highlighted configure button.

MIPS output should appear on the VGA monitor after downloading is complete. Flex PB1 is run/step and Flex PB2 is reset (note: these are the pushbuttons to the right and top of the board). The FLEX seven segment LED shows the values of the MIPS PC signals in hexadecimal.



Altera UP1 CLPD Demo Board

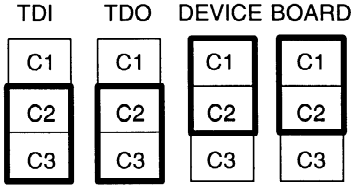
EPF10K20 Configuration

This section describes the procedures for configuring only the EPF10K20 device, (i.e., how to set the on-board jumpers, connect the ByteBlaster download cable, and set options in the MAX+PLUS II software).

Setting the On-Board Jumpers for EPF10K20 Configuration

To configure only the EPF10K20 device in a JTAG chain, set the jumpers TDI, TDO, DEVICE, and BOARD as shown in Figure 8.

Figure 8. Jumper Settings for Configuring Only the EPF10K20 Device



Connecting the ByteBlaster Download Cable for EPF10K20 Configuration

Attach the ByteBlaster directly to the PC's parallel port and to the JTAG_IN connector on the UP 1 Education Board. For more information on setting up the ByteBlaster, go to the *ByteBlaster Parallel Port Download Cable Data Sheet*.

Setting the JTAG Options in MAX+PLUS II for EPF10K20 Configuration

The following steps describe how to use MAX+PLUS II to configure the EPF10K20 device in a JTAG chain. For more information on how to configure a device, see MAX+PLUS II Help.

1. To configure more than one EPF10K20 device, turn on the *Multi-Device JTAG Chain* command (JTAG menu) in the MAX+PLUS II Programmer.

2. Choose **Multi-Device JTAG Chain Setup** (JTAG menu).

3. In the **Multi-Device JTAG Chain Setup** dialog box, select *EPF10K20* in the *Device Name* drop-down list box.

4. Type the name of the programming file for the EPF10K20 device in the *Programming File Name* box. The **Select Programming File** button can also be used to browse your computer's directory structure to locate the appropriate programming file.

5. Choose **Add** to add the device and associated programming file to the *Device Names & Programming File Names* box. The number to the left of the device name shows the order of the device in the JTAG chain. The device's associated programming file is displayed on the same line as the device name. If no programming file is associated with a device, "<none>" is displayed next to the device name.

6. Choose **Detect JTAG Chain Info** to have the ByteBlaster check the device count, JTAG ID code, and total instruction length of the JTAG chain. A message just above the **Detect JTAG Chain Info** button reports the information detected by the ByteBlaster. You must manually verify that this message matches the information in the *Device Names & Programming File Names* box.

7. To save the current settings to a JCF for future use, choose **Save JCF**. In the **Save JCF** dialog box, type the name of the file in the *File Name* box and then select the desired directory in the *Directories* box. Choose **OK**.

8. Choose **OK** to save your changes.

9. In the MAX+PLUS II Programmer, choose **Configure**.

Steps Required for Future Labs

1. If working on a PC at home, install the Altera software and obtain via email the secret code. Install the top_flex project directory on your machine.

2. Read the example VHDL program and see how it implements the MIPS computer by looking at the diagrams in the text.

3. For future labs, modify your *vhd files using the Altera VHDL Editior. In the editor the right mouse button can be used to pull up a VHDL template for most common statements. The main help menu also has many VHDL syntax examples. Compile the program(s) and fix any syntax errors. Clicking on the error will send you to the error location in the editor.

4. Edit Top_spim.vec, to add test cases for your new state machine. This will only require changes to the sensor input patterns. The number next to each pattern is the time at which it will change in the simulation. You can add more times and patterns, when needed.

5. Run the simulation and verify that your state machine is working properly. See running the simulation section for details.

6. Once the simulation works, to run on real hardware compile top_flex.vhd with your new *.vhd files in the project directory. This adds the video output and simulation code to your state machine. See running the video MIPS section for details.