Building a bitstream file for the Insight Virtex-II/1000 Evaluation Board.

MicroBlaze™

Application Note # 202 *Version 2.0*

by Jack Neithardt
Nohau
Campbell, California (888) 886-6428 (408) 866-1820 www.nohau.com

August 1, 2002

Background

This application note is designed to help connect the Nohau MICROBLAZE-PC emulator to the Insight Virtex(TM)II-1000 evaluation Board. It is assumed you have a working knowledge of the Xilinx ISE tools.

The Emulator

The Nohau EMUL-MICROBLAZE-PC connects to targets containing a VirtexII-1000 or Spartan-II/E FPGA running a Micro-Blaze Core. The EMUL-PC/USB-JTAG hardware has 10-pin connector for the target connection, the other side connects to a USB connector port on the host PC. The debugger software includes a pre compiled bit stream containing the MicroBlaze core, a debug trace module and the logic for the Nohau evaluation boards. This is automatically downloaded when the debugger is started. For using the debugger with other targets we provide a net list file for the Spartan-IIE (debugtraces.edf), Virtex-II (debugtracev.edf) and the pin description files (system.mhs and system.mss). We also provide pre-compiled bit streams for other targets. This application note outlines the steps to add the debug module to the Insight example project.

The Insight VirtexII-1000 MicroBlaze evaluation Kit

The Insight VirtexII-1000 MicroBlaze evaluation Kit as distributed by Insight consists of a development board containing a VirtexII-1000 and example projects. The board also has two connectors for plugging in optional cards, some switches and two 7-segment LED's.

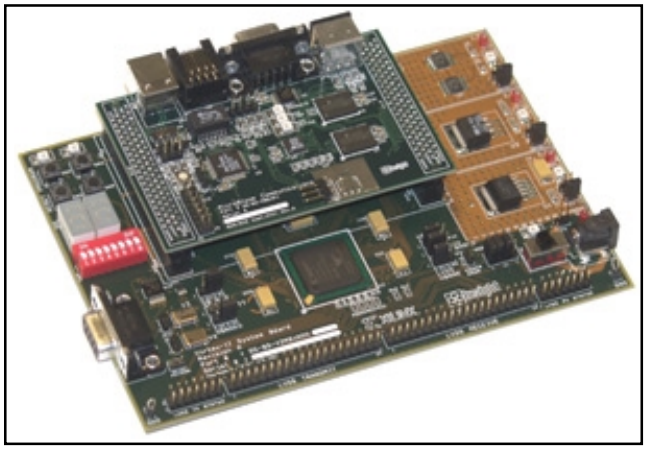

Figure 1 Insight Memec VirtexII Evavluation Board

Connecting the EMUL-PC/USB-JTAG

The target board as shown in Figure 1 has a single row connector (J2) on the edge of the board. Plug the EMUL-PC/USB-JTAG onto the adapter and J4, on the bottom of the adapter onto J2 of the target. See Figure 2.

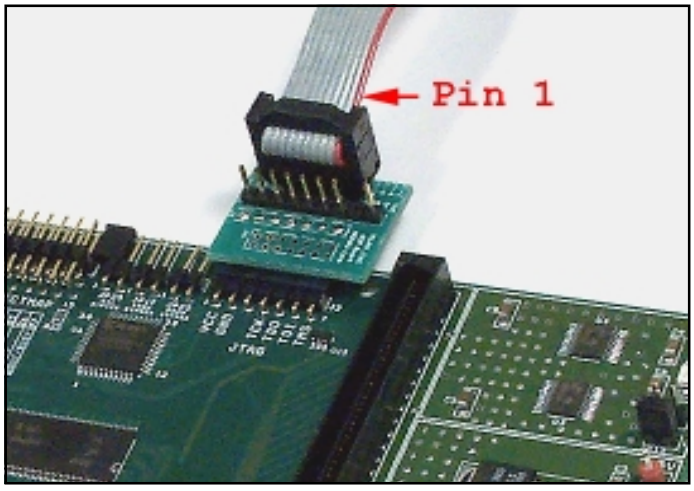

Figure 2 Adapter connection

If you already have the Xilinx JTAG programming cable attached to the target then plug that into J2 of the adapter as shown in Figure 3.

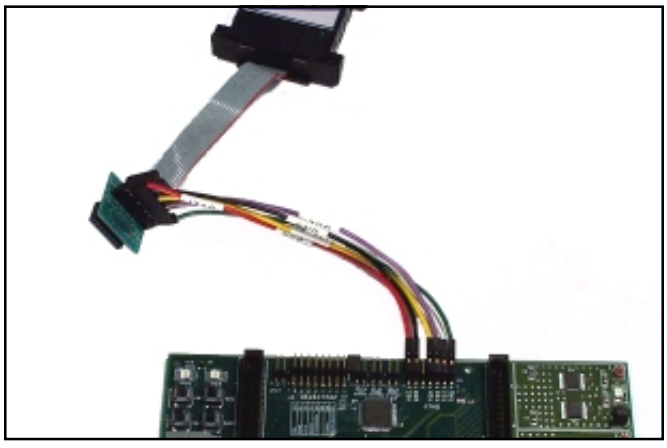

Figure 3 Target Connection Using the Xilinx JTAG Cable

Creating the bit stream file

The process requires editing a couple of files, checking for other JTAG instances, and using the makefile to rebuild the project using the steps outlined here.

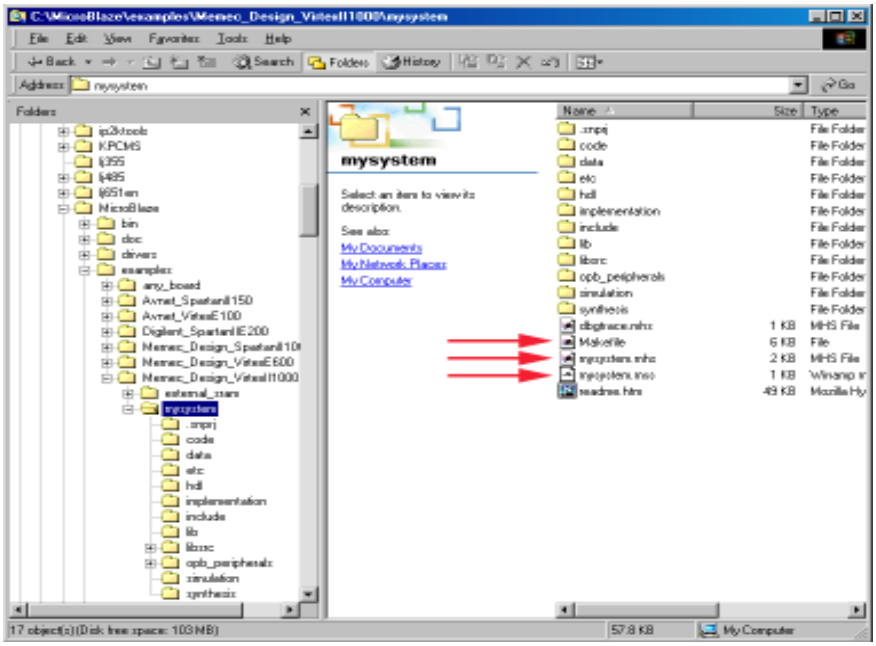

Figure 4. Directory structure

1) Locate the mysystem.mhs, mysystem.mss and makefile files in project directory using explorer. This directory is C:\microblaze\examples\Memec_Design_VirtexII1000\mysystem shown in figure 4.

2) Browse to the C:\nohau\seehaublaze\examples directory and copy the obp_peripherals directory to the C:\microblaze\examples\Memec_Design_VirtexII1000\mysystem directory.

3) Go to the ..mysystem\opb_peripherals\debugtraceblaze\netlist directory. Copy the debugtracev.edf file to debugtrace.edf.

4) Open the makefile in an editor. Search for the string "mode". This will usually be MODE = executable. Place a # sign before the word mode and add new line MODE = xmdstub.

5) Find the rule for implementation/$(SYSTEM).bit. add a second copy statement: cp implementation/$(SYSTEM).bit implementation/EVAL_VX2_1000.bit. Be sure to follow the indent structure. After any changes have been made, save the file exit.

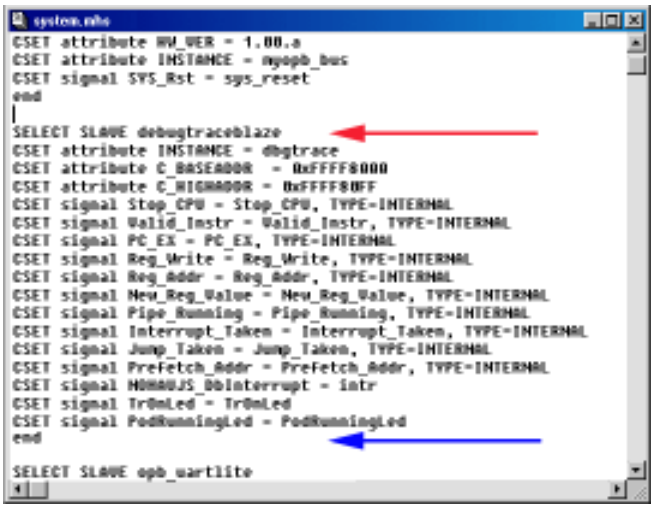

Figure 5 Copy debugtraceblaze from the system.mhs file

6) Open the file mysystem.mss in an editor. An excerpt from the file is shown here. If a DEBUG_PERIPHERAL is defined replace the text in red with "dbgtrace". If it is not defined, add the line SET attribute DEBUG_PERIPHERAL = dbgtrace.

SET attribute HW_SPEC_FILE = mysystem.mhs

SET attribute DEBUG_PERIPHERAL = *myuart*

SET attribute EXECUTABLE = code/gpio_interface.out

SET attribute XMDSTUB = code/xmdstub.out

7) Check for the XMDSTUB attribute, if it is not present then you need to add it. This is the last line shown above.

8) Open the file mysystem.mhs in an editor. Using find search for JTAG. If you find a JTAG peripheral select the definition and delete it.

9) Open the file system.mhs in C:\nohau\seehaublaze\examples\opb_peripheral directory.

10) Now highlight from the text "Select Slave debugtraceblaze" to the text "end" in ..\mysystem\opb_peripherals\system.mhs. This is the text between the red and blue arrows as shown in figure 5. Copy this text.

11)Now insert this into the mysystem.mhs file at the point shown by the red arrow in figure 6. Save the file and exit.

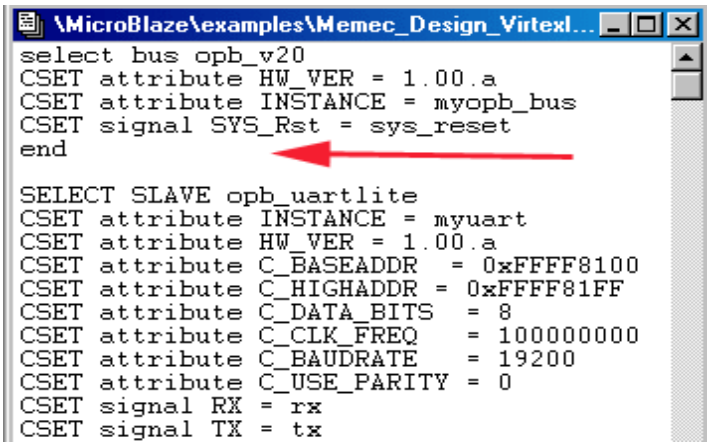

Figure 6 debugtraceblaze.mhs inserted in the mysystem.mhs file

12) Now highlight from the text "SELECT MASTER microblaze" to the text "end" in ..\mysystem\opb_peripherals\system.mhs. This is the text between the red and blue arrows as shown in figure 7. Now replace the microblaze definition in the mysystem.mhs file. Save the file and exit.

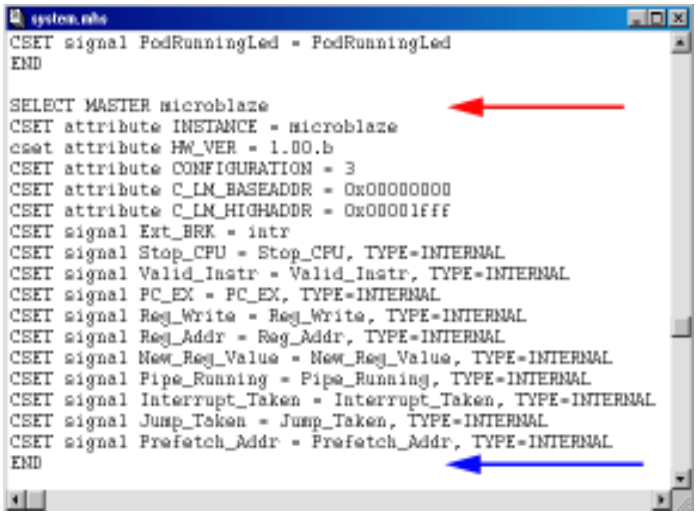

Figure 7 mciroblaze definition in the system.mhs file

13) Open a command prompt window at the project directory. This is C:\microblaze\examples\Memec_Design_VirtexII1000\mysystem, for this example.

14) Type "make bits", now sit back and wait.

15) In the C:\nohau\seehaublaze\logic rename the file eval_VX2_1000.bit to eval_VX2_1000_orig.bit.

16) Now go back to the explorer window and copy the resulting bit file Eval_VX2_1000.bit from the directory C:\microblaze\examples\Memec_Design_VirtexII1000\mysystem\implementation to the C:\nohau\seehaublaze\logic directory.

The bit stream is now ready for testing.

Using the new bit stream in Seehau

Make sure the target board is powered on and EMUL-PC/USB-JTAG is connected to the target and the USB port of the PC. If you have already configured SeehauBlaze for the correct chip than start Seehau, otherwise run the Seehau Config program to select the correct device.

1) After Seehau starts go to the file menu, select Load Code and load your program. For this example it is mysystem.out in the code subdirectory.

2) After the program has loaded go to the Config menu and select Emulator. Click on the Misc. tab and enter 0x400 for the PC and no value fore the stack pointer. Click okay.

3.) Now save the settings by clicking on the Config menu and then select Save Settings.

4.) Click the reset button and the cursor should be 0x400 in the assembly tab.

5) Click the Source Step button and the cursor should move to the first source line in main.

6) Click go and the program should be running.

Some Helpful Notes and Reminders

1) Remember to use the correct debugblazetrace.edf netlist file for the FPGA platform you are using. The file for the Virtex is called DEBUGBLAZETRACEV.edf and is 520K.

2) Ensure that the MICROBLAZE attribute: CSET attribute C_LM_HIGHADDR = 0x00001fff.

Conclusion

A few steps to add the debug and trace to an existing project for target hardware and run a target application.